Neuroevolution Trajectory Networks of the Behaviour Space

Stefano Sarti[0000−0002−1780−2259], Jason Adair[0000−0003−0198−9095], and Gabriela Ochoa[0000−0001−7649−5669]

University of Stirling, Scotland, United Kingdom,
[stefano.sarti, jason.adair, gabriela.ochoa]@stir.ac.uk

Abstract. A network-based modelling technique, search trajectory networks (STNs), has recently helped to understand the dynamics of neuroevolution algorithms such as NEAT. Modelling and visualising variants of NEAT made it possible to analyse the dynamics of search operators. Thus far, this analysis was applied directly to the NEAT genotype space composed of neural network topologies and weights. Here, we extend this work, by illuminating instead the behavioural space, which is available when the evolved neural networks control the behaviour of agents. Recent interest in *behaviour characterisation* highlights the need for divergent search strategies. Quality-diversity and Novelty search are examples of divergent search, but their dynamics are not yet well understood. In this article, we examine the idiosyncrasies of three neuroevolution variants: novelty, random and objective search operating as usual on the genotypic search space, but analysed in the behavioural space. Results show that novelty is a successful divergent search strategy. However, its abilities to produce diverse solutions are not always consistent. Our visual analysis highlights interesting relationships between topological complexity and behavioural diversity which may pave the way for new characterisations and search strategies.

Keywords: Search trajectory networks · Behavioural space · NEAT · Novelty search · Divergent search.

1 Introduction

Hard optimisation problems may require more than following a fitness gradient – this is often due to intrinsic characteristics of deceptive domains. In neuroevolution, search based on fitness alone could become stagnant, due to local optima regions with strong attraction basins. An algorithm offering a solution to this, in the context of neuroevolution, is *Novelty Search*, where a search strategy seeks to discover behaviours which are *novel* and *distant* from those previously explored. Through this, remarkable results were achieved; highlighting that behavioural diversity can enhance exploration to solve difficult neuroevolution tasks.

Sarti and Ochoa [22] showed that modelling and visualising the dynamics of NEAT variants using search trajectory networks [17, 16] can bring new understanding into neuroevolution. Here we propose Neuroevolution Trajectory Networks (NTNs), an adaptation of STNs to the neuroevolution domain, to characterise the dynamics of algorithms in the space of the agent's behaviours. In contrast to previous work [22], where STNs were applied directly on the complex NEAT genomes, here we model the trajectories of the best performing agents in the behaviour space. Given the complexity and redundancy of NEAT genomes, mapping and partitioning the search space as required for STN analysis, results in large network models with sporadic search convergence (frequently visited nodes attracting the search process). The key idea in this article is, instead, to use a much simpler and

smaller space, the behavioural space, which in the context of agents solving 2D mazes is simply the end Cartesian (x, y) coordinates of the simulated agents. Encoding (x, y) coordinates as nodes in NTN models (rather than encoding NEAT genomes), is a much simpler task, already tested for STNs in continuous domains [17, 16], and provides much more compact and thus easy to analyse models.

We aim to visualise and analyse the derived NTN models from solving medium and hard maze problems (as proposed in [7]) using three NEAT search strategies: novelty, objective and random search. The main contributions of this article are to:

- Provide an analysis tool, NTNs, to examine divergent search and behavioural characterisation in neuroevolution
- Assess the relationships between topological structure, behavioural diversity, and divergent search

This paper is organised as follows. Section 2 details related lines of previous work. Section 3 describes our experimental methodology. Section 4 presents the performance and behavioural diversity analysis of the studied search strategies. Section 5 describes the creation and analysis of the NTN models. Finally, section 6 summarises our main findings and suggestions for future work.

2   Related Work

Recent neuroevolution research has focused on divergent optimisation with a specific interest in increasing diversity [20], moving away from the pursuit of objective and convergent optimisation. The rational for this shift in focus may be attributed to observations in evolutionary biology [25] - it is suggested that high diversity and unique niches result in higher fitness and survival qualities.

This has led to considerable research in the identification of mechanisms that allow for high quality and high diversity, often referred to as *quality/diversity* algorithms [9][4][5][1]. One of the first exemplar of these algorithms was proposed by Lehman and Stanley [10][8], where a *novelty* metric was used to force divergent search and increase behavioural diversity. This method was devised to overcome strong regions of local optima in a maze navigation domain with a highly deceiving structure. An extension of this algorithm is *novelty search with local competition* (NSLC) in which the true principals of quality-diversity were incorporated [9]. NSLC uses multi-objective NSGA-II to evaluate solutions, seeking high-quality behavioural diversity.

Behaviour Characterisation. A comprehensive evaluation of this class of algorithms is provided in [20] in which the best performing QD algorithm was in fact NSLC. The authors go on to define *behavioural characterisation* (BC) as a convention by which behaviours are categorised and classified. This classification can either be aligned or unaligned to the notion of *quality* determined by the assessment domain. In their research, nothing in between these two characterisations is examined. In this, the authors find that not all behaviours are equally important but they highlight that a BC aligned to the end goal leads to favorable diversity and high exploitative qualities to reach the domain goal. Although, the counterargument, as seen previously in [10][9][13] is that the evolution of diversity should originate from BC not directly related to the notion of quality, as these non-directed explorations would lead to divergent ways to discover wider varieties of optimal capabilities[12]. In this research, we intend to detect these behavioural characterisations and offer a tool for their evaluation, with particular attention to topological complexities. Topological complexities often dictate the behaviour of an evolved agent. Here, we elucidate what roles, if any, topologies play in BC and diversification.

Search Trajectory Networks (STNs). STNs are data-driven, graph-based models of search trajectories where nodes represent a given state of the search process and edges represent search progression between consecutive states. Once a system is modelled as a graph (network) powerful analytical and visualisation tools from the field of complex networks can be applied [15]. STNs were initially proposed to characterise differential evolution and particle swarm optimisation for several classical continuous optimisation benchmark functions [17]. STN analysis was later extended to cover not only population-based algorithms but also stochastic local search methods, and both continuous and combinatorial optimisation problems [16]. Recently, these models have been applied to the Cyclic Bandwidth Sum problem in [14], which further corroborates the usefulness of this modelling technique and its applicability to different domains.

