R-HybrID: Evolution of Agent Controllers with a Hybridisation of Indirect and Direct Encodings

Fernando Silva
BioMachines Lab & Instituto de Telecomunicações & BioISI – FC/UL
Lisboa, Portugal
fsilva@di.fc.ul.pt

Luís Correia
BioISI – FC/UL
Lisboa, Portugal
luis.correia@di.fc.ul.pt

Anders Lyhne Christensen
BioMachines Lab & Instituto de Telecomunicações
Lisboa, Portugal
anders.christensen@iscte.pt

ABSTRACT

Neuroevolution, the optimisation of artificial neural networks (ANNs) through evolutionary computation, is a promising approach to the synthesis of controllers for autonomous agents. Traditional neuroevolution approaches employ *direct encodings*, which are limited in their ability to evolve complex or large-scale controllers because each ANN parameter is independently optimised. *Indirect encodings*, on the other hand, facilitate scalability because each gene can be reused multiple times to construct the ANN, but are biased towards regularity and can become ineffective when irregularity is required. To address such limitations, we introduce a novel algorithm called *R-HybrID*. In R-HybrID, controllers have both indirectly encoded and directly encoded structure. Because the portion of structure following a specific encoding is under evolutionary control, R-HybrID can *automatically* find an appropriate encoding combination for a given task. We assess the performance of R-HybrID in three tasks: (i) a high-dimensional visual discrimination task that requires geometric principles to be evolved, (ii) a challenging benchmark for modular robotics, and (iii) a memory task that has proven difficult for current algorithms because it requires effectively accumulating neural structure for cognitive behaviour to emerge. Our results show that R-HybrID consistently outperforms three state-of-the-art neuroevolution algorithms, and effectively evolves complex controllers and behaviours.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Connectionism and neural nets, concept learning*

General Terms

Algorithms

Keywords

Agent controller; artificial neural network; evolutionary computation; genetic encoding; neuroevolution

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May, 4–8, 2015, Istanbul, Turkey.*

1. INTRODUCTION

Artificial evolution emerged in the 1990s as a promising alternative to classic artificial intelligence in the synthesis of robust control systems for simulated agents and embodied agents such as robots [9, 15]. In general terms, the artificial evolution approach maintains a population of genomes, each encoding a number of parameters of the agents' control system. Optimisation of genomes is based on Darwin's theory of evolution, namely blind variations and survival of the fittest, as embodied in the neo-Darwinian synthesis [12].

Early studies on artificial evolution advocated the use of artificial neural networks (ANNs) as the underlying control system of autonomous agents [9, 15]. The ANN paradigm was adopted due to a number of features such as the ANNs' robustness in dynamic, real-world environments [9], and the argued tight coupling between perception and action [15]. In this context, the parameters of the ANN such as the connection weights and occasionally the topological structure are optimised by an evolutionary algorithm, a process termed *neuroevolution* [10].

As discussed by Husbands *et al.* [15], a central aspect in neurovolution of controllers is the *genetic encoding*. A key commonality among the majority of past and current studies is the use of a *direct encoding* [24], in which each parameter of the ANN is independently specified and optimised. As a result, direct encoding methods tend not to perform well in high-dimensional and in epistatic tasks, and are therefore unsuited for the evolution of agent behaviours that require complex large-scale ANNs [15, 21]. *Indirect encodings*, also called generative or developmental encodings, enable representational efficiency by allowing the same gene to be reused multiple times to construct different parts of the phenotype [38]. That is, indirect encodings allow solutions to be represented as *patterns* of parameters, rather than requiring each parameter to be represented individually [2, 36, 38].

Indirect encodings can facilitate scalability because evolution can search in a low-dimensional space and produce arbitrarily large phenotypes. One particular indirect encoding neuroevolution algorithm called HyperNEAT [36] has shown significant potential in the evolution of controllers for agents because of its ability to automatically find the geometric aspects of a task [5, 6, 36]. For example, in gait learning for simulated quadruped robots, HyperNEAT can exploit multiple symmetric relationships between the robot's legs to evolve high-performing controllers [4]. An important result is that, contrary to other approaches, HyperNEAT does not need a human experimenter to manually decom-

pose the task to find the underlying regularities [5, 7, 11, 36]. However, as typical algorithms relying on indirect encodings, HyperNEAT's bias towards controllers with regular structure makes it difficult to properly account for irregularities such as faults in the joints of quadruped robots [5].

In this paper, we propose R-HybrID, a novel approach for the evolution of ANN-based controllers for autonomous agents. R-HybrID automatically combines HyperNEAT with its direct encoding counterpart NEAT [37]. We compare the performance of R-HybrID with the performance of three state-of-the-art algorithms: (i) switch-HybrID [5], a comparable algorithm that requires the human experimenter to decide when to switch from an indirect encoding to a direct encoding, (ii) HyperNEAT [36], and (iii) NEAT [37].

We assess the performance of the four algorithms in three tasks. The first task is a visual discrimination task [36] in which ANNs have to distinguish a large object from a small object. The task allows us to assess how R-HybrID behaves in high-dimensional search spaces, and if R-HybrID can successfully find the geometric principles required to solve the task. The second task is the coupled inverted pendulum task [13], a challenging benchmark for modular robotics scenarios with multiple local optima, and in which task complexity can be scaled up by increasing the number of robots in the environment. The third task is a memory task [16, 26] that requires successful agents to display cognitive behaviours [16]. Evolving memory behaviours requires accumulating neurons and connections that enable ANNs to maintain a persistent representation of some aspects of the task. However, such scaffolding structure typically does not provide immediate performance benefits, and current algorithms therefore tend to converge to reactive controllers [16].

Overall, our results show that: (i) R-HybrID effectively evolves large-scale ANNs and finds the geometric principles necessary to solve the visual discrimination task, and that (ii) R-HybrID consistently outperforms switch-HybrID, HyperNEAT, and NEAT in multiple configurations of the coupled inverted pendulum task and in the memory task. The main conclusion is that R-HybrID is a promising approach to the evolution of complex agent behaviours due to its ability to automatically explore multiple indirect and direct encoding combinations simultaneously.

2. BACKGROUND

2.1 NEAT

NeuroEvolution of Augmenting Topologies (NEAT) [37] is a prominent direct encoding neuroevolution algorithm that optimises both network topologies and weighting parameters. NEAT has been successfully applied to distinct control and decision-making tasks, outperforming several fixed-topology algorithms [34, 37].

