The Effect of Connection Cost on Modularity in Evolved Neural Networks

Jessica Lowell[1] and Jordan Pollack[1]

[1]DEMO Lab, School of Computer Science, Brandeis University, Waltham, MA 02453
jessiehl@cs.brandeis.edu

Abstract

Modularity, often observed in biological systems, does not easily arise in computational evolution. We explore the effect of adding a small fitness cost for each connection between neurons on the modularity of neural networks produced by the NEAT neuroevolution algorithm. We find that this connection cost does not increase the modularity of the best network produced by each run of the algorithm, but that it does lead to increased consistency in the level of modularity produced by the algorithm.

Introduction

As evolutionary systems are increasingly used in a variety of applications, the organization of the evolved solutions has become important. Modularity, the organization of a system into interacting subparts, is observed in both many natural and many engineered networks (Koza, 1992; Simon, 1996; Hartwell et al., 1999; Wagner and Altenberg, 1996). We briefly discuss modularity in natural and simulated evolution, as well as the NEAT neuroevolution algorithm, a widely used evolutionary algorithm.

Modularity

Both biological networks, such as neural networks and bacterial metabolic networks, and biological systems in general, such as tissues assembled from cells, tend to be modular. The evolutionary reasons for this are still unclear, especially because computational models of biological evolution tend to produce nonmodular solutions. The nonmodular solutions produced by network-evolving algorithms are often connected in complicated ways and better-performing on the specific task for which they are optimized than modular solutions designed by humans (Thompson, 1998; Vassilev et al., 2000). However, this lack of modularity in computational evolution means that while it can produce highly optimized solutions for simple problems, it has difficulty solving more complex problems (Kashtan and Alon, 2005).

Some work has been done on creating modularity through evolutionary algorithms by building the encapsulation of modules into the algorithms (Garibay et al., 2004; Wiegand et al., 2009). This is sufficient if the goal is simply to create modular solutions. However, if the goal is to understand how modularity evolves in nature, imposing modularity on the algorithm does not suffice. Furthermore, even if the goal is an engineering one rather than one of biological discovery, allowing the evolutionary process to discover modularity naturally ensures that high-performing species of solutions are not excluded from the search space from the start because they began as strongly nonmodular and took many generations to evolve modularity.

How modularity evolves in both natural and simulated evolution is a complex problem that may involve multiple forces whose contributions must be teased out. A major hypothesis is that it evolves in response to varying environments (called modularly varying goals), in which solutions must perform different tasks with common subtasks (Kashtan and Alon, 2005; Kashtan et al., 2007). This effect was proposed by Lipson et al. (2002) in a study of minimal substrate modularization which found that modular separation is logorithmically proportional to rates of environmental variation, and suggested that evolutionary design of engineered systems should use variable rather than fixed fitness critiera. The effect of modularly varying goals has been observed in both computational (Kashtan and Alon, 2005) and natural (Kashtan et al., 2007; Parter et al., 2007) evolution studies. An alternative hypothesis, supported by the work of Clune et al. (2013), is that modular networks evolve in response to a "connection cost", or small decrease in fitness for each connection in the network, analagous to the energy cost of forming a physical link between two cells, organisms, or other nodes in the biological network. The debate over which of these hypotheses, if either, plays the larger role in the evolution of modularity, is ongoing. We investigated the connection cost hypothesis as applied to NeuroEvolution of Augmenting Topologies (NEAT), a major method for evolving artificial neural networks. Future work may compare both hypotheses on NEAT or another neuroevolution algorithm.

NEAT

There are numerous algorithms for evolving neural networks, some of which evolve topology only or weights only, and some of which evolve both. NEAT (Stanley and Miikkulainen, 2002) is an example of a neuroevolution algorithm that evolves both topology and weights. It starts with very simple networks and gradually complexifies candidate solutions using crossover and three forms of mutation: adding neurons, adding connections, and changing connection weights. To protect innovations long enough to see if they will be evolutionarily useful, it also uses speciation, which isolates subsets of the population into reproductive groups based on how different they are topologically. Finally, it tracks the history of innovations through historical innovation numbers to mitigate the competing conventions problem. NEAT has proven useful in several problem domains (Stanley and Miikkulainen, 2002, 2004), and is one of the most popular neuroevolution algorithms.