3   Experimental Methodology

As a benchmark, we use the classic 2D maze navigation domain outlined in [8]. The task involves an agent (robot) controlled by a neural network navigating a maze from a starting point to an end point, for a fixed number of time steps. The agent's physiognomy is presented in figure 1. The agent has six rangefinder sensors that indicate the distance to the nearest obstacle. These are rays (represented as arrows in figure 1) originating from the body of the agent, which detect obstacles that are in close proximity, returning the computed distance to such obstacles. The four pie-slice radar sensors are known as the field of view (FOV) that act as a compass, pointing the agent towards the goal (maze exit point). When the line from the goal location to the centre of the robot falls within these (FOV degrees are specified in figure 1) the specific radar sensor becomes activated. The activation of the sensors are returned as inputs for the maze-navigating agent to compute behaviours and stored to represent the state of the agent at each simulation time steps.

The outputs computed by the ANN represent the actions that the agent can perform. There are two actuators (actions) which relate to forces that either rotate and/or propel the agent's body. These correspond to changes in linear and/or angular velocity.
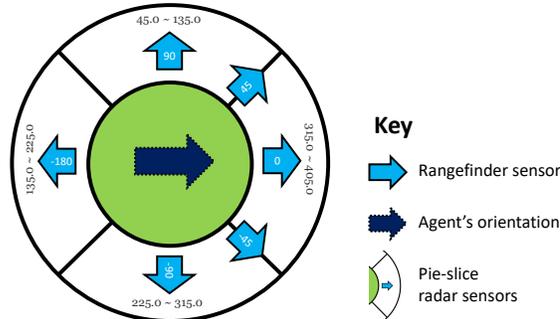
Fig. 1: Physiognomy of the maze navigating agent. The agent has six rangefinder sensors for obstacles detection and four pie-slice radar sensors acting as a compass to detect the goal orientation. Pie-slice labels indicate the degree range of the compass, and arrow labels indicate the rangefinder sensors positions, both in reference to the agent's orientation. Illustration adapted from [19]

The maze domain is relevant for testing novelty search as it has a deceptive fitness landscape. The fitness function used in [8] is formulated on the proximity of the agent to the goal at the end of any given maze navigation simulation. The navigation is made difficult as the maze present walls that form "culs-de-sac". These dead ends that lead close to the goal are local optima to which an objective-based algorithm is likely to converge to. These are especially accentuated in the hard maze map (see figure 2b); traps to the search progress are highlighted in red. We used the two maps designed in [8], described below.

Medium Maze (low deception). Figure 2a shows the map for the medium maze. This configuration is of low to medium difficulty. The map presents areas of low deception that can be circumvented by the agent without major difficulty. The path from the starting point (yellow square) to the goal (green square) is reasonably linear with a lower chance, as compared to the hard map, for the agent to get trapped in between walls.

Hard Maze (high deception). Figure 2b illustrates the hard maze configuration. This map is harder as the placement of the walls generate local minima (red shaded circles) capable of trapping the search progress of agents traversing the maze. These areas of high deception are what most challenges the neuroevolution search strategies.

(a) Medium map.
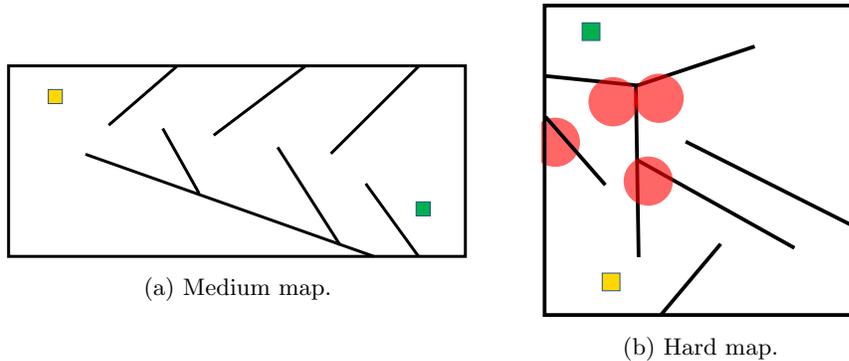
(b) Hard map.

Fig. 2: Maze navigation maps. In both maps, the yellow square represents the starting position of the simulation and the green square represents the goal (exit point). In the hard maze, the landscape local optima are highlighted in red. Image adapted from [19]

3.1   NEAT Variants

Previous work [26][23][24][21][2][22] indicates that the use of crossover in NEAT does not improve the algorithm's performance. Therefore, we deemed the use of crossover unnecessary and possibly counterproductive for this experiment. We use the original NEAT-Python implementation [19], but switch off the recombination operator. The search strategies compared, outlined below, are the same ones used in [8].

Fitness-based Search Standard evolutionary algorithms use a fitness function to guide the search process. In the maze domain, high fitness is achieved based on an agent's proximity to the goal at the end of the navigation task (trial time steps).

$$\mathcal{L} = \sqrt{\sum_{i=1}^{2} \left(a_i - b_i\right)^2} \tag{1}$$

Equation 1 is used to calculate the Euclidean distance between the agent's simulated location with respect to the goal (exit point of the maze). $\mathcal{L}$ represents the specific root-mean-squared error function used for the proximity evaluation. Where a denotes the position of the agent at the end of the simulation and b the location of the maze exit (expressed as 2 dimensional coordinates). The result of this loss function is subtracted from the initial distance of the agent to the end goal and subsequently normalised to produce the fitness value.

Novelty Search This strategy, as the authors describe, is somewhat counter-intuitive [8]; in order to yield successful results in deceptive domains the objective must be abandoned. The idea is to define a function which uses the *novelty* of the agents' behaviours as a metric of performance. Novelty, specific to NEAT, can either be *structural* (novelty of the ANNs topologies) or *behavioural* (novelty of the ANNs explorative behaviours). The goal is to achieve an effective diverging search. That is, a solver that exhibits diverse explorative behaviours, rewarding those behaviours that yield path towards new and unexplored locations of the domain. This way agents should develop unforeseen tactics to escape the maze's basins of attractions, ultimately to identify the goal.

$$\mathrm{dist}(x, \mu) = \frac{1}{n} \sum_{j=n}^{n} |x_j - \mu_j| \tag{2}$$

In this specific scenario, the performance of the neurocontrollers is calculated using the metric of sparseness. To do so, the implementation, derived from [19], similarly to [8], uses the *k-nearest neighbours* algorithm.

