Evolving Generalised Maze Solvers

David Shorten and Geoff Nitschke

Department of Computer Science, University of Cape Town,
7700 Rondebosch, Cape Town, South Africa
{dshorten,gnitschke}@uct.ac.za
http://cs.uct.ac.za

Abstract. This paper presents a study of the efficacy of comparative controller design methods that aim to produce generalised problem solving behaviours. In this case study, the goal was to use neuro-evolution to evolve generalised maze solving behaviours. That is, evolved robot controllers that solve a broad range of mazes. To address this goal, this study compares *objective*, *non-objective* and *hybrid* approaches to direct the search of a neuro-evolution controller design method. The objective based approach was a fitness function, the non-objective based approach was novelty search, and the hybrid approach was a combination of both. Results indicate that, compared to the fitness function, the hybrid and novelty search evolve significantly more maze solving behaviours that generalise to larger and more difficult maze sets. Thus this research provides empirical evidence supporting novelty and hybrid novelty-objective search as approaches for potentially evolving generalised problem solvers.

Keywords: Neuro-Evolution, Evolutionary Robotics, Novelty Search, Maze Solving

1 Introduction

A long time goal of *Artificial Intelligence* (AI) is to produce artificial brains capable of eliciting generalised problem solving behaviours equilvalent to those observed in nature [1]. Some research has focused on controller design methods which specifically aim to be general problem solvers across a broad range of task domains [2], [3], [4], [5], [6]. However, an alternate approach is to demonstrate the efficacy of existing controller design methods as a generalised problem solvers in a given task, and then extract the method's underlying principles in order that the method is applicable to a broad range of task domains.

Given that *Neuro-Evolution* (NE) [7] aims to emulate the evolutionary process that has produced generalised problem solvers in nature, NE is one such promising approach. That is, biological neural networks have evolved to be capable of learning a vast range of behaviors to potentially solve any task an organism may encounter in its natural environment [8]. Although the methods used in NE are vast simplifications of the processes which occurred in nature, a resemblance does exist.

This study's research objective is to test if NE is an appropriate controller design method for evolving generalised problem solving behaviours, and to elucidate the necessary defining features of such controller design methods. As an initial step towards addressing this general objective, this study tests the efficacy of an NE controller design method with a *fitness function* [9], *novelty search* [10], and a *novelty-fitness* hybrid for evolving generalised maze solvers. That is, evolved maze solving behaviours that can solve any given maze.

Even though various NE controller design methods are frequently only tested on *specific* tasks, we hypothesize that only small changes are required in order that such methods evolve *general* problem solving behaviours. That is, evolved behaviours that are applicable across a range of task domains. In related NE research, a *specific* task can be viewed as one in which an NE method evolves a controller that solves a single instance of a fully deterministic task. Thus, if a given *Artificial Neural Network* (ANN) controller is evaluated multiple times on this task it will always follow an identical trajectory through the task's state space. This implies that such tasks are solvable by a controller that *memorizes* a specific sequence of sensory-motor couplings, rather than meaningfully interpreting sensory inputs and appropriately mapping them to motor outputs. Examples of such specific tasks are pole-balancing [11], [12], navigation of a single maze [10], [13], [14] and biped [10] and quadruped [15] gait evolution.

A highly specific task can be made *general* either by making the environment stochastic or requiring that a controller is solve multiple instances of the task, each of which differ in some manner. Examples of stochastic environments in the NE literature are abstracted *Markov Decision Processes* [16] and GO playing against a non-deterministic opponent [17]. An example task domain with multiple instances is the multi-agent pursuit-evasion task with variable agent starting positions [18], [19]. A particularly relevant study was the evolution of ANN controllers for generalised helicopter control [20] using both stochastic environments and multiple task instances. However, with the exception of notable research such as that of Rajagopalan *et al.* [21], finding the evolution of generalised problem solvers was positively correlated with connection density in ANNs, there is a lack of research on how NE can be scaled to more general tasks.

Recent research established the evolutionary robotics task of evolving maze solving controllers as a useful controller evolution benchmark [10], [13], [14]. However, all previous work has focused on controller evolution to solve a single maze. In this research, the task was made general via requiring that evolved controllers be able to solve *any* given maze. Such generalised maze solving controllers are evolvable using novelty search or a hybrid fitness-novelty search to direct controller evolution. In comparison, fitness function directed controller evolution performed significantly worse in evolving generalised maze solvers.