The standard NEAT algorithm does not tend to produce modular solutions (Reisinger et al., 2004). Reisinger et al. (2004) created Modular NEAT, a version of NEAT that does produce modular solutions, but forced the algorithm's predisposition toward modularity by requiring it to reuse neural substructures in different spatial locations to form complete neural networks. Clune et al. (2010) found that HyperNEAT, a variant of NEAT, did not tend to produce modular networks. Verbancsics and Stanley (2011) were able to influence HyperNEAT toward modularity by seeding it with a bias toward local connection - the connection of components that are spatially near each other - which is another force, besides those previously mentioned, that may play a role in biological modularity.

In this paper, we take the connection cost hypothesis and apply it to NEAT in order to study whether it influences the emergence of modularity in networks produced by NEAT in the same way that it influenced the emergence of modularity in other problem domains in (Clune et al., 2013). This includes the level of modularity, but it also includes variance in modularity. In other words, we ask whether connection cost makes the amount of modularity emerging from NEAT more consistent from trial to trial, in addition to looking at its influence on the overall amount of modularity.

Methods

To test the effect of connection cost on modularity in NEAT, we used two different versions of the eight-pixel retina problem, which was developed by Kashtan and Alon (2005) for studies on the emergence of modularity, and which has been used in other studies on the emergence of modularity (Clune et al., 2010; Verbancsics and Stanley, 2011). In the retina problem, neural networks attempt to recognize certain patterns (objects) in an four-pixel by two-pixel retina, and return true or false based on whether those objects are present. The first is modularly decomposable, in that the neural network has to determine whether objects exist on each of the left and right sides (which each contain four pixels) in order to determine whether objects exist on both sides. The second version of the problem is not modularly decomposable, in that an object need only exist on one side in order for the function to return an output of true. Which patterns count as objects are slightly different for the left and right sides of the retina for both versions of the problem. This is illustrated in Fig. 1.

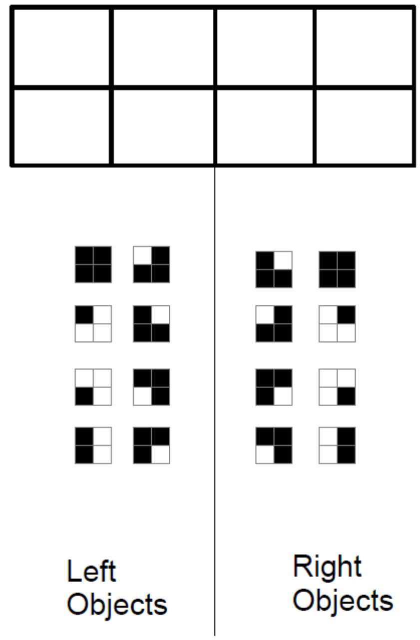

Figure 1: Illustration of left and right objects in th eight-pixel retina problem (adapted from Kashtan and Alon (2005)).

We ran the NEAT evolutionary process on both versions of the problem, using the NEAT4J open source Java implementation of NEAT (Simmerson, 2006) as a basis, with and without connection cost. The parameters used are listed in Table 1.

Problem Version	Trials	Connection Cost
Modular	20	0
Modular	20	1.0×10^{-5}
Nonmodular	20	0
Nonmodular	20	5.0×10^{-5}

Table 1: Retina problem versions and parameters used.

The connection cost for each problem was determined by testing several different orders of magnitude for connection cost, and then several different connection costs in intervals of 0.00001, looking for the highest cost that consistently allowed solutions to evolve performance improvements of 10% or greater over the first-generation solution, a fully-connected network of eight input and one output neurons with no hidden layer.

Each run of the NEAT algorithm used a population of 500 neural networks evolving over 2000 generations. Fitness was measured by mean standard error on the problem, meaning that a lower fitness number indicated better performance.

We quantified modularity by using the metric Q, defined by the approach of Newman and Girvan (2004). This approach determines Q by looking at the percentage of edges in the network that connect nodes in the same module, and substracts the expected value for that percentage in a network with the same number of modules but random connections. The modules are defined by a previous part of the algorithm that splits the network into the modules that would maximize Q. Mathematically, the equation for Q is:

$$Q = \sum_{s=1}^{k} \left[\frac{l_s}{L} - \left(\frac{dS}{2L} \right)^2 \right] \tag{1}$$

where L is the number of edges, K is the number of modules, d_s is the sum of degrees of nodes in module s, and l_s is the number of edges in that module.

