Competitive Coevolution through Evolutionary Complexification

Kenneth O. Stanley and Risto Miikkulainen
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712 USA
{kstanley, risto}@cs.utexas.edu

Technical Report UT-AI-TR-02-298

December 10, 2002

Abstract

Two major goals in machine learning are the discovery of complex multidimensional solutions and continual improvement of existing solutions. In this paper, we argue that *complexification*, i.e. the incremental elaboration of solutions through adding new structure, achieves both these goals. We demonstrate the power of complexification through the NeuroEvolution of Augmenting Topologies (NEAT) method, which evolves increasingly complex neural network architectures. NEAT is applied to an open-ended coevolutionary robot duel domain where robot controllers compete head to head. Because the robot duel domain supports a wide range of sophisticated strategies, and because coevolution benefits from an escalating arms race, it serves as a suitable testbed for observing the effect of evolving increasingly complex controllers. The result is an arms race of increasingly sophisticated strategies. When compared to the evolution of networks with fixed structure, complexifying networks discover significantly more sophisticated strategies. The results suggest that in order to realize the full potential of evolution, and search in general, solutions must be allowed to complexify as well as optimize.

1 Introduction

Traditional genetic algorithms evolve fixed-length genomes under the assumption that the space of the genome is sufficient to encode the solution. A genome containing n genes encodes a single point in an n-dimensional search space. In many cases, a solution is known to exist somewhere in that space. For example, the global maximum of a function of three arguments *must* exist in the three dimensional space defined by those arguments. Thus, a genome of three genes can encode the location of the maximum.

However, many common structures are defined by an arbitrary number of parameters. In particular, those solution types that can contain a variable number of parts can be represented by *any* number of genes. For example, the number of parts in neural networks, cellular automata, and electronic circuits can vary (Miller et al. 200a; Mitchell et al. 1996; Stanley and Miikkulainen 2002d). In fact, two neural networks with different numbers of connections and nodes can approximate the same function (Cybenko 1989). Thus, it is not clear what number of genes is appropriate for solving a particular problem. Because of this ambiguity,

researchers evolving fixed-length genotypes must use heuristics, such as smaller neural networks generalizing better than larger ones, in order to estimate *a priori* the appropriate number of genes to encode such structures.

A major obstacle to using fixed-length encodings is that heuristically determining the appropriate number of genes becomes impossible for very complex problems. For example, how many nodes and connections are necessary for a neural network that controls a ping-pong playing robot? Or, how many bits are needed in the neighborhood function of a cellular automata that performs information compression? The answers to these questions can hardly be based on empirical experience or analytic methods, since little is known about the solutions. One possible approach is to simply make the genome extremely large, so that the space it encodes is extremely large and a solution is likely to lie somewhere within. Yet the larger the genome, the higher dimensional the space that evolution needs to search. Even if a ping-pong playing robot lies somewhere in the 10,000 dimensional space of a 10,000 gene genome, searching such a space may take prohibitively long.

Even more problematic are open-ended problems where phenotypes are meant to improve indefinitely and there is no known final solution. For example, in competitive games, estimating the complexity of the "best" possible player is difficult because making such an estimate implicitly assumes that no better player can exist. How could we ever know that? Moreover, many artificial life domains are aimed at evolving increasingly complex artificial creatures for as long as possible (Maley 1999). Fixing the size of the genome in such domains also fixes the maximum complexity of evolved creatures, defeating the purpose of the experiment.

In this paper, we argue that the right way to evolve arbitrarily structured phenotypes is to start evolution with a population of small, simple genomes and systematically *complexify* solutions over generations by adding new genes. That way, evolutions begins searching in a small easily-optimized space, and adds new dimensions as necessary. This approach is more likely to discover highly complex phenotypes than an approach that begins searching directly in the intractably large space of complete solutions. In fact, natural evolution has itself utilized this strategy, occasionally adding new genes that lead to increased phenotypic complexity (Martin 1999; Section 2). In biology, this process is called *complexification*, which is why we use this term to describe our approach as well.

When a good strategy is found in a fixed-length genome, the entire representational space of the genome is used to encode it. Thus, the only way to improve it is to *alter* the strategy, thereby sacrificing some of the functionality learned over previous generations. In contrast, complexification *elaborates* on the existing strategy by adding new structure without changing the existing representation. Thus the strategy does not only become different, but *more elaborate* (figure 1).

