

# Improving Crossover of Neural Networks in Evolution Through Speciation

Phillip Verbancsics<sup>(✉)</sup>

SPAWAR Systems Center Pacific, 53560 Hull Street, San Diego, CA, USA  
phillip.verbancsics@navy.mil

**Abstract.** Crossover is an important genetic operator that re-combines beneficial genes together and rapidly traverses the fitness landscape. Unfortunately, neuro-evolution (NE) has not experienced the benefits of crossover. Indeed, observations have shown that crossover has been detrimental to NE approaches. Tangentially, speciation has become an important feature in NE for diversity maintenance; however, such speciation research has focused on *what* measure is driving speciation versus *how* the measure determines species. This research posits that appropriate speciation implementations enable effective crossover by determining an individual's potential mating partners. Prior speciation research demonstrated the impact of restricting the mating pools of genomes on search performance. This paper investigates these concepts in the context of NE and results demonstrate; (1) the impact of speciation implementation in NE, (2) crossover's negative effect on search in NE, and (3) a novel speciation approach that enables effective crossover in NE.

**Keywords:** Speciation · Crossover · Artificial neural networks · NEAT

## 1 Introduction

Speciation and crossover are two prominent features of natural evolution that have significant mutual interaction [2]. In nature, speciation results in reproductive isolation that, in turn, limits the mating partners for sexual reproduction. This limitation of mating partners for reproduction then influences the genetic make-up of a population [1]. Diverging genetics among populations can be reinforced through crossover, which potentially results in speciation. Thus speciation plays an important role in crossover and vice versa. Similarly, evolutionary algorithms have explored speciation and crossover as important factors in evolutionary search, but not the interaction of the two aspects of evolution. This paper investigates the interaction of speciation and crossover and their impact on performance, in particular their effect on neuro-evolution (NE).

Interestingly, evolutionary approaches for artificial neural networks (ANNs) that rely on crossover have historically been shown to have diminished performance relative to approaches that emphasize mutation [14], despite evidence that natural evolution benefits from crossover [3]. This negative performance

impact is thought to be due to the permutation problem, that is, multiple neural network genotypes can encode functionally equivalent phenotypes. This many-to-one mapping results in crossover for neural networks being ineffective. Additionally, the connectionist nature of neural networks presents a difficulties in representing the “building blocks” necessary for effective crossover. These limitations have resulted in dismissal of crossover as an operator in NE.

The effectiveness of crossover is influenced by the pool of available mating partners. Such pools are limited by reproductive isolation, which is reinforced by speciation [6]. Speciation has proven to be popular in evolutionary algorithms to encourage the formation of “niches” to preserve diversity, that is, rather than being the driver of limited mating pools, the focus is on preventing the population from prematurely converging [4]. Indeed, such speciation has proven effective in NE for preserving diversity in the population and improving performance [9]. Further research has focused on alternative measures to induce speciation, such as genotypic, phenotypic, and behavioral [8]. These alternate approaches are meant to preserve different types of diversity, thus research has focused creating “niches” rather than better mating pools for crossover.

In contrast, this paper explores how varying the heuristic that is applied to the speciation metric in order to select the species for genomes can influence. Four different heuristics for selecting species are investigated: First Compatible, Most Compatible, Parental, and Uncanny Valley (described in Sect. 3). The experiments to explore the interaction of speciation heuristic and crossover are performed in classic benchmark and calibration domains for NE (XOR and double-pole balancing) under high mutation and high crossover rates. Results from the experiments demonstrate the negative effect that relying highly on crossover has on performance, with the majority of speciation heuristics performing worse under such conditions. However, changing the heuristic results in different performance profiles. Indeed, Most Compatible is consistently the worst performing heuristic, while Uncanny Valley is the highest performer and performs similarly well under both high and low crossover conditions. These results indicate the importance of the effect of speciation heuristics on the performance of crossover and reveal the potential to unlock crossover’s power through improved mating pool selection.