To determine whether variances in modularity were equal in our sets of results, we used Levene's test, a statistic for assessing the equality of variances across two or more groups. We used Levene's test rather than an F-test of equality of variances because the F-test is highly sensitive to non-normality of distribution, while Levene's test is robust to non-normality.

Results and Discussion

Our first comparisons were between a set of 20 trials without a connection cost and 20 trials with a connection cost of 1.0×10^{-4} per connection, with mutation probability = 0.25 and crossover probability = 0.2. In Fig.2, we can see that the best solutions produced in the trials with no connection cost had a variance in modularity of 0.0449, while those produced in the trials with a connection cost had a variance of only 0.0177 - more than 60% lower. This difference in variance was statistically significant (P = 0.001803).

We examined the possibility that the difference in variances was caused by interactions between crossover and connection cost, rather than connection cost alone. When we removed crossover but kept all other parameters the same, we obtain similar results, as seen in Fig.3. In this case, the variance in modularity for the trials with no connection cost is 0.0455, while the variance with connection cost is 0.0182.

When we underwent the same process using the nonmodular version of the retina problem, we still saw decreases

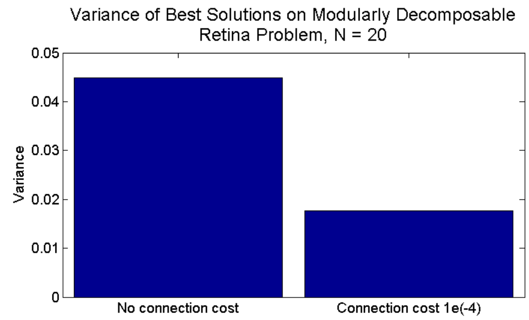

Figure 2: The addition of a connection cost to NEAT, with both mutation and crossover, on the modularly decomposable retina problem produces a decrease in variance across trials. Number of trials N = 20, P = 0.001803

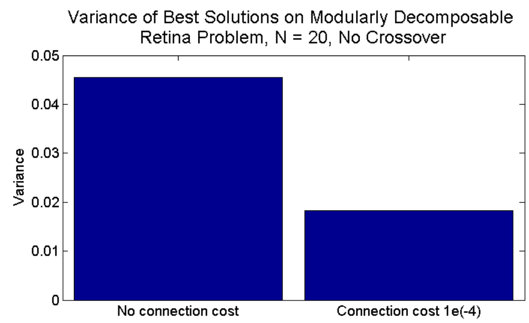

Figure 3: The addition of a connection cost to NEAT, with mutation but without crossover, on the modularly decomposable retina problem produces a decrease in variance across trials. Number of trials N = 20, P = 0.006627

in variance, but they were just below the level of statistical siginficance, in the P = 0.06-0.07 range.

With both mutation and crossover, the variance without connection cost was 0.0189 and the variance with a connection cost of 5.0×10^{-4} per connection was 0.0098 (Fig.4). With mutation but no crossover, the variance without connection cost was 0.0189 and the variance with a connection cost of 5.0×10^{-4} per connection was 0.0078 (Fig.5).

We also examined the effect of crossover itself on modularity, in order to further separate any of its effects from those of connection cost. There was no significant difference between the variances of any set of trials with crossover and the otherwise-equivalent set without crossover (P = 0.617905).

This reduction in variance caused by connection cost represents an increase in predictability. In the presence of connection cost, different runs of NEAT are more likely to produce similarly modular solutions. This suggests that in situations where connecting between nodes involves a physical link and thus an energy cost, there may be an optimal

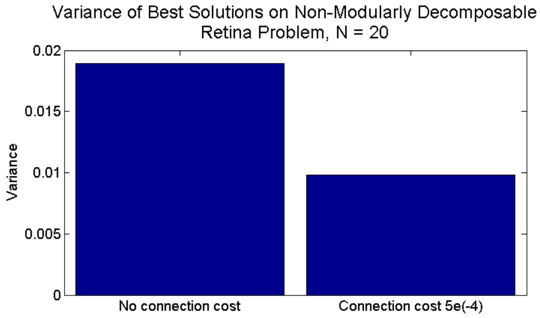

Figure 4: The addition of a connection cost to NEAT, with mutation and crossover, on the non-modularly decomposable retina problem produces a sub-statistically-significant decrease in variance across trials. Number of trials N = 20, P = 0.061516