2 Methods

2.1 Novelty Search

Traditionally, evolutionary algorithms have been driven by a fitness function [9]. This function usually indicates how far a phenotype (solution) is from a user defined objective. The closer the phenotype is to the objective, the more likely it is that the associated genotype will be selected for reproduction. *Novelty search* (NS) [10] represents a radical departure from this paradigm, given that NS does not explicitly define an objective but rather rewards evolved phenotypes based purely on their novelty. That is, a genotype is more likely to be selected for reproduction given that the genotype's encoded behaviour (phenotype) is sufficiently different from all the other phenotypes produced thus far in the evolutionary run. A criticism of NS is that it is equilvalent to random search [13]. However, recent experimental results indicated that controllers evolved with a NS metric attained some degree of generality. That is, controllers evolved to solve one maze could be successfully transferred to solve different mazes [13]. However, the most convincing proof of NS efficacy is that in comparison to objective driven NE, it produces significant performance improvements in a range of tasks that include maze-solving, evolving bipedal robotic gaits [10], evolving programs with genetic programming [22] and grammatical evolution [23].

To elicit further performance gains in these tasks, various research has tested hybrid NS and fitness metrics. These include using a fitness function combining traditional fitness and a novelty metric [24], restarting converged evolutionary runs using novelty [24], a minimal criteria (for survival and reproduction of controller behaviours) novelty search [25], a progressive minimal criteria (incrementing the requirements for reproduction throughout the evolutionary process) [26], and novelty search combined with speciation techniques [27]. Inden *et al.* [27] found that NS was outperformed by a hybrid objective-novelty metric in pole-balancing, maze solving and quadruped gait evolution tasks. Similarly, Lehman and Stanley [25] found that their minimal criteria novelty search evolved solutions more consistently than objective base search. Gomes *et al.* [26] found that their progressive minimal criteria novelty metric outperformed pure NS in a swarm robotics task. However, it has also been found that an objective based search can outperform NS on the deceptive *tartarus* task [24] as well as pole balancing and a visual discrimination task [27].

This raises the question as to what the defining features of a task, controller design method, and environment are such that NS, or a hybrid novelty-objective metric is able to out-perform objective based search. Lehman and Stanley [10] have argued that if a task's fitness landscape is characterized by low fitness regions being necessary stepping stones for evolution to reach desired high fitness regions, then NS will perform well. Lehman and Stanley [10] also propose that if a domain is *deceptive* then NS will perform particularly well. However, aside from the *hard maze* example [10] the exact defining features of a deceptive task remains unclear. Alternatively, Kistemaker and Whiteson [28] propose that the success of NS is dependant on whether differences in evolved controller behaviours are reflected in differences in the fitness of the controllers' descendants.

A key question is how NS performs in tasks with huge solution spaces, where there is a high degree of probability that the continued discovery of novel solutions will not produce a desired solution within a reasonable amount of time. This research question was tested via applying NS to maze solving with some of the outer walls of a maze removed [10]. Results indicated that the performance of NS degraded to be comparable to objective based search. However, the performance of NS in this version of the task could likely be increased via imposing heuristic constraints that bias evolved behaviours.

Thus, this research investigates the performance of NS in the maze solving domain for a range of large and structurally diverse mazes. In this work, the maze solving behaviours of a simulated robot was evolved for 100 different mazes. This allowed for the evolution of a diverse range of novel maze solving behaviours. The behavioural diversity metric used in this research was such that a robot A, can behave identically to another robot B, on ninety-nine mazes, but by differing on only one maze enables it to distinguish itself as being different from B. This is different to the concept of *increasing the dimensionality of the behaviour representation* [10], where a robot's behaviour is frequently sampled during a given task evaluation. This is due to the fact that, during a single task trial, a robot's behaviour at a given simulation iteration is dependant upon its behaviour at an earlier iteration, and a robot's movements early on will affect the probability of it solving the task (finding a path through the maze) at a later point. However, behaving in a certain manner in one maze does not affect the robot's behaviour in another maze.

Any implementation of NS requires that genotypes have a novelty representation, which is typically a vector of floating-point values. In addition to this, a behavioural diversity metric is required which will assign a novelty value, which is analogous to a fitness value, to any given genotype and its corresponding phenotype. An often used metric, also used in this research, is that of *sparseness*, shown in equation 1 [10].

$$\rho(x) = \frac{1}{k}\sum_{i=0}^{k} \mathrm{dist}(\mu_i, x) \tag{1}$$

Here, μ_i is in the k nearest neighbours of x in both the population and an archive of previously seen genotypes and *dist* is a distance measure.