In equation 2, $\mathrm{dist}(x, \mu)$ is the novelty score denoting the behavioural difference between two agents, computed as the distance between the two trajectory vectors (one vector per agent; x and μ). Trajectory vectors, which are traced by agents, are comprised of bi-dimensional maze coordinates of size n. x_j and μ_j are the values of the compared vectors (x and μ) at position j. To simplify the calculation, in this implementation, only the agent's trial end coordinates ($j = n$) are considered as the coordinates of interest. This way we can determine the final position of the agent and therefore the distance to the goal.

Random Search Random search is the simplest of the search strategies. It assigns continuous random values derived from a pseudo-random number generator to the evaluated genomes. This strategy is merely used as control to explore the effect of divergent search, and it is expected not to yield good performance.

3.2   Parameters

Table 1 outlines the parameters values used in our experiments, we emulate the values used in [8]. All parameters, with the exception of the solver time steps, are identical for both maze maps.

Similarly, the parameter values are the same for the three algorithm variants. The k parameter in the k-nearest neighbours algorithm, is required for the sparseness calculation, only necessary in novelty search. The solver time steps had to be increased for the hard maze, as for this specific implementation [19] our tests have shown that 400 time steps was not a sufficient allowance to reach the goal in this map.

The coefficients $c1$, $c2$ and $c3$ are all NEAT specific parameters. Standard default values have been used for these. For NTNs modelling on all variants, the partition of the behavioural space is achieved by rounding off to $1e-0$ the x, y spacial coordinates, and the fitness evaluations to $1e-2$.

Table 1: NEAT parameter values used. The k parameter (k-nearest neighbours) is relevant only for the novelty search strategy.

Parameter	Value
Population size	250
Maximum generations	1,000
Solver time steps (medium maze)	400
Solver time steps (hard maze)	600
Solution fitness value	1.00
Fitness threshold	0.95
Bias range	[-30, 30]
Weight range	[-30, 30]
$c1$	1
$c2$	1
$c3$	3
Probability add link	0.1
Probability add node	0.005
k (k-nearest neighbours)	16

4   Results

4.1   Analysis Setup

Performance is measured in terms of evaluations required to reach a solution (efficiency) and the quality of solutions at the end of the evaluation cycle (efficacy). The former also highlights the success rate of each strategy. Metrics are computed over 30 independent runs. Out of these 30 runs for each search strategy, 10 were selected for the NTNs construction and analysis, in order to have models of manageable size, yet producing a comprehensive picture of the spectrum of search behaviours observed (Section 5.1 details how these were selected).

In our experiments we consider a solution to be achieved, if the fitness reaches a value of 1.00. The maximum number of iterations is set to 1,000 which corresponds to approximately 250,000 evaluations in total (with a population of 250 genomes). The number of maze simulator solving steps is set to 400 for the medium maze and 600 for the hard maze as this requires substantially more exploration time to find the solution (see Section 3.2).

4.2   Performance Analysis

It is known from previous work [10][8] that novelty search outperforms both objective and random search on deceptive domains. For completeness, we reproduce some of those results. Figure 4 illustrates the performance of the three search strategies averaged over 30 runs. We clearly observe that both objective and novelty search are able to reach a fitness of 1.0 and discover a behaviour that solves the maze in Figure 3a. On the other hand, random search does not produce successful solvers and significantly under-performs compared to the other variants, as expected. The main observable difference between objective and novelty search is that objective search on average requires more iterations/evaluations to find solvers. This can be gathered from the distribution of evaluations of the 30 runs shown in Figure 3c.

In the hard maze we observe that novelty search is the only strategy capable of finding solving behaviours; however, higher average fitness does not signify a better quality of solution. Differently from [10] which achieved 97% success rate (39 out of 40), in our approach, out of the 30 runs the success rate was only found to be 16.6% (5 out of 30). As we observe in Figure 3d the spread of the evaluations necessary to reach a solution is very wide, showing a rather unstable system.

4.3   Behavioural Diversity Analysis

Amongst the search strategies analysed, it is known that both novelty and random search are divergent strategies, as opposed to convergent strategies such as objective search. From [10][8], it is known that that objective search is subject to deception due to the greediness of its search. Despite being a divergent strategy, random search does not offer valuable solutions to the problems. The only succeeding variant in the highly deceptive problem (hard maze) is novelty. Therefore, this section focus on the finer details of what makes each strategy different. We assess these using metrics available from the NEAT algorithm such as topologies and behaviours. As previously discussed in Section 2, a correct behavioural characterisation as shown in [20] is fundamental for a successful algorithm that explores solutions with the aim of diversifying. BC, as the name suggests, primarily concerns actual behaviours derived from evolved agents. Another way of setting novelty search is to use the phenotypical structures of evolved agents as the diversity characterisation [19]. To the best of our knowledge this type of setting and related examination has not been attempted before.

Topological Complexity and Behavioural Diversity We take into consideration two metrics for the diversity analysis: *topological complexities* and *distinct behaviours*. From the algorithm we can record the topological information in terms of number of neurons and number of connection. As our analysis is aimed at neuroevolutionary dynamics, we only consider aspects of the topologies that are evolved. Therefore, we augment this metric by summing the number of neurons and connections together, to then subtract those *input* and *output* neurons, as they are not evolved but preset parameters. To perform a fair comparative analysis we normalise this derived complexity metric to be in the range (0 - 1]. The examined literature was unable to offer an alternative metric to this, which would be suitable for our experiment. As these shallow neurocontrollers are formed by neurons and connections which are expressed in the same genetic code, we summarise them as a unique value determining topological complexity. This is also outlined in the documentation of the algorithm used [11].