NEAT starts with a population of simple networks with no hidden neurons. Throughout evolution, networks are recombined via crossover, and new neurons and new connections are progressively added as a result of structural mutations. Because NEAT speciates the population based on the topological similarity between networks, the algorithm maintains a variety of complexifying structures simultaneously over the course of evolution. In this way, NEAT can search for an appropriate degree of complexity to the current task while avoiding *a priori* specification of the network topology. A full description of the algorithm is given in [37].

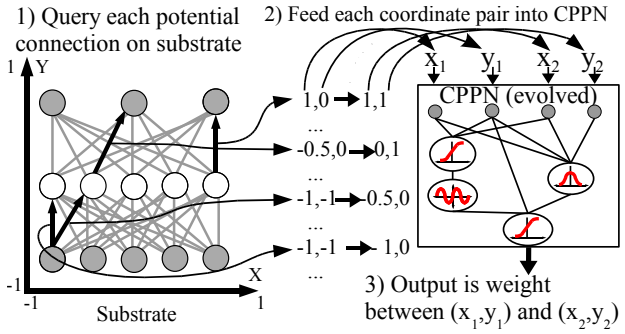

Figure 1: HyperNEAT connectivity patterns production. Neurons in the ANN are assigned coordinates that range from -1 to 1 in all dimensions of a *substrate*. The weight of each connection in the substrate is determined by querying the CPPN. The dark directed lines in the substrate represent a sample of connections that are queried. For each query, the CPPN receives the positions of two neurons, and outputs the weight of the connection between them. As a result, CPPNs can produce regular patterns of connections. From [35]. MIT Press ©.

2.2 HyperNEAT

Hypercube-based NeuroEvolution of Augmenting Topologies (HyperNEAT) [36] is an approach to neuroevolution that employs an evolved indirect encoding based on Compositional Pattern Producing Networks (CPPNs) [35]. Formally, CPPNs are a composition of functions that encode the *weight patterns* of an ANN, as shown in Fig. 1. CPPNs are conceptually similar to ANNs, with the key difference being that different nodes in a CPPN can have different activation functions. Each activation function in CPPNs represents a specific regularity such as symmetry, repetition, or repetition with variation [35]. For example, a periodic function such as sine creates repetition, while a Gaussian function enables left-right symmetry. Regularities *with variation* (e.g. as in the fingers of the human hand) can be evolved by combining regular functions with irregular ones.

In HyperNEAT, NEAT is altered to evolve CPPNs instead of ANNs. CPPNs produce connectivity patterns by interpreting spatial patterns generated within a hypercube as connectivity patterns in a lower-dimensional space (see Fig. 1). Neurons exist at *locations*, and the coordinates of pairs of neurons are input to a CPPN to compute the connection weights for an ANN.

In the evolution of controllers, HyperNEAT yields a number of advantages over other algorithms. One of the key advantages of HyperNEAT is that as the connection weights between neurons are a function of the geometric position of such neurons, HyperNEAT can exploit the neural *topography* and not just the topology. In this respect, HyperNEAT has been shown to: (i) correlate the internal geometry of the ANN with the placement of agent sensors and actuators [36], (ii) evolve high-performing controllers for simulated quadruped robots by exploiting regularities such as four-way symmetry, wherein all legs of the robots continuously move in unison [4, 5], (iii) generalise the learnt geometric principles to create larger-scale ANNs *without* further evolution [36], and (iv) efficiently represent controllers

for large groups of agents as a function of the relationship between the role of the agents and their position in the group [7]. However, HyperNEAT's performance is degraded in tasks that contain irregularities [5]. For example, in the evolution of gaits for quadruped robots, if robots have faulty joints then HyperNEAT's ability to evolve coordinated behaviours is limited. As the number of faulty joints increases, HyperNEAT's performance continuously decreases and becomes statistically indistinguishable from the performance of NEAT, which is blind to task geometry [5].

2.3 Hybridising Encodings

To address HyperNEAT's ineffectiveness when irregularity is required, Clune *et al.* [5] recently proposed the first hybridisation of indirect and direct encodings, an algorithm called *switch-HybrID*. Switch-HybrID first evolves with HyperNEAT, and then switches to its direct encoding counterpart NEAT [37] after a fixed, predefined number of generations. When the switch is made, each HyperNEAT ANN is translated to a NEAT genome. Evolution then continues with regular NEAT until the end of the experiment.

In a number of different tasks, switch-HybrID has been shown to significantly outperform HyperNEAT [5]. Because switch-HybrID is able to make subtle adjustments to otherwise regular patterns, the algorithm can account for irregularities such as faulty joints in quadruped robots [5].

The success of switch-HybrID suggests that indirect encodings may be more effective not as stand-alone algorithms, but in combination with a refining process that adjusts regular patterns in irregular ways. However, a key disadvantage of switch-HybrID is that the switch from HyperNEAT to NEAT is only made after a number of generations defined by the experimenter elapse. Similarly to devising optimal stopping criteria for evolutionary algorithms based on, for example, search space exploration or objective convergence [1], defining an appropriate switch point is a non-trivial task that requires domain-dependent information, and a significant amount of experimentation and human knowledge. In addition, the switch-point criterion limits the applicability of switch-HybrID in open-ended domains such as when agents have to learn while they operate in the task environment and the adaptation process is continuous in time [33].

3. R-HybrID

Conventional evolutionary algorithms are based on either an indirect encoding or on a direct encoding, and therefore ANNs are optimised according to one particular scheme at a given time. Even in switch-HybrID, evolution is limited to first exploring the solution space according to the properties of HyperNEAT and then to the properties of NEAT.

The key idea behind the *representation-based hybridisation of indirect and direct encodings* (R-HybrID)[1] is that ANNs can be partially indirectly encoded and partially directly encoded. Which parts of a given ANN are indirectly encoded or directly encoded is placed under evolutionary control. In this way, the evolutionary process can explore multiple encoding pathways simultaneously, and automatically find appropriate levels of indirect encoding and of direct encoding to solve the current task. Consequently, evolution can explore the continuum of regularity and evolve solutions that are mainly or completely indirectly encoded when the regularity of a task is high, or primarily directly encoded solutions if the regularity of a task is low. Given that many real-world problems have regularities but typically also require irregular exceptions to be made [5], an algorithm that automatically explores indirect and direct encodings may be an important step towards the evolution of more complex agent behaviours.