2.2 Neuro-Evolution of Augmenting Topologies (NEAT)

This study uses the *Neuro-Evolution of Augmenting Topologies* (NEAT) method [29]. NEAT is an established NE method that was selected since it has been previously employed in similar studies [10], [13], [14]. NEAT evolves both the topology and the weights of ANNs via a process of complexification. That is, at the start of artificial evolution, ANNs in the population are functionally simple, with minimal numbers of nodes and connections. During the course of evolution,

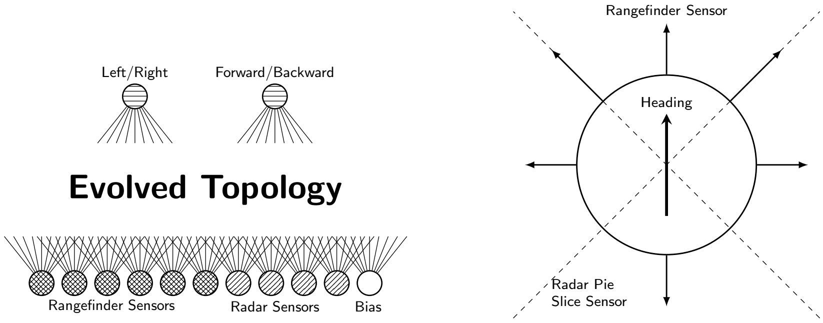

Fig. 1. (Left) ANN controller used in the experiments. (Right) Sensory configuration of robots in the simulated maze task. Both figures adapted from [10].

further nodes and connections are added to ANNs, where increasing the number of nodes and connections in an ANN increases the search space dimensionality. An advantage of NEAT is that this complexifying process is likely to find a solution in a lower dimension search space than the large network which would have to be specified *a priori* if a fixed topology method were to be able to solve a variety of problem types [29]. Other distinguishing features of NEAT are speciation, which protects innovation, and historical markings, which aid in the crossover of structurally different ANNs. In this study, the real-time version of NEAT, rtNEAT [30] was used as it has been demonstrated as effective in related task domains.

Robot ANN controllers were evolved with the goal of being able to solve any perfect maze generated on a grid structure [31], where each grid cell was of a pre-set size. Although evolved on a 13x13 grid, the goal was to produce generalised maze solvers. Hence, evolved controllers were tested on *harder* mazes in a validation set. Figures 2 and 3 present examples of the mazes used for the evolution and validation of maze solving behaviours.

2.3 Maze Generation

The evolution and validation of maze solving behaviours required large maze sets. These sets were produced automatically using the *Daedalus* software written by Walter Pullen [32]. All mazes were perfect mazes generated on a grid structure [31]. Large quantities of mazes were generated using this software's implementation of the randomized Prim's algorithm [33] and then scripts were run to remove the duplicates.

2.4 Generalised Maze-Solvers

The methods are similar to those used by Lehman and Stanley [10], the key difference being the inclusion of a hybrid novelty-objective metric, and evaluating controllers over a set of mazes, rather than a single maze. The parameters for NEAT and NS were also similar to those used by Lehman and Stanley [10].

Experiments were implemented as an extension of *Novelty Search C++* used by Lehman and Stanley [10]. The large number of maze navigation simulations which had to be conducted per robot controller (that is, for each genotype in the population) necessitated that we parallelize the genotype evaluation process. This was done using the Boost MPI library [34] to facilitate parallel processing on clusters. Robots were equipped with six rangefinder sensors and four radar sensors in the configuration presented in figure 2.4. The rangefinder sensors indicated the distance to the nearest wall along a line radiating out from the centre of the robot at a specific angle. The radar sensors divided the space around the robot into four equally sized quadrants and indicated whether or not the goal was in the quadrant. The main difference between the approach presented here and other investigations of NS in the maze domain was that instead of each genotype evaluation equating to one task trial, that is, one attempt to navigate a single maze, each controller was required to navigate every maze in a set of 100. The purpose of making a task trial consist of a set of 100 mazes, was to gauge a controller's general maze solving behaviour.