## 2 NeuroEvolution of Augmenting Topologies (NEAT)

This section briefly reviews the NEAT evolutionary algorithm [9, 10], a prominent method that evolves ANNs. NEAT evolves connection weights as well as adds new nodes and connections over generations, thereby increasing solution complexity. It has been proven to be effective in challenging control and decision making tasks [10–13]. NEAT starts with a population of small, simple ANNs that increase their complexity over generations by adding new nodes and connections through mutation. That way, the topology of the network does not need to be known a priori; NEAT searches through increasingly complex networks as it evolves their connection weights to find a suitable level of complexity. The techniques that facilitate evolving a population of diverse and increasingly complex

networks are described in detail in Stanley and Miikkulainen [9]; Stanley and Miikkulainen [10]; the important concept for the approach in this paper is that NEAT implements a solution to the *competing conventions problem* (a.k.a. the Permutation problem) through *historical markings* thereby enabling crossover and speciation for neural network genomes.

The NEAT implementation for this paper differs in a few details. First, crossover is on a neuron basis rather than connection basis, that is, genes are selected from parents through matching neurons. Thus the unit of crossover is a neuron rather than a connection. This change is inspired by the idea of neurons as feature extractors and detectors [14]. Second, this change to a neuron-based crossover change the way compatibility is counted. Excess connections are now defined as mis-matching connections on matched neurons. Disjoint connections are defined connections on mis-matching neurons. In addition, the count of mis-matching neurons now is incorporated into the compatibility metric. Third, crossover and mutation are mutually exclusive, that is, a new genome is either created through crossover without mutation or cloning with mutation, but mutation is never applied after crossover. Finally, the number of new species allowed to be created each generation is limited to one.

### 3 Speciation Heuristics

This paper investigates four different genotypic-based speciation heuristics. First is First Compatible, the original heuristic in NEAT, wherein genomes are placed with the first species that has compatibility below the current threshold (Algorithm 1). If no species is below the threshold, either a new species is created for the genome (if no new species has already been created) or the genome is placed with the most compatible of the species.

Second, Most Compatible places genomes with the species that is the most compatible (Algorithm 2). If no species is below the threshold, either a new species is created for the genome (if no new species has already been created) or the genome is placed with the most compatible of the species. This heuristic is a logical extension of First Compatible because genomes would ideally always be matched with their most compatible counterparts.

Third, Parental looks *only* at the species of the genome's parents to determine compatibility (Algorithm 3). If the none of the parent species are below the threshold, either a new species is created for the genome (if no new species has already been created) or the genome is placed with the most compatible of the parent and new species. This approach is intuited from nature, in that a child belongs either to its parent species or to a new sub-species of the parent species.

Finally, the fourth is Uncanny Valley looks first at the species of the genome's parents (Algorithm 4). If the genome is below the threshold for the parent, then the remaining species are investigated to see if there is another, non-parent species below the threshold. If there exists an additional compatible species (that is not the parent), the genome is placed in that species, otherwise the genome is placed with the parent species. If the genome is not compatible with

```

Input: Genome to be speciated
Output: Species to which the genome is assigned
Selected Species = null;
Minimum Compatibility =  $\infty$ ;
foreach Species s in current set of species do
    Compatibility = GetCompatibility(s, Genome);
    if Compatibility < Minimum Compatibility then
        Minimum Compatibility = Compatibility;
        Selected Species = s;
    if Compatibility  $\leq$  Threshold then
        break;
if Minimum Compatibility > Threshold then
    if New Species == null then
        New Species = CreateNewSpecies(Genome);
        Selected Species = New Species;
        Add New Species to current set of species;
Output Selected Species;

```

**Algorithm 1.** First Compatible Speciation Heuristic

```

Input: Genome to be speciated
Output: Species to which the genome is assigned
Selected Species = null;
Minimum Compatibility =  $\infty$ ;
foreach Species s in current set of species do
    Compatibility = GetCompatibility(s, Genome);
    if Compatibility < Minimum Compatibility then
        Minimum Compatibility = Compatibility;
        Selected Species = s;
if Minimum Compatibility  $\geq$  Threshold then
    if New Species == null then
        New Species = CreateNewSpecies(Genome);
        Selected Species = New Species;
        Add New Species to current set of species;
Output Selected Species;