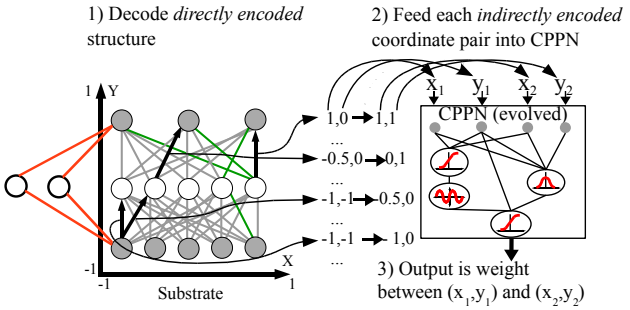

Figure 2: R-HybrID connectivity patterns production. Samples of directly encoded and of indirectly encoded connections are represented in the substrate by the green undirected connections and by the black directed connections, respectively. Firstly, directly encoded connection weights are decoded. Secondly, indirectly encoded connection weights are decoded by querying the CPPN. Besides evolving the CPPN, R-HybrID can also add new directly encoded neurons and connections, represented in the left part of the substrate by the red undirected connections and corresponding neurons.

R-HybrID combines HyperNEAT and NEAT in an adaptable manner. Genomes are composed of a direct encoding part, similar to NEAT, and an indirect encoding part, i.e., a CPPN. Each genome in the initial population is assigned a ratio $r \in [0, 1]$ of directly encoded connection weights, which are randomly chosen. The remaining $1 - r$ connection weights are indirectly encoded via a CPPN. Throughout evolution, the set of operators of NEAT and HyperNEAT (crossover, mutation) is applied to the direct encoding and to the indirect encoding parts of the genome. In addition, through mutation, connections that are directly encoded can become indirectly encoded and vice-versa. When a connection is changed from directly to indirectly encoded, the corresponding weight will afterwards be produced by a CPPN. In this way, the evolutionary process can search for solutions across multiple ratios of indirect vs. direct encoding.

The decoding of R-HybrID genomes is shown in Fig. 2. Firstly, the directly encoded connection weights are decoded, if any. Secondly, the indirectly encoded connection weights are decoded by querying a CPPN. One key advantage of R-HybrID is that, because of the hybrid genomic representation, the evolutionary process can simultaneously optimise and complexify the CPPNs *and* the directly encoded structure. Because R-HybrID inherits NEAT and HyperNEAT's speciation dynamics, the algorithm can effectively maintain a variety of ANNs with differing complexities and encodings simultaneously over the course of evolution.

[1]An extensive description of R-HybrID, the source code of the experiments described in Section 4, and a complete list of the experimental parameters is available at: `http://fgsilva.com/?page_id=282`.

4. METHODS

In this section, we define our methodology and the experimental setups used to evaluate R-HybrID. We compare R-HybrID with switch-HybrID, HyperNEAT, and NEAT in three tasks with distinct characteristics. The first task is a visual discrimination task [36] that serves to assess how R-HybrID behaves in high-dimensional search spaces, and if R-HybrID can discover the same geometric principles that HyperNEAT has been shown to successfully exploit [6]. The second task is the coupled inverted pendulum task [13], a benchmark for modular robotics scenarios with non-linear dynamics. Besides being a challenging task with multiple local optima [13], the coupled inverted pendulum task allows us to scale up the task complexity by increasing the number of robots that operate in a fixed-length track. In addition, the fact that both NEAT and HyperNEAT struggle to evolve effective solutions when multiple robots are considered [3] raises the question of whether hybrid algorithms such as R-HybrID and switch-HybrID can leverage their distinctive encoding properties to better solve the task. The third task is a memory task [16, 26] that requires agents to display cognitive behaviours. In particular, memory behaviours are challenging to evolve because they require the accumulation of neurons and connections that serve a memory function, and building only part of such scaffolding may provide no immediate performance benefits [16]. When such behaviours have been successfully evolved, domain-dependent information such as task-decomposition approaches and complex neural models have been typically used [16]. Contrary to such approaches, our goal is to study if R-HybrID, a potentially more general algorithm because of its encoding properties, can facilitate the evolution of complex memory capabilities. Each task is described in the following sections.

4.1 Visual Discrimination

Vision tasks are a suitable domain for testing learning algorithms on high-dimensional input and output spaces. The goal in the visual discrimination task [36] is to evolve an ANN that distinguishes a large square object from a small square object in a two-dimensional substrate, as illustrated in the examples in Fig 3. The ANNs' input layer is composed of a two-dimensional array of neurons that are either active or inactive (black or white in Fig. 3). The output layer is an equivalent two-dimensional array of neurons whose activation levels can vary between zero and one. In a given trial, the ANN is presented with the large object and the small object in different locations. The ANN then has to determine the centre of the large object in the visual field. The neuron in the output layer with the highest activation is interpreted as the ANN's selection.

The input layer and the output layer both have a resolution of 11 x 11 neurons, resulting in a total of 14,641 connection weights to be optimised. Solutions to the task need to detect the relative sizes of the two objects, and display a connectivity pattern between the input layer and the output layer that causes the correct neuron to become the most active regardless of the locations of the objects. Thus, one important question is whether R-HybrID can leverage HyperNEAT's ability to discover the task's regularity, and converge to an effective type of solutions. As in [36], the evaluation of an ANN includes 75 trials. Each trial places the two objects in different locations. The fitness score f (to be minimised) is defined as the sum of the squared distances between the centre of the large object and the point of highest activation in the output layer over the 75 trials.

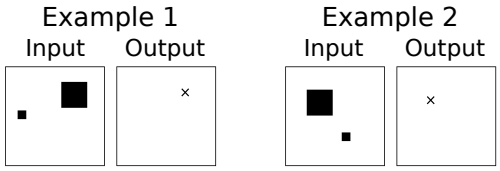

Figure 3: Visual discrimination task. Two examples of input patterns provided to an ANN and the corresponding correct activations in the output layer. The "×" in each example denotes the output neuron with the highest activation, which is the how the ANN specifies the centre of the large object.