Robots were given 8000 time steps to navigate any given maze, where as 800 were used in the work of Lehman and Stanley [10]. This increase is due to the larger size of mazes used in this study and also generalised maze solving requires sufficient exploration of the maze, as opposed to simply finding the shortest path. A robot's behaviour representation was a vector of floating point numbers which consisted of the x and y coordinates of the robot every 2000 time steps in each of its 100 maze solving simulations. Thus, each robot's behavioural representation was a vector of 800 floating point numbers. The novelty metric used was that of sparseness, as shown in equation 1. The distance metric between vectors was simply the average difference between corresponding elements.

Also, the simulated robot was changed from a wheeled robot with momentum to a tracked one without any momentum. That is, every time step, a robot's speed and angular velocity were specified by equations 2 and 3, respectively.

$$s = (o_1) * 0.5 \tag{2}$$

$$\omega = (o_2 - 0.5) * 10.0 \tag{3}$$

Here s represents the speed of the robot, ω represents its angular velocity and o_1 and o_2 represent the ANN outputs, in the range [0, 1]. The robot can only move forwards, since a wall following behaviour (a typical maze solving behaviour [32]) requires that the robot be able to move in only one direction. However, exploratory experimentation (results not presented here), showed that allowing the robot to move in reverse had a minimal impact on task performance.

The collision radius of the robot with the walls was reduced from four to 0.5 units. The purpose of these changes in the robot's movement and its collision radius was the result of preliminary experiments finding that the maze environment of Lehman and Stanley [10] did not allow for the evolution of generalised solvers. Exploratory experiments attempted to elucidate the exact relationship

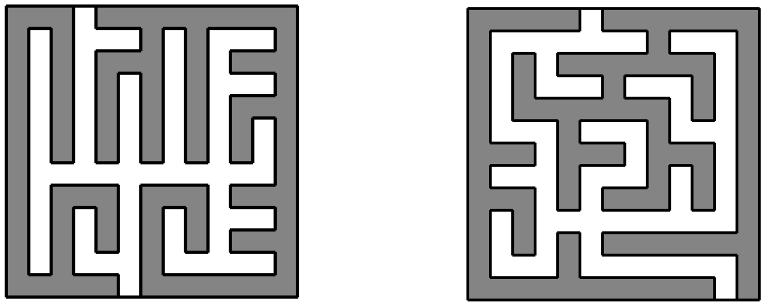

Fig. 2. Examples of the mazes used in the training set. (Left) Less difficult and deceptive. (Right) More difficult and deceptive.

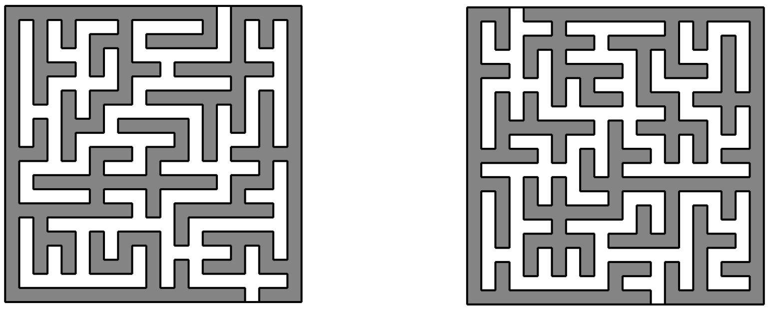

Fig. 3. Examples of the mazes used in the validation set. (Left) Less difficult and deceptive. (Right) More difficult and deceptive.

between these parameters, the defining maze features, and successful evolution of maze solving behaviours. However, due to the large computational and time expense of evolving maze solving behaviours across a vast range of mazes for various task and method parameters, this is still the subject of ongoing research.

3 Experiments

3.1 Objective versus Non-Objective Search

Experiments compared the task performance of non-objective (NS) versus objective versus a hybrid NS-Objective search approach. Thus the three test cases were NE directed by novelty, NE directed by fitness, where robot behaviour fitness was the number of mazes it was able to solve, and finally NE directed by a hybrid fitness-novelty function. This hybrid fitness-novelty function was $f = n + \frac{m}{2}$. Here n is the robot's novelty score and m is the number of mazes that it solved. Exploratory experiments indicated that using this hybrid fitness function, the behaviour of NE was distinct from NE directed by NS and NE