```

**Algorithm 2.** Most Compatible Speciation Heuristic

its parent species, either a new species is created for the genome (if no new species has already been created) or it is placed with the most compatible of the parent and new species. This heuristic is named after the “uncanny valley” principle from robotics [7], since it follows a similar pattern where there is a dip in probability of being placed with your parents the closer you are in compatibility. This heuristic is similar to “anti-incest” approaches [5] in that it suppresses mating between substantially similar genomes by encouraging genomes to join the next most similar species. The next section details the experiments that reveal the differences resulting from these speciation heuristics.

```

Input: Genome to be speciated
Output: Species to which the genome is assigned
Selected Species = null;
Minimum Compatibility =  $\infty$ ;
foreach Species s in species of genome's parents do
    Compatibility = GetCompatibility(s, Genome);
    if Compatibility < Minimum Compatibility then
        Minimum Compatibility = Compatibility;
        Selected Species = s;
if Minimum Compatibility > Threshold then
    if New Species == null then
        New Species = CreateNewSpecies(Genome);
        Selected Species = New Species;
    else
        Compatibility = GetCompatibility(New Species, Genome);
        if Compatibility < Minimum Compatibility then
            Selected Species = New Species;
Output Selected Species;

```

**Algorithm 3.** Parental Speciation Heuristic

## 4 Experimental Approach

Each of the speciation heuristics are tested in two domains under 0.1 and 0.9 crossover rates to demonstrate the effects that speciation heuristic selection has on performance and crossover. First is the XOR problem, an important calibration domain to ensure neuro-evolution can correctly solve and optimize non-linear functions. XOR is a logical operator that returns true iff only one of the inputs is true. True is represented by +1 and false is represented by -1. The two inputs to XOR must be combined at a hidden unit, as opposed to only at the output. In the experiments with XOR, evolution is limited to 1000 generations and will not terminate at the first solution.

The second domain examined is the double-pole balancing domain [10], which is a well-known benchmark in reinforcement learning domains. In double-pole balancing, two poles are attached at a hinge to a movable cart. The role of the learning agent is to discover how to keep both pole elevated and not allow either to hit the cart. The agent keeps the poles elevated by instructing the cart to move at particular velocity and within the boundaries of the track. Fitness is determined by the number of time-steps the agent keeps the poles elevated, which is capped at 100000 for these experiments. Furthermore, the number of generations is limited to 500 and evaluation of a given run is stopped once a solution that achieves 100000 time steps is discovered. The next section describes the results of these experiments with speciation and crossover.

```

Input: Genome to be speciated
Output: Species to which the genome is assigned
Selected Species = null;
Minimum Compatibility =  $\infty$ ;
foreach Species s in species of genome's parents do
    Compatibility = GetCompatibility(s, Genome);
    if Compatibility < Minimum Compatibility then
        Minimum Compatibility = Compatibility;
        Selected Species = s;
if Minimum Compatibility < Threshold then
    foreach Species s in current set of species do
        Compatibility = GetCompatibility(s, Genome);
        if Compatibility < Threshold then
            Minimum Compatibility = Compatibility;
            Selected Species = s;
else
    if New Species == null then
        New Species = CreateNewSpecies(Genome);
        Selected Species = New Species;
    else
        Compatibility = GetCompatibility(New Species, Genome);
        if Compatibility < Minimum Compatibility then
            Selected Species = New Species;
Output Selected Species;

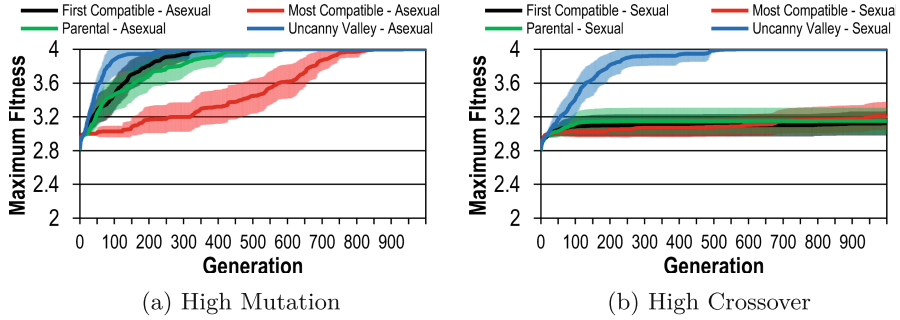
```