4.2 Coupled Inverted Pendulum

The coupled inverted pendulum task [13] is a benchmark designed to evaluate controllers in modular robotics scenarios. The task includes interactions between autonomous robots with decentralised control and non-linear dynamics. We setup the coupled inverted pendulum task as defined in its original formulation [13]. Robots (carts) are coupled by chains and operate in a limited length track. A pendulum is attached to each robot, and all robots have to balance their pendulum in the upper equilibrium position (see [13] for details). Pendulums start in lower positions and therefore a non-linear upswing phase is needed. In combination with a limited acceleration of the robot motor, the upswing can only be achieved by moving back and forth multiple times. In addition to balancing the pendulum, each robot has to avoid pulling the chains and colliding with walls and other robots. Individual robots have a total of 10 sensors [13]: (i) S_0 to S_3: four sensors for determining the pendulum angle, (ii) S_4 and S_5: two proximity sensors that respond to the walls of the track and other robots, (iii) S_6 and S_7: two robot velocity sensors, and (iv) S_8 and S_9: two pendulum angular velocity sensors. Each robot has two actuators, A_0 and A_1, and the acceleration control of the robot is determined by their difference.

In our experiments, we vary the task complexity by increasing the number of robots that operate in a fixed-length track from one to five. Robots have a length of 0.1 m. The chain and track lengths are respectively 0.35 m and 2 m. The fitness score f is defined as the percentage of time that all pendulums spend in the upper equilibrium position. If any constraint of the task is violated (e.g., robots collide with each other or with walls, robots run into chains), a trial is aborted and the fitness score is reduced inversely proportional to the elapsed time.

4.3 Memory Task

Controllers capable of cognitive behaviours, including those requiring memory, are challenging to evolve [16, 26]. Specifically, neuroevolution of memory behaviours requires scaffolding structure, i.e., the accumulation of neurons and connections that enable the ANN to maintain an internal representation of some aspects of the task, but that typically provide no immediate performance benefits [16]. Such controllers are different from *reactive* controllers, whose behaviour is based on responses to current sensory input without any persistent state.

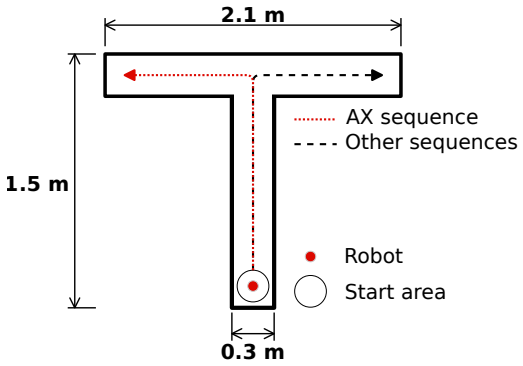

Figure 4: T-maze environment for the memory task. A simulated robot is presented with time-delayed stimuli before it is allowed to navigate in the maze. The robot must reach the end of the left branch when the combined AX stimuli is presented, and the end of the right branch otherwise.

The memory task is conducted in a large T-maze environment (see Fig. 4) similar to those used in previous evolutionary robotics experiments [16, 26] and animal cognition studies [27]. The task consists of a first context stimulus (A or B) followed by a second context stimulus (X or Y). The agent must reach the end of the left branch when the combined stimulus AX is presented, and the end of the right branch otherwise (for AY, BX, BY). In our experiments, the agent is a simulated robot modelled after the e-puck [22], a 7.5 cm in diameter differential drive robot capable of moving at up to 13 cm/s. The robot has six infrared distance sensors with a range of 25 cm and four virtual letter sensors (one for each letter A, B, X, Y). Each letter sensor receives 1 if the letter is presented, 0 otherwise. The ANN inputs are the readings of the sensors. The ANN output layer contains two neurons, one for controlling each wheel of the robot.

The robot is evaluated on each letter sequence six times, for a total of 24 trials. Both motors are disabled during the presentation of the stimulus. Each trial lasts 475 time steps (47.5 seconds) as follows: (i) $1 < t \leq 25$: presentation of the first stimulus, (ii) $25 < t \leq 50$: delay, all sensors are set to zero, (iii) $50 < t \leq 75$: presentation of the second stimulus, and (iv) $75 < t \leq 475$: the robot is allowed to move and must reach the correct end of the maze. To prevent overfitting to a specific initial configuration, the robot's initial position in the start area is randomly chosen in each trial. A trial is aborted if the robot collides with the walls of the maze.

4.3.1 Guiding Evolution

Recent studies on the evolution of memory have shown that one of the main challenges in the synthesis of controllers with such abilities is *deception* [16, 26]. Deception occurs when the fitness function fails to build a gradient that leads to a global optimum, and instead drives evolution towards local optima. That is, a fitness function that rewards the fulfilment of a task requiring cognitive abilities does not necessarily reward the stepping stones that lead to cognitive controllers. As a result, evolution typically converges to *non-cognitive* controllers [16].

To analyse the potential of the algorithms, we define two experimental configurations, henceforth called **fitness** setup and **novelty** setup. In the fitness setup, we use a fitness function that rewards candidate solutions reaching the end of the maze branches [16] and the intermediate progress:

$$f = \sum_{trial=1}^{24} \begin{cases} 500 & \text{if reaches correct end} \\ 250 & \text{if reaches incorrect end} \\ progress/3 & \text{if collides with wall} \\ progress & \text{otherwise} \end{cases} \quad (1)$$

where $progress \in [0, 100]$ is the linearly mapped Euclidean distance between the start position and the end position travelled by the robot in the trial.

In the novelty setup, we use *novelty search* [17]. In novelty search, the objective is to maximise the novelty of behaviours instead of their fitness, i.e., to search directly for novel behaviours as a means to circumvent convergence to local optima. To that end, candidate solutions are scored based on how behaviourally different they are from previously evaluated solutions. Novelty search has attained considerable success in the evolution of controllers for a number of tasks, see [16, 17, 18, 23, 28] for examples.

The application of novelty search requires defining a behaviour characterisation. We follow the approach employed in [16], which consists of a summary of the robot's behaviour across the different trials. A robot's behaviour is characterised by four values: what fraction of trials the robot visits the left branch, visits the right branch, visits the *end* of the correct branch, and visits the *end* of the incorrect branch. Novelty search parameters are set as in [17].