Behavioural diversity is also a metric obtained from the actions of an agent. Several consecutive action will trace a path of locations in the Cartesian space of the maze domain. The dimensionality

(a) Convergence Medium Maze

(b) Convergence Hard Maze

(c) Distribution Medium Maze

(d) Distribution Hard Maze
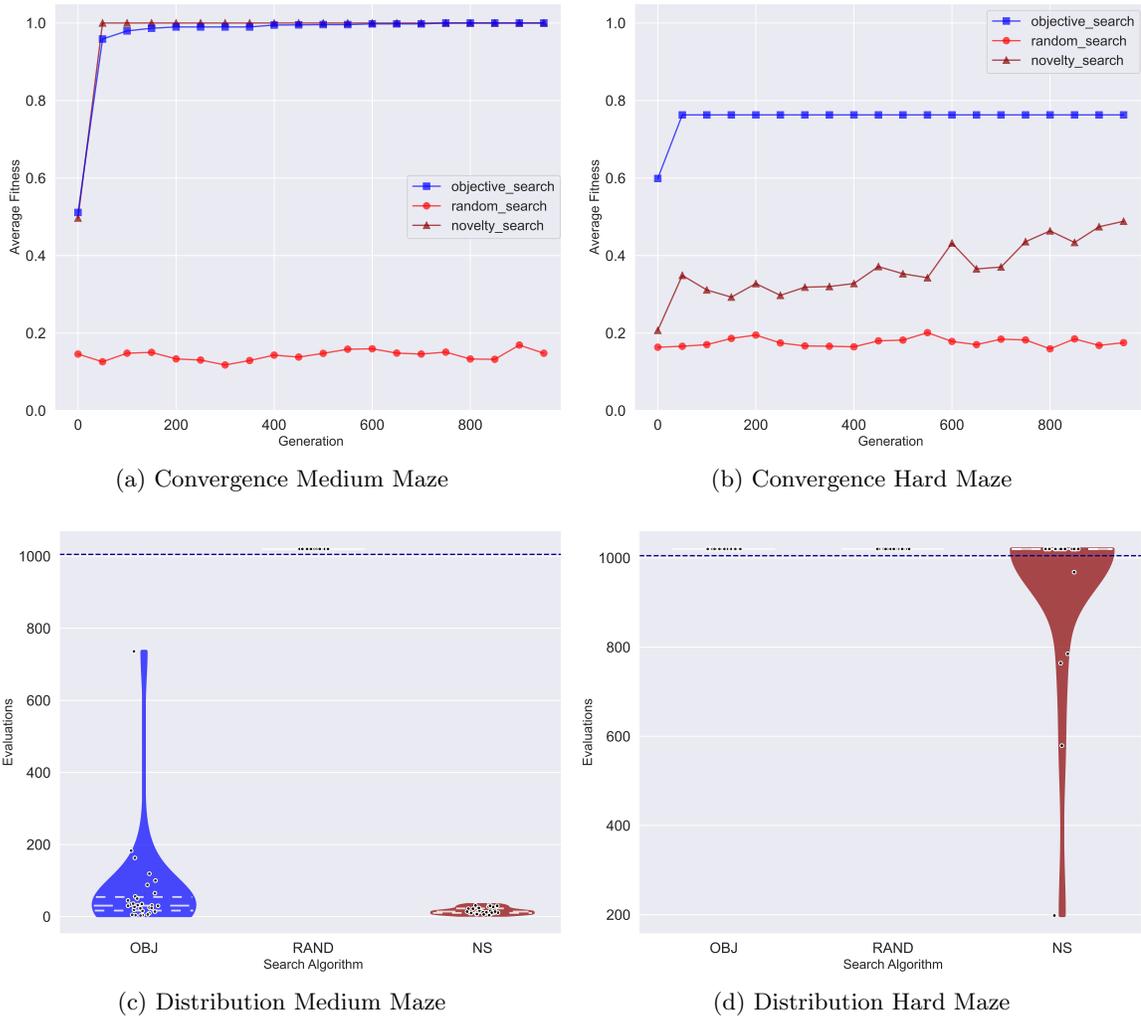
Fig. 3: Convergence plot (top) representing the averaged fitness performance over 30 runs and violin plots (bottom) showing distributions of successful vs failed runs (failed runs shown above the dotted line) for all tested search strategies in the *medium* maze (a, c) and in the *hard* maze (b, d) domains. In the x axis, from the left in (c, d) the search strategies shown are objective, random and novelty respectively.

of this actions vector can vary; therefore, to simplify this, as it is used in this implementation (Section 3.1), we consider a behaviour solely as the ending x and y coordinates. Other behaviour vectors configurations, capable of capturing more information may be possible, but in this approach, we deemed the chosen one to be sufficient for the purpose of this examination.

In order to make sense of these coordinates and to discretise the space of possible behaviours we pre-process these values by reducing their precision. This is a common technique and a requirement of STNs analysis.