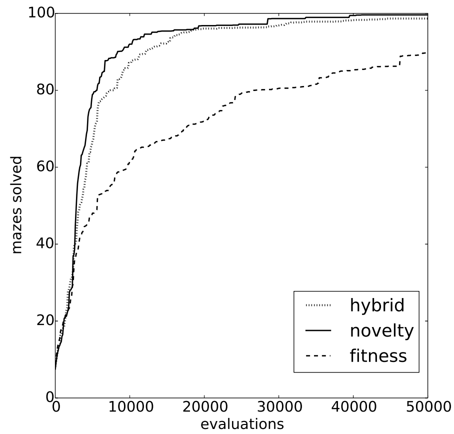

Fig. 4. Number of mazes solved by the best evolved controller, averaged over 20 runs, against the number of evaluations.

	training		generalisation	
	Mazes Solved	Successful Runs	Mazes Solved	Successful Agents
Hybrid	98.7 (4.1)	18	9911 (267)	10
Novelty	99.6 (1.2)	17	9954 (88)	9
Fitness	89.8 (17.6)	10	9735 (419)	1

training			
	Hybrid	Novelty	Fitness
Hybrid	-	0.4	0.009
Novelty	0.4	-	0.007
Fitness	0.009	0.007	-

generalisation			
	Hybrid	Novelty	Fitness
Hybrid	-	0.5	0.02
Novelty	0.5	-	0.03
Fitness	0.02	0.03	-

Table 1. (Top) Average number of training mazes solved by best evolved robot controller after 50000 evaluations, over 20 run,s as well as the number of runs which produced a controller capable of solving all 100 mazes. Also shown is the average number of mazes solved in the generalisation test as well as the number of tested agents capable of solving all 10000 mazes. Standard deviations are in parentheses. (Bottom) p values for the differences in the means shown in the upper table (Mann-Whitney-U test[35]).

directed by fitness. More specifically, the populations converged on solving behaviours, unlike NS, yet evolution performed better than fitness.

Each experiment tested one of these three approaches, and each experiment was run 20 times. Each run was ended after 50000 new individuals had been added to the population. That is, given that rtNEAT was used [30], there was continuous replacement of genotypes. Also, each genotype (robot) was tested on all 100 mazes in the training set, where each run consisted of over five million maze navigation simulations. Results are presented in figure 4 and table 1. It was found that the NS and hybrid NS-objective schemes yielded statistically comparable task performance, where as both approaches out-performed the fitness function directed NE (section 4).

3.2 Validation of General Maze Solving

One test of behavioural generality, is to place evolved controllers (behaviours) in new environments in which they were not evolved and to measure their task performance in these new environments. The 100 mazes used for maze solving behavioural evolution were a sample of the space of all mazes for the given maze size and structural criteria.

To test the general nature of evolved behaviours, the first behaviour (genotype) capable of solving all 100 mazes in the evolution set for each of the 20 runs, under each of the different incentives, was saved.

Since some evolutionary runs did not evolve maze solving behaviours, 17 saved genotypes from NS directed NE, 18 from NS-Objective and 10 from fitness were tested. A set of 10000 mazes was constructed using the same methods as for the evolution set except that these mazes were *harder*, constructed on a 21x21 grid instead of a 13x13 grid. Figures 2 and 3 present examples of some harder mazes. An agent which is a perfect wall-follower will not find these larger mazes more difficult. However, the larger distance which an agent is required to cover in one of these mazes means that there are more opportunities for mistakes to be made.

Nine of the behaviours evolved under NS were able to solve all 10000 mazes. The mean number solved was 9954, with a standard deviation of 88 and a minimum of 9662. Ten of the behaviours evolved under the NS-Objective hybrid were able to solve all 10000 mazes. The mean number solved was 9911, with a standard deviation of 264 and a minimum of 8815. Only one of the behaviours evolved under pure fitness were able to solve all 10000 mazes. The mean number solved was 9735, with a standard deviation of 419 and a minimum of 8799.

4 Discussion

Experimental results (figure 4) indicate that given the three schemes for directing the evolution of maze solving behaviours, NS, NS-Objective hybrid and a fitness function, there was no statistically significant difference between the average task performance of NS and NS-Objective directed NE. However, there was a statistically significant difference between these approaches and fitness function directed NE (table 1). This result contributes to increasing empirical evidence on the value of NS or an an NS-Objective hybrid in tasks with varying degrees of deception versus a purely objective function based search [25], [26], [27], [10], [13]. For example, related work has similarly yielded comparable task performances for evolved maze solving behaviors between a range of hybrid NS-Objective functions and NS search [27]. Moreover, the results presented here show that NS and NS-Objective hybrids can be successful in very large behaviour spaces.