4.4 Experimental Setup and Treatments

In the visual discrimination task, as described in Section 4.1, all algorithms have to optimise the weights of a fixed-topology ANN with no hidden layers. As in [36], neurons have no bias values. In the coupled inverted pendulum and memory tasks, the standard NEAT algorithm is used because the optimal ANN topology is not known *a priori*. The remaining algorithms start with networks that have a hidden layer with five neurons. For the four algorithms, recurrent and self-recurrent connections are allowed, and neuron biases are used. R-HybrID and switch-HybrID can add new neurons and new connections to the CPPNs and to the directly encoded structure. CPPNs compute the ANN weights and bias values as in previous studies [36]. CPPNs use signed activation, resulting in a node output range of [-1,1]. If the magnitude of the output for a particular query is less than or equal to 0.2, the connection weight is set to zero; otherwise the output is scaled to a magnitude between zero and three (the sign is preserved). Each R-HybrID genome in the initial population is equiproportionally assigned: (i) completely directly encoded structure ($r = 1.0$), (ii) entirely indirectly encoded structure ($r = 0.0$), or (iii) both directly and indirectly encoded structure, with r randomly chosen in $]0.0, 1.0[$.

For each task and each algorithm, we conduct 30 independent runs. Each run lasts 1000 generations with a population size of 100. Following previous studies by Clune *et al.* [5], we set the switch point of switch-HybrID at halfway through the evolutionary process, that is, at 500 generations.

When reporting the performance across runs, we consider the best individual of the population: the lowest scoring in the visual discrimination task; the highest scoring in the cou-

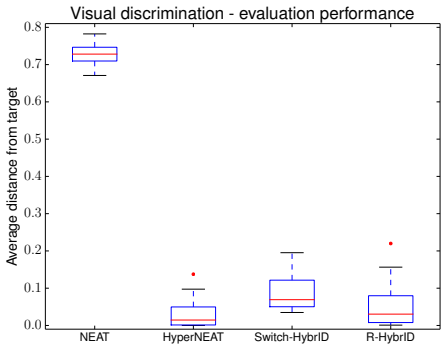

Figure 5: Distribution of the average distance from target of the target's field in the evaluation trials from evolution. Lower scores are better.

pled inverted pendulum task and in the memory task. Besides performance, we also quantify the complexity of ANNs. To the best of our knowledge, there is no general metric for ANN complexity. We use the effective number of parameters in each network, i.e., the sum of the number of connections and the number of neurons that have a bias value.

We use the two-tailed Mann-Whitney U test to compute statistical significance of differences between results because it is a non-parametric test, and therefore no strong assumptions need to be made about the underlying distributions. Success rates are compared using the two-tailed Fisher's exact test, a non-parametric test suitable for this purpose [8].

5. RESULTS

5.1 Visual Discrimination

The goal in the visual discrimination task is to evolve large-scale ANNs that distinguish a large object from a small object. The task provides an important test for R-HybrID because only algorithms that can effectively search through a high-dimensional search space and find the geometric principles that distinguish between the two objects regardless of their location can evolve suitable solutions to the task.

Figure 5 summarises the performance of the algorithms with respect to the *average distance from target* of the target field's chosen position in the *evaluation* trials from evolution (lower scores are better). Note that the width and height of the substrate are 2.0 because the neurons' coordinates range from -1 to 1 (see Fig. 1). Indicative that the space of 14,641 connections is too high-dimensional for NEAT, the algorithm yields a median distance from target of 0.73, and is significantly outperformed by the other algorithms ($\rho < 0.001$, Mann-Whitney). R-HybrID, switch-HybrID, and HyperNEAT, on the other hand, effectively evolve solutions to the task. These algorithms are able to discover the underlying geometric regularities as shown by the median distance from target of 0.03, 0.07, and 0.01, respectively. Both R-HybrID and HyperNEAT significantly outperform switch-HybrID ($\rho < 0.01$, Mann-Whitney). Differences between HyperNEAT and R-HybrID are not statistically significant ($\rho \geq 0.01$, Mann-Whitney), despite HyperNEAT yielding better performance on average.

One important aspect of the visual discrimination task is that the algorithms have to optimise the weights of a fixed-topology ANN. Because the number of connections and neurons does not vary, the regularity of solutions evolved can be isolated and compared in a meaningful manner. Regular structures require less information to be described, meaning that regularity can be measured by compression [20]. Because this minimum description length cannot be computed exactly [19], we approximate regularity by using the procedure described in [5]: we compress an ANNs' weights using the gzip algorithm, write the compressed weights to a plain text file, and assess by which fraction the file size was reduced. Because the order of the weights matters, we repeat this process for 1,000 different permutations. The ratio between the size of the original file and the average size of the compressed files is defined as the *regularity ratio*.

The gzip algorithm is a conservative test of regularity because it searches for repeated symbols but is not able to compress all mathematical regularities (e.g. each connection weight offset by a constant amount). Nonetheless, the regularity ratio is informative and shows that R-HybrID (average ratio of 3.22) evolves solutions of high regularity when compared with those optimised by NEAT and by switch-HybrID (lower regularity, average ratio of 2.26 and 2.51 respectively) and by HyperNEAT (higher regularity, average ratio of 3.91). The high regularity of R-HybrID ANNs is the result of the algorithm converging to solutions that are completely indirectly encoded in 21/30 runs. In the remaining 9 runs, less than 8% of the connections are directly encoded. Overall, these results confirm that R-HybrID can effectively converge to an encoding combination suitable for solving the high-dimensional visual discrimination task.

Table 1: Summary of the coupled inverted pendulum task with one robot. Results are averaged over 30 independent runs for each algorithm.

	Fitness score	Complexity
NEAT	0.96 ± 0.04	37.20 ± 18.27
HyperNEAT	0.72 ± 0.15	26.57 ± 11.12
Switch-HybrID	0.89 ± 0.11	117.80 ± 35.96
R-HybrID	0.87 ± 0.10	107.47 ± 27.50

5.2 Coupled Inverted Pendulum

In the coupled inverted pendulum task, it has been shown that NEAT and HyperNEAT have difficulties in evolving effective solutions [3], especially as the number of robots increases. This result raises the question of whether R-HybrID and switch-HybrID can take advantage of their hybrid encodings properties to produce more effective controllers.