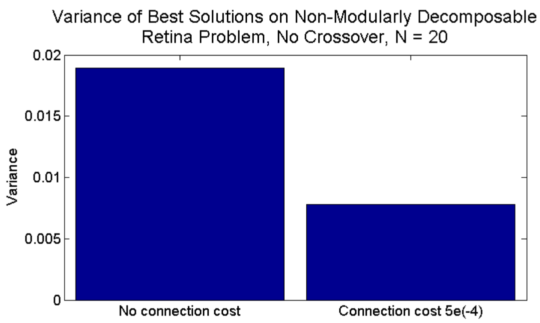

Figure 5: The addition of a connection cost to NEAT, with mutation but without crossover, on the non-modularly decomposable retina problem produces a sub-statistically-significant decrease in variance across trials. Number of trials N = 20, P = 0.062036

level of modularity, a balance between modularity and efficient use of space. Such a balance may be related to work on wiring economy in the human brain described by Bullmore and Sporns (2012), which found that the brain balances modularity with efficient use of space and that an imbalance in either direction causes neurological disorders.

We include images of a few sample networks from our experiments, with varying structures and amounts of modularity. Fig. 6 is a network with $Q = 0$ (all nodes are found to be part of a single module by the Newman-Girvan algorithm), produced by running NEAT4J on the modularly decomposable test problem with no connection cost. Fig. 7 was also produced by running NEAT4J on the modularly decomposable test problem with no connection cost, and is a large network with a high modularity of $Q = 0.4567$, divided by the Newman-Girvan algorithm into 13 modules. Fig. 8 is a small network with a moderate modularity of $Q = 0.1760$, produced on the nonmodularly decomposable test problem

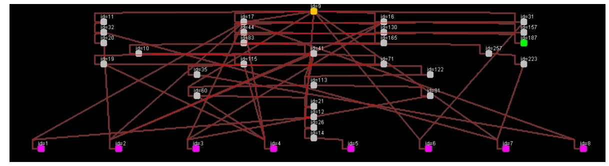

Figure 6: Example network created by running NEAT4J on modularly decomposable test problem with crossover. $Q = 0$, 37 neurons.

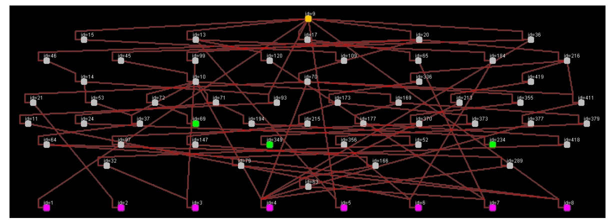

Figure 7: Example network created by running NEAT4J on modularly decomposable test problem with crossover. $Q = 0.4567$, 61 neurons, 13 modules.

with a connection cost, divisible into 6 modules. In these images, input neurons are pink, output neurons are orange, neurons that are not connected to any input neuron are green, and all other neurons are gray.

An observer can see from these networks that modularity is not necessarily a function of network size. However, connection cost did appear to promote smaller networks - in experiments with crossover, average network sizes on the modularly decomposable problem were 47.1 neurons with connection cost vs 62.75 neurons without, and average network sizes on the nonmodularly decomposable problem were 30.6 neurons with connection cost vs 38.4 without. It is possible that the connection cost is preventing new nodes and links from being formed, which we might expect since it slightly penalizes connections between nodes, without necessarily increasing modularity in the process. One reason that connection cost may promote the evolution of modularity in some cases is that in a modular system there are fewer connections between nodes. When nodes and links evolve together, fewer links may also encourage fewer nodes.

It is worth considering whether connection cost had an effect on fitness itself, since if NEAT could maintain fitness in the presence of connection cost, this would present a problem for using connection cost for engineering purposes. As we previously stated, fitness was represented by mean standard error, with a lower fitness being better. On the nonmodularly decomposable problem, fitnesses ranged from 0.0777 to 0.3807, while on the modularly decomposable problem, they ranged from 0.0408 to 0.2856. Connection cost made no statistically significant difference to either the variances of fitnesses or the mean fitness on either problem. Fitness

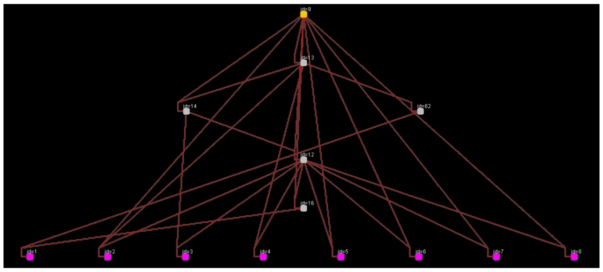

Figure 8: Example network created by running NEAT4J on modularly decomposable test problem with crossover. Q = 0.1760, 16 neurons, 6 modules.