This experimental comparison was not explicitly designed to support the efficacy of NS or an NS hybrid search in contrast to objective based approaches, but rather to elucidate what constitutes an effective NE method capable of evolving generalised problem solvers (in this case study, specifically, maze solving behaviours). Further to this research goal, an analysis of evolved behaviours indicates that multiplicative factors converting controller outputs to robot speed and angular velocity, are important contributors to the functionality of the best evolved maze solving behaviours. However, the impact of controller parameters and task environment features on the evolution of behaviours able to generalise to harder task versions, is the subject of ongoing research.

In terms of the functionality of all evolved behaviours, emergent wall following behaviours were observed in all general maze solving behaviours evolved using NS, NS-Objective and fitness function directed NE, supporting the notion that a well established general maze solving behaviour [32] is attainable by NE. Also, the capability of many of the highest performing behaviours (those that solved all 100 mazes in the initial set), to solve all 10000 of a set of *harder* validation mazes, further supports the efficacy of NE for producing generalised maze solvers. These results support the study's research objective of using NE to evolve generalised problem solvers. The evolution of generalised maze solvers is an initial step towards this objective, where comparatively testing NS, NS-Objective hybrid, and objective based search was necessary to help elucidate the defining features of an NE controller design method able to evolve generalised problem solving behaviours.

The generality test was to validate evolved behaviours in a *harder* maze set. A majority of the best behaviours evolved by NS, NS-Objective and fitness function directed NE were able to solve all 10000 mazes in the validation set. However, the NS and NS-Objective approaches evolved more maze-solving behaviours that generalised to the validation set (table 1).

A key result is the higher (statistically significant) task performance of NS and NS-Objective hybrid search, compared to objective based search. This suggests that the task environment contains features making it amenable to the evolution of effective maze solvers by NS or an NS-Objective hybrid. Previous work [10], [13] indicates that NS performs well in deceptive tasks. Assuming that the high performance of NS in a domain indicates that it is deceptive, then we can conclude that generalised maze-solving is such a domain. However, the notion of deception remains ill defined, and it is difficult to tell *a priori* if a task is deceptive. In mazes, deceptiveness is intuitively gauged by observation, but it is unclear whether tasks such as generalised maze-solving, pole-balancing, quadruped robotic locomotion, and visual discrimination tasks also have elements of deception. In such tasks, the efficacy of NS or a NS-Objective hybrid search approach is yet to be satisfactorily demonstrated [27].

5 Conclusion

As a step towards addressing the research goal of defining controller design methods that elicit generalised problem solving behaviour, this paper presented a comparison of NE methods for generating generalised maze solving behaviours. The experimental comparison used objective (fitness function) versus non-objective (NS) versus a hybrid NS-Objective search as a means of guiding the NE controller design method. This study's specific aim was to elucidate if NE is appropriate for generating generalised maze solvers, and tested the three search metrics as the NE method's salient feature. Results indicated that the NS and NS-Objective approaches yielded comparable task performances, but out-performed a fitness function directed NE. These results support previous work that indicate that NS directed NE is appropriate for solving deceptive tasks. However, the efficacy of controller evolution driven by fitness, NS and NS-Objective hybrid search for eliciting problem solving behaviours in a broader range of tasks, especially those that do not include deception, is the subject of future research.