Table 1 summarises the results of the experiments with one robot. Contrary to the visual discrimination task, the algorithms that can make use of a direct encoding component are more well-adapted to solve the task. NEAT yields the highest fitness scores and performs significantly higher than the other algorithms ($\rho < 0.001$, Mann-Whitney). HyperNEAT, on the other hand, performs significantly lower than the remaining algorithms ($\rho < 0.001$, Mann-Whitney). Differences between R-HybrID and switch-HybrID are not statistically significant.

An analysis of the complexity of solutions evolved shows that the algorithms considered synthesise controllers with different characteristics. Overall, HyperNEAT evolves the least complex ANNs. Throughout evolution, HyperNEAT

tries to make exceptions to the regular patterns by progressively decreasing the magnitude of several connection weights because it has a limited ability to make single-connection exceptions. As a result, the final ANNs on average use only 36% of the 74 parameters that are possible to express (remaining are set to zero by the CPPN), and the properties of the ANNs are not well-correlated with the task. On other hand, NEAT evolves slightly more complex ANNs that do reflect the task's underlying structure, as shown by the superior fitness scores.

A different strategy is used by R-HybrID and switch-HybrID. Comparing with NEAT, the two algorithms evolve ANNs with significantly higher complexity. In switch-HybrID, when the switch from HyperNEAT to NEAT is made, the 74 possible ANN parameters are always expressed by the CPPNs, and the average fitness score of the highest scoring ANNs is 0.65 ± 0.09. That is, the majority of the ANN structure is created by the indirect encoding in the first phase of the evolutionary process. After the switch, the direct encoding refines solutions by adjusting connection weights and adding 33 new parameters on average, thereby effectively contributing to the evolution of competitive controllers. In R-HybrID, the synergy between the indirect encoding and the direct encoding components is continuous throughout evolution. The evolutionary process explores multiple encoding combinations simultaneously while progressively adding new directly encoded structure. The final solutions evolved by R-HybrID have on average 84 out of 107 parameters directly encoded, indicating that both encodings contribute to the synthesis of effective controllers.

To assess the algorithms' performance when task complexity is scaled up, we increase the number of robots that operate in a fixed-length track. Figure 6 summarises the results of the experiments that use from two robots to five robots. Indicative of increased task difficulty, the transition from a single robot setting to a two robots setting causes a significant decrease in the fitness scores of all algorithms considered ($\rho < 0.001$, Mann-Whitney).

In terms of the scalability of the algorithms with respect to the number of robots, results show that R-HybrID typically performs better than the remaining algorithms as the number of robots increases. For three and four robots, R-HybrID significantly outperforms NEAT and HyperNEAT ($\rho < 0.01$ and $\rho < 0.001$, Mann-Whitney). For five robots, R-HybrID significantly outperforms all other algorithms ($\rho < 0.001$, Mann-Whitney). In effect, the average complexity of ANNs is quantitatively similar across the distinct multirobot experiments, and comparable (except for NEAT) to the results listed in Table 1 for the single-robot experiments. On average, when multiple robots are used: (i) NEAT ANNs have from 51.78 to 69.83 parameters, (ii) HyperNEAT ANNs have from 27.43 to 33.93 parameters, (iii) switch-HybrID ANNs have from 121.20 to 131.57 parameters, and (iv) R-HybrID ANNs have from 105.37 to 120.13 parameters.

R-HybrID and switch-HybrID always evolve controllers that are significantly more complex than those evolved by HyperNEAT and NEAT. The key idea is that the indirect encoding component of the algorithms, given its ability to reuse information, can effectively create large-scale ANNs that the direct encoding component refines and adjusts. This refinement process can occur either in parallel as in R-HybrID or in a second phase as in switch-HybrID. Importantly, as illustrated in Table 2 by sampling the performance of controllers at intermediate points of the evolutionary process, R-HybrID's ability to search across multiple encoding combinations simultaneously allows it to evolve better solutions in the early stages of the evolution, which contributes to its superior performance in the experiments with multiple robots.

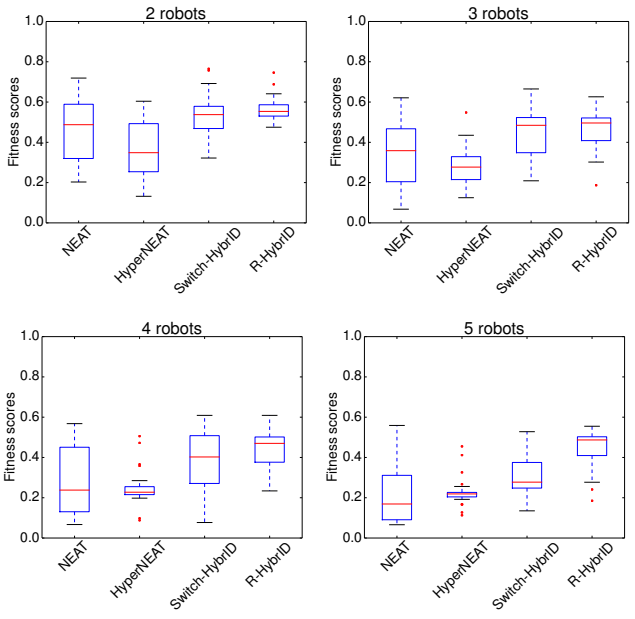

Figure 6: Distribution of the fitness scores in the coupled inverted pendulum experiments with multiple robots. Higher scores are better.

5.3 Evolving Memory Behaviours

In previous studies on the evolution of cognitive behaviours, see [16] for a review, successful synthesis of memory behaviours typically relies on the specialised domain knowledge such as task-decomposition approaches, and on more complicated neural models. Our goal, on the other hand, is to study if R-HybrID's ability to explore multiple encoding combinations simultaneously can contribute to the evolution of memory capabilities. Figure 7 summarises the performance of the algorithms considered in the fitness setup and in the novelty setup. Controllers are divided into four classes based on performance: (i) *poor solutions*, i.e., controllers that often fail to navigate through the maze (fitness score below 7,500), (ii) *reactive policies*,

Table 2: Coupled inverted pendulum task with five robots. Average fitness score of the highest scoring solution at intermediate points of the evolutionary process. Results are averaged over 30 independent runs for each algorithm.