**Algorithm 4.** Uncanny Valley Speciation Heuristic

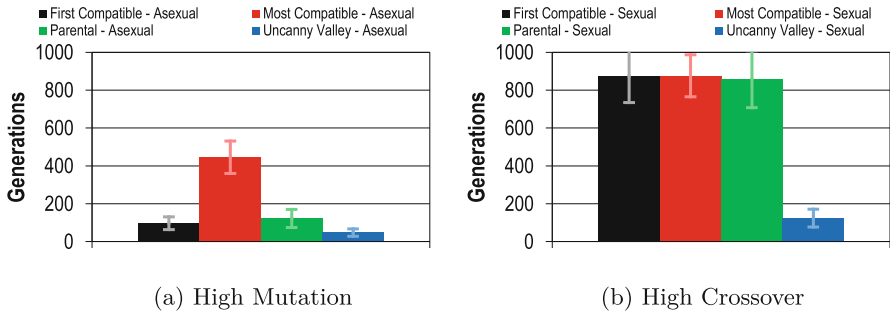
## 5 Results

All results are averaged over 40 runs and under identical settings (Appendix A) except for speciation heuristic and crossover rate. In the XOR domain with a low crossover (high asexual) rate of 10 %, all the speciation heuristics find a solution and optimizes to a perfect solution of XOR within 1000 generations (Fig. 1a), but demonstrate different learning trajectories getting there. While Most Compatible and Parental are substantially similar, the Uncanny Valley quickly spike upward in fitness and the Most Compatible slowly reaches the optimal. Figure 2a show the number of generations to achieve the first (not most exact) solution to XOR. The Uncanny Valley finds a solution to XOR significantly ( $p < 0.01$ ) faster than the other speciation heuristics at an average of 47.6 generations to the first solution. The First Compatible and Parental are the next faster performers and are not significantly different with 97.2 and 122.5 generations on average, respectively. Finally, Most Compatible is significantly ( $p < 0.01$ ) worse than the other heuristics at 445.8 generations on average.

In contrast, a high crossover rate of 90 % results in First Compatible (874.6 generations to first solution), Most Compatible (987.6 generations), and Parental (857.9 generations) as having similarly dismal performances (Fig. 1b) and, in



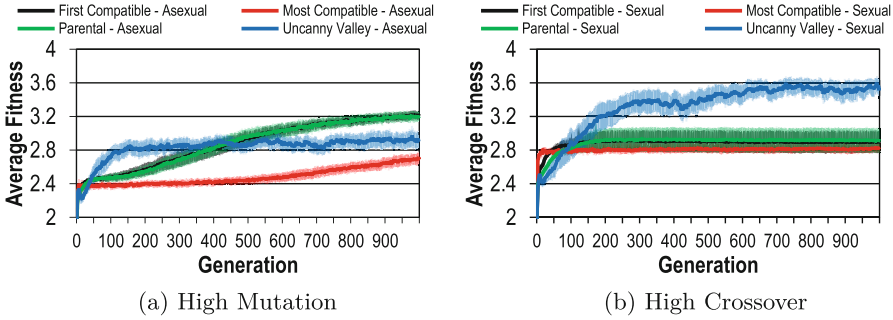
**Fig. 1.** Maximum fitness for XOR by speciation approach. The maximum fitness (averaged over 40 runs) for each generation is shown, surrounded by a 99 % confidence interval, under high mutation (a) and high crossover (b) rates.



**Fig. 2.** Average generations to first solution for XOR by speciation approach. The generation count to the first genome that solves XOR (averaged over 40 runs) is shown with a 99 % confidence interval for high mutation (a) and high crossover (b) rates.

fact, do not find a solution to XOR in a majority of runs (Fig. 2b). However, Uncanny Valley (123.7 generations) significantly ( $p < 0.01$ ) outperforms the other approaches at all generations past generation 50. Indeed, the time to first solution for Uncanny Valley with high crossover is not significantly different from First Compatible and Parental with low crossover rates.