References

1. Goertzel, B., Pennachin, C.: Artificial general intelligence. Springer (2007)
2. Schmidhuber, J.: Ultimate cognition à la gödel. Cognitive Computation **1**(2) (2009) 177–193
3. Hutter, M.: Universal artificial intelligence. Springer (2005)
4. Looks, M., Goertzel, B., Pennachin, C.: Novamente: An integrative architecture for general intelligence. In: AAAI fall symposium, achieving human-level intelligence. (2004)
5. Genesereth, M., Love, N., Pell, B.: General game playing: Overview of the aaai competition. AI magazine **26**(2) (2005) 62
6. Finnsson, H., Björnsson, Y.: Simulation-based approach to general game playing. In: AAAI. Volume 8. (2008) 259–264
7. Floreano, D., Dürr, P., Mattiussi, C.: Neuroevolution: from architectures to learning. Evolutionary Intelligence **1**(1) (2008) 47–62
8. Rozin, P.: The evolution of intelligence and access to the cognitive unconscious. In: Progress in Psychology. Academic Press (1976) 245–280
9. Eiben, A., Smith, J.: Introduction to evolutionary computing. springer (2003)
10. Lehman, J., Stanley, K.: Abandoning objectives: Evolution through the search for novelty alone. Evolutionary computation **19**(2) (2011) 189–223
11. Gomez, F., Miikkulainen, R.: Solving non-markovian control tasks with neuroevolution. In: IJCAI. Volume 99. (1999) 1356–1361
12. Gomez, F., Schmidhuber, J., Miikkulainen, R.: Efficient non-linear control through neuroevolution. In: Machine Learning: ECML 2006. Springer (2006) 654–662
13. Velez, R., Clune, J.: Novelty search creates robots with general skills for exploration. In: Proceedings of the 2014 conference on Genetic and evolutionary computation, ACM (2014) 737–744
14. Shorten, D., Nitschke, G.: How evolvable is novelty search? (2014)
15. Clune, J., Beckmann, B., Ofria, C., Pennock, R.: Evolving coordinated quadruped gaits with the hyperneat generative encoding. In: Evolutionary Computation, 2009. CEC'09. IEEE Congress on, IEEE (2009) 2764–2771

16. Coleman, O., Blair, A., Clune, J.: Automated generation of environments to test the general learning capabilities of ai agents
17. Richards, N., Moriarty, D., Miikkulainen, R.: Evolving neural networks to play go. Applied Intelligence **8**(1) (1998) 85–96
18. Yong, C., Miikkulainen, R.: Coevolution of role-based cooperation in multiagent systems. Autonomous Mental Development, IEEE Transactions on **1**(3) (2009) 170–186
19. Shorten, D., Nitschke, G.: Generational neuro-evolution: restart and retry for improvement. In: Proceedings of the 2014 conference on Genetic and evolutionary computation, ACM (2014) 225–232
20. Koppejan, R., Whiteson, S.: Neuroevolutionary reinforcement learning for generalized helicopter control. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation, ACM (2009) 145–152
21. Rajagopalan, P., Rawal, A., Holekamp, K., Miikkulainen, R.: General intelligence through prolonged evolution of densely connected neural networks. In: Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion, ACM (2014) 35–36
22. Lehman, J., Stanley, K.: Efficiently evolving programs through the search for novelty. In: Proceedings of the 12th annual conference on Genetic and evolutionary computation, ACM (2010) 837–844
23. Urbano, P., Georgiou, L.: Improving grammatical evolution in santa fe trail using novelty search. In: Advances in Artificial Life, ECAL. Volume 12. (2013) 917–924
24. Cuccu, G., Gomez, F., Glasmachers, T.: Novelty-based restarts for evolution strategies. In: Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE (2011) 158–163
25. Lehman, J., Stanley, K.: Revising the evolutionary computation abstraction: minimal criteria novelty search. In: Proceedings of the 12th annual conference on Genetic and evolutionary computation, ACM (2010) 103–110
26. Gomes, J., Urbano, P., Christensen, A.: Progressive minimal criteria novelty search. In: Advances in Artificial Intelligence–IBERAMIA 2012. Springer (2012) 281–290
27. Inden, B., Jin, Y., Haschke, R., Ritter, H., Sendhoff, B.: An examination of different fitness and novelty based selection methods for the evolution of neural networks. Soft Computing **17**(5) (2013) 753–767
28. Kistemaker, S., Whiteson, S.: Critical factors in the performance of novelty search. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation, ACM (2011) 965–972
29. Stanley, K., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary computation **10**(2) (2002) 99–127
30. Stanley, K., Bryant, B., Miikkulainen, R.: Real-time neuroevolution in the nero video game. Evolutionary Computation, IEEE Transactions on **9**(6) (2005) 653–668
31. Xu, J., Kaplan, C.: Image-guided maze construction. In: ACM Transactions on Graphics (TOG). Volume 26., ACM (2007) 29
32. Pullen, W.: Think labyrinth: Daedalus (2014)
33. Osmankovic, D., Konjicija, S.: Implementation of q learning algorithm for solving maze problem. In: MIPRO, 2011 Proceedings of the 34th International Convention, IEEE (2011) 1619–1622
34. Gregor, D., Troyer, M.: Boost. mpi (2006)
35. Flannery, B., Teukolsky, S., Vetterling, W.: Numerical Recipes. Cambridge University Press, Cambridge, UK (1986)