	Generation				
	200	400	600	800	1000
Switch-HybrID	0.15	0.19	0.19	0.27	0.32
R-HybrID	0.22	0.30	0.38	0.43	0.44

controllers that always reach the end of the same branch of the maze regardless of which stimuli are provided (fitness score = 7,500), (iii) controllers that *exceed* the reactive policy but are unable to solve the complete task (7,500 < fitness score < 12,000), and (iv) *perfect solutions* that solve the task (fitness score = 12,000).

In the fitness setup, NEAT successfully solves the task in 9/30 runs, HyperNEAT cannot solve the task, switch-HybrID evolves perfect solutions in 4/30 runs, and R-HybrID yields a success rate of 6/30 runs. Indicative of the task's deceptiveness, evolution converges to *non-cognitive* behaviours in 101/120 runs when considering all runs of the fitness setup. This tendency of fitness-based evolution to converge to non-cognitive behaviours highlights that regardless of the underlying algorithm, a fitness function that evaluates in detail the robot's progress and performance is typically not sufficient to realise instances of cognitive behaviour [16]. In this particular case, the reason is that effectively evolving memory behaviours requires accumulating ANN structure that provides no immediate performance benefits, and that is therefore not recognised by the fitness function.

In the novelty setup, R-HybrID can solve the memory task in 16/30 runs, significantly more often than the other three algorithms ($\rho < 0.01$, Fisher's exact test). NEAT solves the complete task in 6/30 runs, HyperNEAT is not able to solve the task in any of the runs, and switch-HybrID yields a success rate of 3/30 runs. Whereas NEAT and switch-HybrID yield similar performance in the fitness setup and in the novelty setup, HyperNEAT cannot solve the task in any of the setups and evolves poor solutions significantly more often in the novelty setup ($\rho < 0.001$, Fisher's exact test). In addition to evolving perfect solutions more frequently than the other algorithms, R-HybrID also finds them faster. R-HybrID requires on average 388 ± 202 generations to find perfect solutions, while NEAT and switch-HybrID respectively require 487 ± 200 generations and 772 ± 58 generations. In terms of how memory abilities are implemented in the neural architecture, R-HybrID ANNs have on average 7.25 ± 8.58 recurrent connections, while NEAT ANNs and switch-HybrID ANNs have respectively 4.50 ± 4.76 and 4.00 ± 2.65 recurrent connections. The results in the fitness setup are qualitatively similar in the sense that R-HybrID also finds solutions that have more recurrent connections earlier in the evolutionary process. Overall, these combined results illustrate that when open-ended methods such as novelty search are used to drive the evolutionary process instead of a fixed fitness objective, a suitable encoding and evolutionary algorithm is also a key factor underlying the evolution of memory behaviours.

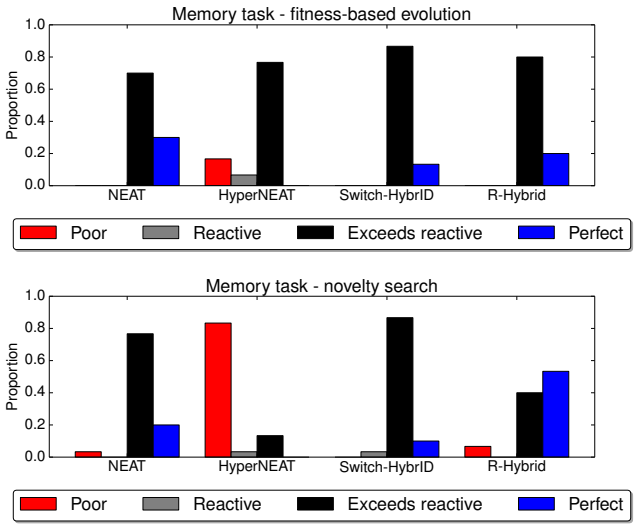

Figure 7: Performance in the memory task. The ability of the algorithms to evolve: poor solutions, reactive controllers, controllers that exceed a reactive behaviour, and controllers that solve the task. The evolutionary process is guided by: a low-level fitness function (top), and novelty search (bottom). Once deception is not present, R-HybrID solves the task more consistently than the other algorithms.

6. CONCLUSIONS

In this paper, we introduced R-HybrID, a novel approach to the evolution of ANN-based controllers for autonomous agents. Contrarily to switch-HybrID, which requires a switch point defined by the experimenter to change from an indirect encoding to a direct encoding, R-HybrID can automatically explore multiple encoding combinations. We assessed R-HybrID in three tasks: (i) a high-dimensional visual discrimination task that requires discovering geometric principles to distinguish two square objects regardless of their location, (ii) the coupled inverted pendulum task, a challenging benchmark for modular robotics scenarios with multiple local optima, (iii) a memory task that has been shown difficult for current algorithms because it requires accumulating neural structure that may provide no immediate performance benefits. We showed that R-HybrID can effectively evolve regular, large-scale ANNs to solve the visual discrimination task, and that R-HybrID typically outperforms switch-HybrID, HyperNEAT, and NEAT in the coupled inverted pendulum task and in the memory task. Overall, our results show that R-HybrID is a potential path forward in the evolution of controllers by automatically combining the pattern-producing capabilities of HyperNEAT with a direct encoding-based refinement process to account for the irregularities that exist in challenging problems [5].

One direction for future work is applying R-HybrID to other challenging domains such as online evolution, in which robots evolve controllers while they operate in the task environment [32], and to tasks that require integrating multiple cognitive behaviours simultaneously [16]. In particular, extending R-HybrID to combine evolution and lifetime learning algorithms [25, 29, 30, 31] may be an important step in the synthesis of controllers for agents. Learning algorithms have been to accelerate the evolution of suitable solutions, a phenomenon known as the Baldwin effect [14]. Because learning rules and parameters are encoded in the genome to be optimised, learning algorithms are likely to face the same issues of non-learning algorithms with respect to the regularity vs. irregularity and suitable encoding aspects.

Acknowledgements

This work was partially supported by FCT under grants SFRH/BD/89573/2012, UID/EEA/50008/2013, UID/Multi/04046/2013, and EXPL/EEI-AUT/0329/2013.

REFERENCES

[1] H. Aytug and G. J. Koehler. New stopping criterion for genetic algorithms. *European Journal of Operational Research*, 126(3):662–674, 2000.

[2] P. Bentley and S. Kumar. Three ways to grow designs: A comparison of evolved embryogenies for a design problem. In *1st Genetic and Evolutionary Computation Conference*, pages 35–43. Morgan Kaufmann, San Francisco, CA, 1999.