(a) Average Topologies Medium Maze

(b) Distinct Behaviours Medium Maze

(c) Average Topologies Hard Maze

(d) Distinct Behaviours Hard Maze
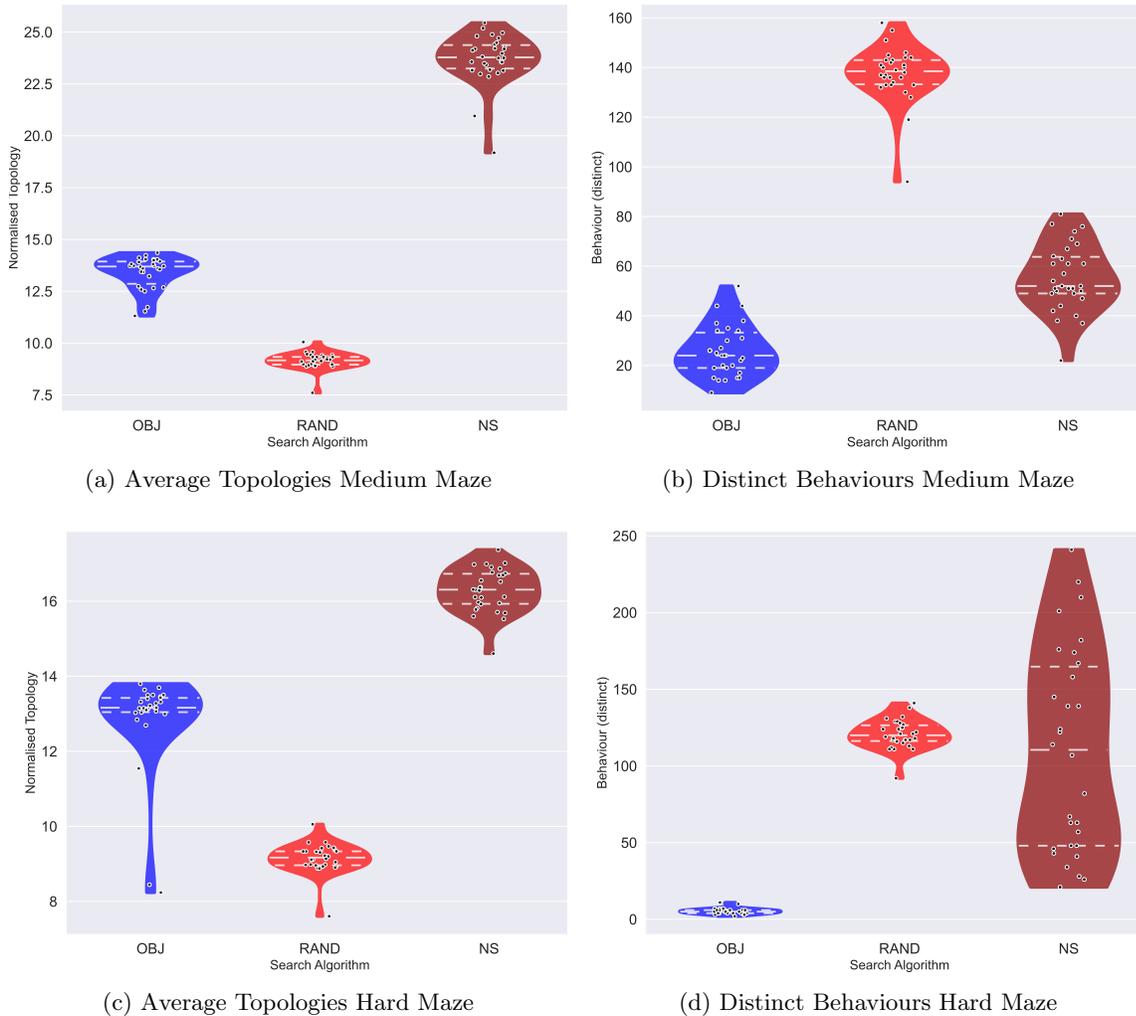
Fig. 4: Violin plots representing the distributions of averaged topologies, as well as distributions of distinct behaviours found in each of the 30 runs for all tested search strategies in the *medium* maze (a, b) and in the *hard* maze (d, c) domains. In the *x* axis, from the left in (c, d) the search strategies shown are objective, random and novelty respectively.

Figure 4b and 4d show the distributions of the normalised topologies for each of the 30 runs for the medium and hard maze respectively. The normalised topology values are derived as an average of all the best topologies from all the iterations for each of the runs. This way we are proposing a

planarisation of time, flattening this dimension, to capture all the topologies seen during the run. Similarly, this practice is done in NTNs modelling. Figure 4b and 4d present the number of distinct behaviours recorded in each run for both mazes respectively. These distributions are a metric to assess the diversity of solution and we argue they provide a sense of how divergent a search strategy can be.

The main insight observable from these plots, which will be used to further validate the NTNs analysis are:

- Novelty search has on average more complex topological structures of the genomes than random and objective search. The spread of topologies in random search is narrow and they are notably less complex.
- Looking at random search, we observe high and consistent diversity values (narrow spread). As simpler topologies appear capable of creating more diverse behaviours.
- Random search is not an effective strategy, due to the poor fitness performance. Judged in terms of diversity, this strategy produces more consistent results.
- Novelty search distribution of distinct behaviours presents high variance. This signals an instability of this strategy and high variation between low and high diversity generation.

5   Neuroevolution Trajectory Networks

NTNs are STNs applied to neuroevolution dynamics. The model definitions of STNs are applicable to NTNs and can be found in [16], we reproduce these below for completeness and extend these to include some exceptions, such as a model variation named compressed NTNs (CNTNs). This compressed model, originally inspired form an idea applied to local optima networks [18], is devised to deal with search spaces that present large amounts of neutrality. Neutrality refers to adjacent portions of the search space with the same fitness. Modelling neutrality is relevant for NEAT, as it is well known that there are many ways to set neural network weights which may instantiate a similar behaviour [25]. Compressing equal edges will provide clearer visualisations.

To define a network model, we need to specify their nodes and edges. The relevant definitions are given below.

Representative Solution. A solution to the problem (in this study, an evolved behaviour) at a given iteration is one that represents the status of the search process in the space of behaviours. Since NEAT is a population-based algorithm, the solution with best fitness in the population at a given iteration is chosen as the representative solution.

Location. A non-empty subset of solutions that results from extracting behaviours as x and y coordinates. Each solution in the search space is mapped to one location. Several similar solutions are generally mapped to the same location, as the locations represent a partition of the behaviour search space.

Search Trajectory. Given a sequence of representative solutions in the order in which they are encountered during evolution. A search trajectory is a sequence, mapping solutions to corresponding location.

Node. A location in a search trajectory. The set of nodes is denoted by N.

Edges. Edges are directed and connect two consecutive locations in the search trajectory. Edges are weighted with the number of times a transition between two given nodes occurred during the NTN modelling process. The set of edges is denoted by E.

Neuroevolution Trajectory Network (NTN). A directed graph NTN $= (N, E)$, with node set N, and edge set E as defined above.

Compressed Node. A node that aggregates a set of connected nodes (a connected component) in the NTN with the same fitness value. The set of compressed nodes, is denoted by CN.

Compressed Edges. After the compression (supra), there are no edges between nodes with the same fitness. The set of edges belonging to the compressed nodes are aggregated and their weights summed. This is the compressed edges set, CE.

Compressed NTN. A directed graph CNTN $= (CN, CE)$, where nodes are the compressed nodes CN and edges the compressed edge set CE.