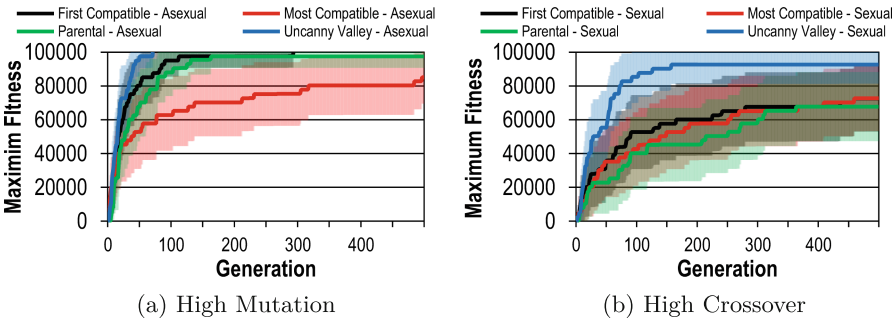
Further contrast in the differences in heuristic and the effect of crossover can be seen in the average population fitness (Fig. 3). Similar to the maximum fitness results for the high asexual reproduction, the average population fitness under high asexual reproduction results in the Most Compatible and Parental as being statistically the same at most generations and the Uncanny Valley and Most Compatible as different from all other heuristics with significance  $p < 0.01$ . Interestingly, the final average population fitness scores for First Compatible (3.2) and Parental (3.2) are higher than Uncanny Valley (2.9), though Uncanny Valley's max fitness is better, while Most Compatible (2.7) remains the worst performer. For the high crossover rate setting, the average population fitness of Uncanny Valley (3.6) is significantly ( $p < 0.01$ ) greater than the other speciation heuristics,



**Fig. 3.** Average population fitness for XOR by speciation approach. The mean population fitness (averaged over 40 runs) is shown for each of the 1000 generations and surrounded with a shaded region representing a 99 % confidence interval under high mutation (a) and high crossover (b) rates.

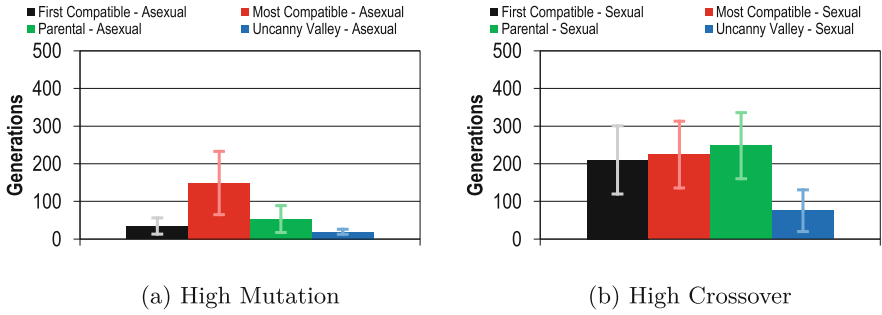
which quickly converge to 2.91 (First Compatible, Parental) and 2.83 (Most Compatible). Indeed, the Uncanny Valley under high crossover significantly outperforms itself under low crossover rates.

In the DPB domain with low crossover, only First Compatible and Uncanny Valley always find a solution with a perfect score (Fig. 4a), Parental achieves an average max fitness of 95716 and Most Compatible achieves significantly ( $p < 0.01$ ) less than all other approaches at 82717. Figure 5a show the number of generations to achieve a solution. In this domain with the low crossover rate, only Most Compatible (148.8 generations) is significantly different from the other heuristics. Most Compatible performs worse than Parental (53.1), First Compatible (34.6), and Uncanny Valley (19.2).



**Fig. 4.** Maximum fitness for Double Pole Balancing (DPB) by speciation approach. The maximum fitness (averaged over 40 runs) is shown for each generation until the stopping criteria is met (performance of 100000 time steps or up to 500 generations) and surrounded with a shaded region representing a 99 % confidence interval under high mutation (a) and high crossover (b) rates.