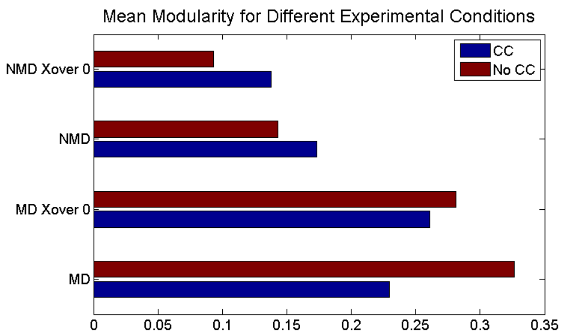

Figure 9: Mean modularity across 20 trials for all experimental conditions, with direct comparisons in the same color. MD represents the modularly decomposable retina problem version, NMD represents the nonmodular, ”Xover 0” indicates that crossover was not used as an evolutionary operator. No direct comparisons showed statistically significant differences. Comparisons of sets with statistically equal variances were done using an unpaired t-test, others were done using Welch’s t-test.

was maintained, though not improved, in the presence of connection cost.

There was no statistically significant effect of connection cost on the $magnitude$ of modularity for either version of the retina problem (Fig.9). This may seem rather surprising given the apparent contradiction with the results of Clune et al. (2013), discussed earlier. This raises the possibility that some aspect of the NEAT algorithm itself prevents connection cost from increasing modularity.

A possible explanation for this difference with previous work on connection cost and modularity is NEAT’s built-in protections against bloat, which do not exist in many other evolutionary algorithms. NEAT complexifies its structures slowly, and separates sufficiently different topologies into species so that they do not interfere with each other. The speciation also means that as long as simpler networks are competitive, they will survive in the population, as their sufficiently complex offshoots will branch into different species. Connection cost and bloat reduction both make it less beneficial for the evolutionary algorithm to form the large, intricately-wired, highly-nonmodular networks that are often the result of evolutionary algorithms, and so the effect of connection cost on the modularity of solutions produced by NEAT may be somewhat redundant.

While none of the results in Fig. 9 were statistically significant, it is interesting to note that for the nonmodularly-decomposable problem there was a trend toward higher modularity with connection cost, but that with the modularly-decomposable problem the trend ran in the opposite direction, with connection cost correlating with lower modularity. The latter trend may be a side effect of the lowering of variance - in a relatively small population, there were fewer high-modularity outliers to bias the mean upward. However, it may also suggest that modularity emerges differently on modularly decomposable and nonmodularly-decomposable problems.

As we previously mentioned, Verbancsics and Stanley (2011) were able to increase the modularity of networks produced by HyperNEAT, a generative, hypercube-based, extension of NEAT, by starting the algorithm with a bias toward local connectivity. Local connectivity is also a part of the basis for the hypothesis that connection cost increases modularity in general. The fact that this failed to happen in our experiments with NEAT raises the question of whether local connectivity has different effects on evolution of modularity depending on whether the network being evolved has a generative or indirect representation, as is the case with HyperNEAT, or a direct representation, as is the case with NEAT. One possible reason for this is that in generative and indirect systems, the evolution of modularity-promoting factors and performance-promoting factors can be separated - for example, in the aforementioned extension of HyperNEAT, weights and connection expression patterns can be evolved separately.

In these experiments, we used a form of connection cost that assumes a constant cost per connection, rather than one that assumes a greater cost for a longer connection (for instance, one based on the principle that in a physical brain there would be a greater energy cost in creating an axon and synapse between two neurons that are far apart than in creating them between two nearby neurons). We made this choice because Clune et al. (2013), using both forms of connection cost, found that there was little difference between them in promoting modularity, and because NEAT does not have a built-in concept of physical distance between neurons. It may be that if an extension of NEAT that did have this concept built-in was developed, different forms of connection cost would have different effects on the evolution of modularity in NEAT-produced networks. In order to obtain the modularity-promoting benefits of indirect representations, as was done with HyperNEAT, the geometry of the network would need to be evolved separately from other aspects of the network.