5.1   Sampling and Model Construction

Models are constructed from a data log derived by running the studied search strategies. Specifically, a list of steps connecting two adjacent representative solutions in the search process is recorded. Solutions (evolved agents in the case of NEAT) are represented by their behavioural signature; their final x and y maze coordinates. Each stored search step is formed by two consecutive representative solutions being linked; these transitions become the edges of the network model. Each representative solution stores other attributes for the analysis; such as fitness value and topological complexity. A log file is populated with steps from multiple selected runs. Following this, a post-processing stage maps solutions to locations, and models the network object as per the definitions above.

To extract the models, 10 out of the 30 independent runs were selected for each search strategy on both maze problems. We ranked the runs and selected top 3, the bottom 3, and 4 intermediate runs in therms of fitness performance. The idea was to select a representative sample of the 30 runs; as generating models for all the runs would have made the plots too complex to extract and visually perceive meaningful features.

Network Metrics and Visualisation Once a system is modelled as a graph or network, a variety of metrics can be be computed; such as the degree distribution, length of paths, community structure, and centrality of nodes to name a few [15]. For simplicity, we selected seven network metrics to bring insight into the behaviour of the search variants studied. These metrics are summarised in Table 2.

Visual assessments enable us to appreciate structural features which may be difficult to infer from the network metrics alone. The commonly used node-edge diagrams that we have deployed assign nodes to points in the 2-dimensional Euclidean space, and connect adjacent nodes by lines. For directed graphs, arrowheads are used to indicate the direction of connections. Vertices are drawn here using basic geometrical shapes. Vertices properties such as size, colour and shapes are used to convey more information. In our approach we decorate the network with a palette gradient to signify the complexity of the evolved topologies, size is the strength of incoming edges. The graph visualisations in this paper were produced with the igraph library [3] of the R programming language.

Table 2: Description of network metrics.

nodes	Number of nodes.	Metric detailing diversity of solutions.
solutions	Number of nodes that reach the desired fitness target.	Number of behaviours achieving the maze goal.
w-edges	Number of worsening edges.	Indicative of the amount of non-greedy exploration.
n-paths	Number of shortest paths from start nodes to solution nodes in the CNTN.	Indicative of the effectiveness and solutions' reachability. (not expressed if solution not identified).
p-length	Average lengths of the shortest paths from start nodes to solution nodes in the CNTN.	Indicative of efficiency of explorations. (not expressed if solution not identified).
complexity	Average topological complexity	Indicative of complexity in relation to divergent and convergent searches respectively.
in-strength	Average strength of vertices' incoming connections	Indicative of the extent to which some nodes attract the search process. (calculated as a normalised average)

For the plots we used a *force-directed* layout algorithms [6], which strive to satisfy some generally accepted criteria, such as distributing the nodes evenly on the plane, minimising the number of edges crossing and keeping edges lengths approximately uniform.

5.2   Results and Discussion

Figure 5 shows the CNTNs for the two maze maps. Each individual figure depicts the 10 selected runs of each search strategy. These illustrations and their relative metrics of table 3 helped to reveal the following main observations.

- The networks of objective search highlight the greedy, convergent behaviour of this search strategy (see *w-edges* in metrics table 3). In the medium maze the best solution is easily achieved with some convergence to common behaviours. In the hard maze, the behaviour search process rapidly gets attracted to a common sub-optimal behaviour. A range of simple to complex topologies reach the goal in the medium maze. In the hard maze, all agents that get stuck in a local optimum are of high topological complexities (dark red).
- The networks of random search are, in both domains, maintaining levels of high diversity, with attracting behaviours. In both cases, with this search strategy there seems to be common attracting behaviours at the end of the trajectories. The large value in *w-edges* of the metrics Table 3 as well as *in-strength* are visible in the network and appear as an oscillating behaviour between improving and worsening behaviours. This highlights the truly random, yet diverse characteristics of this divergent search. Here the agents' complexities appears to be mid to low (blue).
- The networks of novelty search in the medium maze present very similar levels of node convergence, same locations are re-visited several times close in the trajectories, generating a higher *w-edges* value. Paths to solution are similar in terms of average length (*p-length*) but novelty being a divergent search, can be slightly more explorative. Due to this nature we can also observe similar levels of topological complexities as random search.

– An interesting feature can be discerned from this visualised model (figure 5f). In the centre, a distinctive rim is drawn, composed of nodes that have similar size (convergence), the nodes inside this circle present lesser diversity (high convergence values) and they appear to be more topologically complex. The nodes outside this circle visually appear simpler, more diverse, and are capable of greater exploration towards reaching the solving behaviour.

– We assume that this conformation (supra) signifies a breakthrough from higher complexity, low diversity of nodes attracted by local minima, to simpler networks, leading to higher diversity, leading to more successful explorations.

Table 3 shows the values of the network metrics described in Table 2 for the two mazes and the 3 search strategies.

Table 3: Network metrics.

	Medium Maze			Hard Maze		
	Objective	Random	Novelty	Objective	Random	Novelty
nodes	66	653	67	25	536	701
solutions	1	0	1	0	0	1
w-edges	0	0.460	0.293	0	0.421	0.456
n-path	10	0	10	0	0	9
p-length	5.7	N/A	5.9	N/A	N/A	8.555
complexity	0.360	0.155	0.177	0.339	0.135	0.226
in-strength	0.015	0.015	0.016	0.070	0.022	0.004

As behaviours in this case are characterised by Cartesian x and y coordinates, in Figure 6 we propose an illustration of the neuroevolution trajectories in both geographical representation of the mazes. For simplicity we only represent the best run found for each search strategy. This will provide a visual appreciation of the diversity of each variant and the behavioural optimisation and topological diversity necessary to fully explore and achieve each domain's goal. In this instances, the size of the nodes are a proxy for the normalised topological complexity of each solution.

From these visualisation we can appreciate the following salient features:

– Looking at the behaviours of random search and their topologies, we are able to further corroborate that simpler topologies present higher explorative capabilities as they diverge away from the starting point of the maze.

– In novelty search, specifically on the hard maze, the breakthrough seen in the NTN model of Figure 5f, is somehow resembling the changes in topological complexities happening from the maze's diagonal upward channel. When topologies diverge from this deceptive area they appear to become topologically less complex.