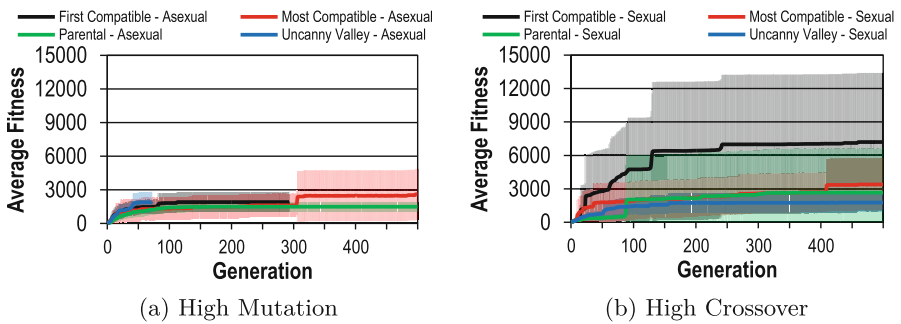




**Fig. 5.** Average generations to a solution for Double Pole Balancing (DPB) by speciation approach. The generation count to the first genome that meets the stopping criteria (performance equal to 100000 time steps) for DPB (averaged over 40 runs) is shown with a 99 % confidence interval for high mutation (a) and high crossover (b).

Under a high crossover rate, none of the heuristics find a solution in every run within 500 generations. However, Uncanny Valley achieves an average champion fitness of 92647 that significantly ( $p < 0.01$ ) outperforms the non-significantly different First Compatible (72590), Most Compatible (72678), and Parental (67756) at all generations past 70 (Fig. 4b). Indeed, the generations to a solution (Fig. 5b) for Uncanny Valley (75.3) is significantly fewer than First Compatible (210.1), Most Compatible (224.2), and Parental (248.1).

The average population fitness (Fig. 6) for DPB provides further insight into the interaction of speciation and crossover. In the low crossover scenario, there is no significant difference among the average population fitness by speciation heuristic, through the Uncanny Valley and First Compatible terminate early due to finding solutions. However, the high crossover scenario reveals differences



**Fig. 6.** Average population fitness for Double Pole Balancing (DPB) by speciation approach. The mean population fitness (averaged over 40 runs) is shown for each generation or until the stopping criteria of a champion with a 100000 time step performance is found and surrounded with a shaded region representing a 99 % confidence interval under high mutation (a) and high crossover (b).

among the approaches. The final average population fitness scores for First Compatible (7189) is significantly ( $p < 0.01$ ) higher than the other heuristics. Interestingly, Uncanny Valley’s average population fitness reverses the pattern from its max fitness, that is, Uncanny Valley’s average population fitness of 1772 is the lowest of the heuristics, while its max fitness is the highest.

## 6 Discussion

In nature, crossover is the primary (or only) form of reproduction for the most complex of organisms, especially those whose intelligence neuro-evolution (NE) is attempting to replicate. However, crossover in NE has not been beneficial, resulting in degraded performance in approaches that rely in crossover. Indeed, this paper provides evidence that crossover negatively affects NE performance.

Speciation has become an important aspect of NE approaches, with significant research examining measures to differentiate genomes into species to preserve diversity. This paper showed that different heuristics to place genomes into species based off a particular metric can enhance or impede performance. The Uncanny Valley speciation heuristic showed benefits in both the XOR and DPB domains, significantly improving performance over the baseline NEAT heuristic (First Compatible) as well as two other intuitively sensible heuristics (Most Compatible and Parental). Speciation improves performance of crossover by acting as a means of limiting the pool of candidates that may mate. Thus speciation, appropriately implemented in artificial evolution, can provide benefits to crossover by maintaining appropriate candidates pools.