These results suggest multiple possible directions for future work. We have already mentioned the possibility of developing a version of NEAT that contains a concept of physical distance between neurons and studying how connection cost that is weighted by distance affects the evolution of modularity compared to simple connection cost per link. Another possibility is to investigate whether connection cost leads to more predictability in modularity in general, rather than just when NEAT is the evolutionary algorithm used. Still another is to study the effect of connection cost on populations of neural networks, rather than just the best network produced by each run of the algorithm. Since, as we have discussed, there may be properties particular to NEAT that influenced our results, it could be clarifying to see if the results are similar if a different neuroevolution algorithm is used. Finally, it may be worth testing further how modularity is influenced differently when evolution is occurring on modularly vs nonmodularly decomposable problems.

In the results reported here, we find that adding a connection cost to the NEAT algorithm does not significantly affect the modularity of the top networks produced by NEAT. We speculate that this lack of effect is caused by NEAT's protections against bloat. We find, however, that it does reduce the variance in that modularity, leading to a more consistent level of modularity in the resulting networks, and thus more predictability in the outcome of the algorithm.

Acknowledgements

This work was supported by NSF grant No. 1068620. Thanks to Kyle Harrington for his valuable feedback on the ideas in this paper.

References

Bullmore, E. and Sporns, O. (2012). The economy of brain network organization. *Nature Reviews Neuroscience*, 13(5):336–349.

Clune, J., Beckmann, B., McKinley, P., and Ofria, C. (2010). Investigating whether hyperneat produces modular neural networks. In *Proceedings of the 12th annual conference on Genetic and Evolutionary Computation*, pages 635–642. ACM.

Clune, J., Mouret, J., and Lipson, H. (2013). The evolutionary origins of modularity. *Proceedings of the Royal Society B*, 280(1755):20122863.

Garibay, I. I., Garibay, O. O., and Wu, A. S. (2004). Effects of module encapsulation in repetitively modular genotypes on the search space. In *Proceedings of the 6th annual conference on Genetic and Evolutionary Computation*, pages 1125–1137. Springer.

Hartwell, L., Hopfield, J., Leibler, S., and Murray, A. (1999). From molecular to modular cell biology. *Nature*, 402:C47–52.

Kashtan, N. and Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *Proceedings of the National Academy of Sciences*, 102(39):13773–13778.

Kashtan, N., Noor, E., and Alon, U. (2007). Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences*, 104:13711–13716.

Koza, J. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA.

Lipson, H., Pollack, J. B., and Suh, N. P. (2002). On the origin of modular variation. *Evolution*, 56(8):1549–1556.

Newman, M. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113.

Parter, M., Kashtan, N., and Alon, U. (2007). Environmental variability and modularity of bacterial metabolic networks. *BMC Evolutionary Biology*, 7(1):169.

Reisinger, J., Stanley, K., and Miikkulainen, R. (2004). Evolving reusable neural modules. In *Proceedings of the 6th annual conference on Genetic and Evolutionary Computation*, pages 69–81. Springer Berlin Heidelberg.

Simmerson, M. (2006). Neat4j homepage. online, 2006.

Simon, H. (1996). *The Sciences of the Artificial*. MIT Press, Cambridge, MA.

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

Stanley, K. and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100.

Thompson, A. (1998). *Hardware Evolution: Automatic Design of Electronic Circuits in Reconfigurable Hardware by Artificial Evolution*. Springer, NY.

Vassilev, V., Job, D., and Miller, J. (2000). Towards the automatic design of more efficient digital circuits. In *Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware*, pages 151–160.

Verbancsics, P. and Stanley, K. (2011). Constraining connectivity to encourage modularity in hyperneat. In *Proceedings of the 13th annual conference on Genetic and Evolutionary Computation*, pages 1483–1490. ACM.

Wagner, G. and Altenberg, L. (1996). Complex adaptations and the evolution of evolvability. *Evolution*, 50(3):967–976.

Wiegand, R., Anil, G., Garibay, I. I., Garibay, O. O., and Wu, A. S. (2009). On the performance effects of unbiased module encapsulation. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 1729–1736. ACM.