[3] L. Cazenille, N. Bredeche, H. Hamann, and J. Stradner. Impact of neuron models and network structure on evolving modular robot neural network controllers. In *14th Genetic and Evolutionary Computation Conference*, pages 89–96. ACM Press, New York, NY, 2012.

[4] J. Clune, B. E. Beckmann, C. Ofria, and R. T. Pennock. Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *IEEE Congress on Evolutionary Computation*, pages 2764–2771. IEEE Press, Piscataway, NJ, 2009.

[5] J. Clune, K. O. Stanley, R. T. Pennock, and C. Ofria. On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*, 15(3):346–367, 2011.

[6] D. D'Ambrosio, J. Gauci, and K. O. Stanley. HyperNEAT: The first five years. In *Growing Adaptive Machines*, volume 557 of *Studies in Computational Intelligence*, chapter 5, pages 159–185. Springer, Berlin, Germany, 2014.

[7] D. D'Ambrosio, J. Lehman, S. Risi, and K. O. Stanley. Evolving policy geometry for scalable multiagent learning. In *9th International Conference on Autonomous Agents and Multiagent Systems*, pages 731–738. IFAAMAS, Richland, SC, 2010.

[8] R. Fisher. *Statistical Methods for Research Workers*. Oliver & Boyd, Edinburgh, UK, 1925.

[9] D. Floreano. Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot. In *3rd International Conference on Simulation of Adaptive Behavior*, pages 421–430. MIT Press, Cambridge, MA, 1994.

[10] D. Floreano, P. Dürr, and C. Mattiussi. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1):47–62, 2008.

[11] J. Gauci and K. O. Stanley. A case study on the critical role of geometric regularity in machine learning. In *23rd AAAI Conference on Artificial Intelligence*, pages 628–633. AAAI Press, Menlo Park, CA, 2008.

[12] S. Gould. *The Structure of Evolutionary Theory*. Harvard University Press, Cambridge, MA, 2002.

[13] H. Hamann, T. Schmickl, and K. Crailsheim. Coupled inverted pendulums: a benchmark for evolving decentral controllers in modular robotics. In *13th Genetic and Evolutionary Computation Conference*, pages 195–202. ACM Press, New York, NY, 2011.

[14] G. E. Hinton and S. J. Nowlan. How learning can guide evolution. *Complex Systems*, 1(3):495–502, 1987.

[15] P. Husbands, I. Harvey, D. Cliff, and G. Miller. Artificial evolution: a new path for artificial intelligence? *Brain and Cognition*, 34(1):130–159, 1997.

[16] J. Lehman and R. Miikkulainen. Overcoming deception in evolution of cognitive behaviors. In *16th Genetic and Evolutionary Computation Conference*, pages 185–192. ACM Press, New York, NY, 2014.

[17] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

[18] J. Lehman, K. O. Stanley, and R. Miikkulainen. Effective diversity maintenance in deceptive domains. In *15th Genetic and Evolutionary Computation Conference*, pages 215–222. ACM Press, New York, NY, 2013.

[19] M. Li and P. M. Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer, New York, NY, 1997.

[20] H. Lipson. Principles of modularity, regularity, and hierarchy for scalable systems. *Journal of Biological Physics and Chemistry*, 7(4):125–128, 2007.

[21] J.-A. Meyer, P. Husbands, and I. Harvey. Evolutionary robotics: A survey of applications and problems. In *1st European Workshop on Evolutionary Robotics*, volume 1468 of *LNCS*, pages 1–21. Springer, Berlin, Germany, 1998.

[22] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. In *9th Conference on Autonomous Robot Systems and Competitions*, pages 59–65. IPCB, Castelo Branco, Portugal, 2009.

[23] J.-B. Mouret and S. Doncieux. Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary Computation*, 20(1):91–133, 2012.

[24] A. L. Nelson, G. J. Barlow, and L. Doitsidis. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345–370, 2009.

[25] Y. Niv, D. Joel, I. Meilijson, and E. Ruppin. Evolution of reinforcement learning in uncertain environments: A simple explanation for complex foraging behaviors. *Adaptive Behavior*, 10(1):5–24, 2002.

[26] C. Ollion, T. Pinville, and S. Doncieux. With a little help from selection pressures: evolution of memory in robot controllers. In *13th International Conference on the Simulation and Synthesis of Living Systems*, pages 407–414. MIT Press, Cambridge, MA, 2012.

[27] D. S. Olton. Mazes, maps, and memory. *American Psychologist*, 34(7):583, 1979.

[28] S. Risi, C. E. Hughes, and K. O. Stanley. Evolving plastic neural networks with novelty search. *Adaptive Behavior*, 18(6):470–491, 2010.

[29] S. Risi and K. O. Stanley. Indirectly encoding neural plasticity as a pattern of local rules. In *11th International Conference on Simulation of Adaptive Behavior*, volume 6226 of *LCNS*, pages 533–543. Springer, Berlin, Germany, 2010.

[30] F. Silva, P. Urbano, and A. L. Christensen. Adaptation of robot behaviour through online evolution and neuromodulated learning. In *13th*

Ibero-American Conference on Artificial Intelligence, pages 300–309. Springer, Berlin, Germany, 2012.

[31] F. Silva, P. Urbano, and A. L. Christensen. Online evolution of adaptive robot behaviour. *International Journal of Natural Computing Research*, 4(2):59–77, 2014.

[32] F. Silva, P. Urbano, L. Correia, and A. L. Christensen. odNEAT: An algorithm for decentralised online evolution of robotic controllers. *Evolutionary Computation*, 2015. In press.

[33] F. Silva, P. Urbano, S. Oliveira, and A. L. Christensen. odNEAT: An algorithm for distributed online, onboard evolution of robot behaviours. In *13th International Conference on the Simulation and Synthesis of Living Systems*, pages 251–258. MIT Press, Cambridge, MA, 2012.

[34] K. O. Stanley. *Efficient Evolution of Neural Networks through Complexification*. PhD thesis, University of Texas, Austin, TX, 2004.

[35] K. O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162, 2007.

[36] K. O. Stanley, D. D'Ambrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.

[37] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[38] K. O. Stanley and R. Miikkulainen. A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130, 2003.