Note, this paper doesn’t claim one speciation heuristic is better than another, rather the deeper concept is that seemingly small implementation details (e.g. the speciation heuristic) can have a significant effect on the performance of an algorithm and interaction effects with varied parameters. Indeed, one naive implementation of speciation in NEAT is to group genomes that are most compatible together; however, these results show that such an approach is the least ideal performance-wise on the simple tasks of XOR and double pole balancing. Thus, it is important when considering algorithm performance that these disregarded details may be the cause of negative performance, such as in the negative performance of crossover in NE that is solved by better mate selection.

## 7 Conclusion

Crossover and speciation are two important features of natural evolution in their own right that can have significant mutual interaction. NE approaches have embraced speciation, but have not been able to exploit the power of crossover. Indeed, experiments in this paper demonstrated that crossover can be detrimental to performance in NE. However, results shows that different speciation heuristics change the performance of crossover. In fact, careful selection of speciation heuristic can allow NE to perform as well under high crossover as low crossover and improve performance under all conditions. Overall, speciation

plays an important role in NE in preserving diversity, but a well-designed speciation heuristic has the potential to improve performance and unlock the power of crossover.

**Acknowledgments.** This work was supported and funded by the SSC Pacific Naval Innovative Science and Engineering (NISE) Program.

## A NEAT Parameters Shared Across Experiments

Parameter name	Parameter value
Population size	200
Minimum species #	5
Maximum species #	15
Interspecies mating rate	0.01
Selection proportion	0.5
Elitism proportion	0.01
Disjoint weight coefficient	1
Excess weight coefficient	1
Matching weight coefficient	0.4
Recurrence	no
Maximum weight magnitude	5
Initial connection ratio	1
Probability mutate weights	0.9
Probability add neuron	0.01
Probability add connection	0.08
Probability delete neuron	0.001
Probability delete connection	0.09
Probability select fitter gene	0.5
Probability recombine excess (fitter genome)	0.8
Probability recombine excess (less fit genome)	0.15

## References

1.

Barton, N.H., Bengtsson, B.O.: The barrier to genetic exchange between hybridising populations. *Heredity* **57**(3), 357–376 (1986)

2.

Bernstein, H., Byerly, H.C., Hopf, F.A., Michod, R.E.: Sex and the emergence of species. *J. Theor. Biol.* **117**(4), 665–690 (1985)

3. Colegrave, N.: Sex releases the speed limit on evolution. *Nature* **420**(6916), 664–666 (2002)
4. Deb, K., Goldberg, D.E.: An investigation of niche and species formation in genetic function optimization. In: *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 42–50. Morgan Kaufmann Publishers Inc. (1989)
5. Eshelman, L.J., Schaffer, J.D.: Preventing premature convergence in genetic algorithms by preventing incest. *ICGA* **91**, 115–122 (1991)
6. Mayr, E.: *The Growth of Biological Thought: Diversity, Evolution, and Inheritance*. Harvard University Press, Cambridge (1982)
7. Mori, M.: Bukimi no tani [the uncanny valley]. *Energy* **7**(4), 33–35 (1970). (online)
8. Mouret, J.-B., Doncieux, S.: Encouraging behavioral diversity in evolutionary robotics: an empirical study. *Evol. Comput.* **20**(1), 91–133 (2012)
9. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. *Evol. Comput.* **10**, 99–127 (2002)
10. Stanley, K.O., Miikkulainen, R.: Competitive coevolution through evolutionary complexification. *J. Artif. Intell. Res.* **21**, 63–100 (2004)
11. Taylor, M.E., Whiteson, S., Stone, P.: Comparing evolutionary and temporal difference methods in a reinforcement learning domain. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006)*, pp. 1321–1328. ACM Press, New York, July 2006
12. Whiteson, S.: Improving reinforcement learning function approximators via neuroevolution. In: *AAMAS 2005, Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 1386–1386. ACM, New York (2005)
13. Whiteson, S., Whiteson, D.: Stochastic optimization for collision selection in high energy physics. In: *IAAI 2007, Proceedings of the Nineteenth Annual Innovative Applications of Artificial Intelligence Conference*, Vancouver, British Columbia, Canada. AAAI Press, July 2007
14. Yao, X.: Evolving artificial neural networks. *Proc. IEEE* **87**(9), 1423–1447 (1999)