(a) Objective, medium map

(b) Objective, hard map

(c) Random, medium map

(d) Random, hard map

(e) Novelty, medium map

(f) Novelty, hard map
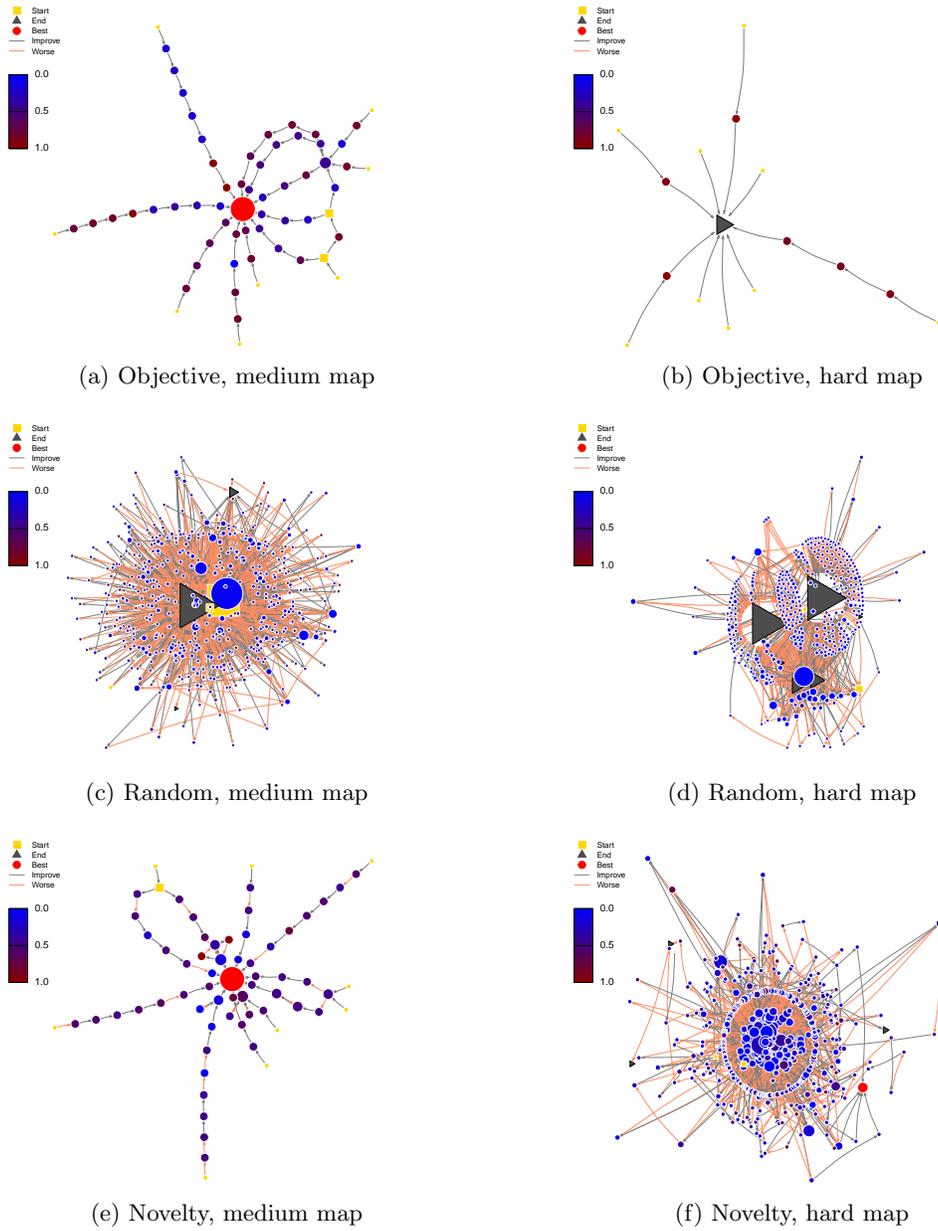
Fig. 5: CNTNs for objective, random and novelty search on the two maze maps. Each sub-figure presents the network model composed of 10 runs for a single search strategy. The size of nodes is proportional to the number of locations in the compressed nodes. Nodes colors are a gradient that signifies the complexity of the network from (0-1], and the shapes specify key search stages.

(a) NTNs, medium map

(b) NTNs, hard map
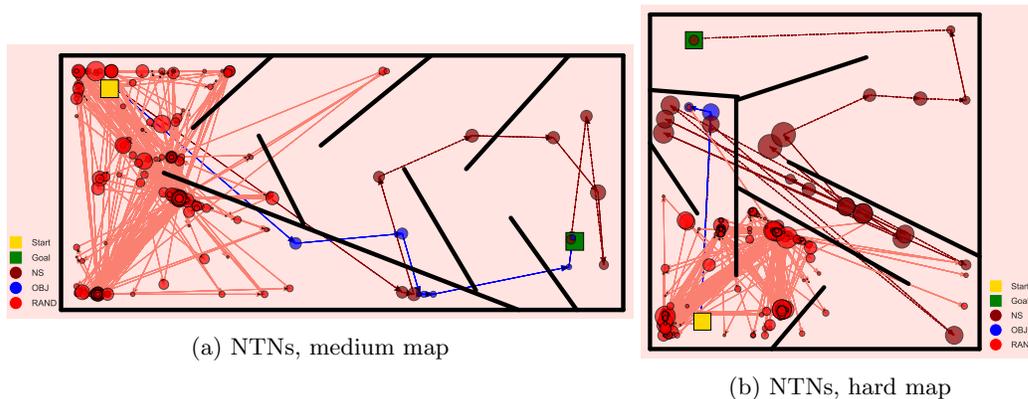
Fig. 6: Visualisation of the best NTN for each search strategy in the medium and hard maze respectively. Size of node signifies topological complexity.

6   Conclusion and Future Work

We propose a visual assessment of the behavioural characterisation of three search strategies deployed in NEAT, tested in medium to highly deceptive maze domains. We adapted an emerging visualisation technique STNs, to the neuroevolution realm: NTNs. We used NTNs to model the evolutionary search process as seen in the behavioural space. Our analysis highlights important characteristics and relationships between behavioural diversity and topological complexities. Although random search is a divergent strategy that does not produce good results, using it in our analysis helped to highlight salient characteristics. Amongst others: topologies that are less complex tend to create increasingly distinct and diverse behaviours with traits of high exploration capabilities, which can eventually lead to successful solutions. This phenomena was also visible by examining the conformation of the best NTNs of novelty search in the hard maze domain geographical representation.

Our examination provides the inceptive insight that alternatives between a closely aligned and non-aligned BC may exist. *Transitive BC* may be an appropriate way to describe this. That is, the focus of the search strategy should be placed on a BC that is indirectly related to divergence and exploration, in the pursuit of quality-diversity. We propose that this indirect search mechanism could further enhance the performance of divergent search strategies. Future work will analyse this claim of *transitive behaviour characterisation* by deploying it as a search technique and testing its validity, while extending its generality by exploiting different domains.

References

1. Chatzilygeroudis, K., Cully, A., Vassiliades, V., Mouret, J.B.: Quality-diversity optimization: a novel branch of stochastic optimization. In: Black Box Optimization, Machine Learning, and No-Free Lunch Theorems, pp. 109–135. Springer (2021)
2. Costa, V., Lourenço, N., Machado, P.: Coevolution of generative adversarial networks. In: International Conference on the Applications of Evolutionary Computation (Part of EvoStar). pp. 473–487. Springer (2019)

3. Csardi, G., Nepusz, T.: The igraph software package for complex network research. InterJournal **Complex Systems**, 1695 (2006)
4. Cully, A., Demiris, Y.: Quality and diversity optimization: A unifying modular framework. IEEE Transactions on Evolutionary Computation **22**(2), 245–259 (2018)
5. Doncieux, S., Laflaquière, A., Coninx, A.: Novelty search: a theoretical perspective. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 99–106 (2019)
6. Kamada, T., Kawai, S.: An Algorithm for Drawing General Undirected Graphs. Information. Processing Letters **31**, 7–15 (1989)
7. Lehman, J., Stanley, K.O.: Exploiting open-endedness to solve problems through the search for novelty. pp. 329–336. No. Alife Xi (2008)
8. Lehman, J., Stanley, K.O.: Abandoning objectives: Evolution through the search for novelty alone. Evolutionary Computation **19**(2), 189–222 (2011). https://doi.org/10.1162/EVCO_a_00025
9. Lehman, J., Stanley, K.O.: Evolving a diversity of virtual creatures through novelty search and local comp. pp. 211–218. No. Gecco (2011)
10. Lehman, J., Stanley, K.O., et al.: Exploiting open-endedness to solve problems through the search for novelty. (2008)
11. McIntyre, A., Kallada, M., Miguel, C.G., da Silva, C.F.: neat-python. `https://github.com/CodeReclaimers/neat-python`
12. Meyerson, E., Lehman, J., Miikkulainen, R.: Learning behavior characterizations for novelty search. pp. 149–156 (2016). https://doi.org/10.1145/2908812.2908929
13. Mouret, J.B., Doncieux, S.: Encouraging behavioral diversity in evolutionary robotics: An empirical study. Evolutionary computation **20**(1), 91–133 (2012)
14. Narvaez-Teran, V., Ochoa, G., Rodriguez-Tello, E.: Search Trajectory Networks Applied to the Cyclic Bandwidth Sum Problem. IEEE Access **9**, 1–1 (2021). https://doi.org/10.1109/access.2021.3126015
15. Newman, M.E.J.: Networks: an introduction. Oxford University Press, Oxford; New York (2010)
16. Ochoa, G., Malan, K.M., Blum, C.: Search trajectory networks: A tool for analysing and visualising the behaviour of metaheuristics. Applied Soft Computing **109**, 107492 (2021). https://doi.org/https://doi.org/10.1016/j.asoc.2021.107492
17. Ochoa, G., Malan, K.M., Blum, C.: Search trajectory networks of population-based algorithms in continuous spaces. In: Applications of Evolutionary Computation, EvoApps. Lecture Notes in Computer Science, vol. 12104, pp. 70–85. Springer (2020)
18. Ochoa, G., Veerapen, N., Daolio, F., Tomassini, M.: Understanding phase transitions with local optima networks: Number partitioning as a case study. In: Evolutionary Computation in Combinatorial Optimization, EvoCOP. Lecture Notes in Computer Science, vol. 10197, pp. 233–248 (2017). https://doi.org/10.1007/978-3-319-55453-2_16
19. Omelianenko, I.: Hands-On Neuroevolution with Python. Packt Publishing, Limited (2019)
20. Pugh, J.K., Soros, L.B., Stanley, K.O.: Quality diversity: A new frontier for evolutionary computation. Frontiers Robotics AI **3**(JUL), 1–17 (2016). https://doi.org/10.3389/frobt.2016.00040
21. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q.V., Kurakin, A.: Large-scale evolution of image classifiers. In: International Conference on Machine Learning. pp. 2902–2911. PMLR (2017)
22. Sarti, S., Ochoa, G.: A NEAT visualisation of neuroevolution trajectories. In: Applications of Evolutionary Computation. Lecture Notes in Computer Science, vol. 12694, pp. 714–728. Springer (2021). https://doi.org/10.1007/978-3-030-72699-7_45
23. Siebel, N.T., Sommer, G.: Evolutionary reinforcement learning of artificial neural networks. International Journal of Hybrid Intelligent Systems **4**(3), 171–183 (2007)
24. Silva, F., Correia, L., Christensen, A.L.: Evolutionary online behaviour learning and adaptation in real robots. Royal Society Open Science **4**(7), 160938 (Jul 2017). https://doi.org/10.1098/rsos.160938
25. Stanley, K.O., Clune, J., Lehman, J., Miikkulainen, R.: Designing neural networks through neuroevolution. Nature Machine Intelligence **2**, 24–35 (2019)
26. Stanley, K.O., Miikkulainen, R.: Competitive coevolution through evolutionary complexification. Journal of artificial intelligence research **21**, 63–100 (2004)