This idea is implemented in a method for evolving increasingly complex neural networks, called NeuroEvolution of Augmenting Topologies (NEAT; Stanley and Miikkulainen 2002b,c,d). NEAT begins by evolving networks without any hidden nodes. Over many generations, new hidden nodes and connections are added, resulting in the complexification of the solution space. This way, more complex strategies elaborate on simpler strategies, focusing search on solutions that are likely to maintain existing capabilities. We use NEAT to demonstrate the power of complexification.

NEAT was tested in a competitive robot control domain with and without complexification. Coevolution was used to evolve robot controllers against each other to find better strategies. We chose this domain because it is open-ended; there is no known optimal strategy but it is possible to come up with increasingly more sophisticated strategies indefinitely. The main results were that (1) evolution did complexify when possible, (2) complexification led to elaboration, and (3) significantly more sophisticated and successful strategies were evolved with complexification than without. These results imply that complexification allows coevolution to continually elaborate on successful strategies, resulting in an arms race that achieves a

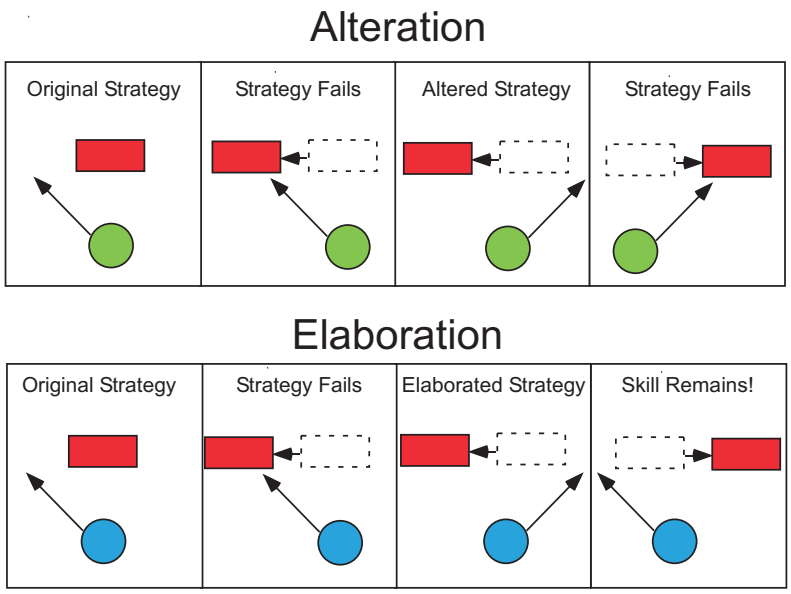

Figure 1: **Alteration vs. elaboration example.** A robot (depicted as a circle) evolves to avoid an obstacle. In the alteration scenario (top), the robot first evolves a strategy to go around the left side of the obstacle. However, the strategy fails in a future generation when the obstacle begins moving to the left. Thus, the robot alters its strategy by evolving the tendency to move right instead of left. However, when the obstacle later moves right, the new, altered, strategy fails because the robot did not retain its old ability to move left. In the elaboration scenario (bottom), the original strategy of moving left also fails. However, instead of altering the strategy, it is *elaborated* by adding a new ability to move right as well. Thus, when the obstacle later moves right, the robot still has the ability to avoid it by using its original strategy. Elaboration is necessary for a coevolutionary arms race to emerge and it can be achieved through complexification.

significantly higher level of sophistication than is otherwise possible.

We begin by reviewing biological support for complexification, as well as past work in coevolution, followed by a description of the NEAT method, and experimental results.

2 Background

2.1 Complexification in Nature

Mutation in nature not only results in optimizing existing structures: New genes are occasionally added to the genome, allowing evolution to perform a complexifying function over and above optimization. In addition, complexification is protected in nature in that interspecific mating is prohibited. Such speciation creates important dynamics differing from standard GAs. In this section, we discuss these important characteristics of natural evolution as a basis for our approach to utilize them computationally in genetic algorithms.

Gene duplication is a special kind of mutation in which one or more parental genes are copied into an offspring's genome more than once. The offspring then has redundant genes expressing the same proteins. Gene duplication has been responsible for key innovations in overall body morphology over the course of natural evolution (Amores et al. 1998; Carroll 1995; Force et al. 1999; Martin 1999).

A major gene duplication event occurred around the time that vertebrates separated from invertebrates. The evidence for this duplication centers around *HOX genes*, which determine the fate of cells along the

anterior-posterior axis of embryos. HOX genes are crucial in shaping the overall pattern of developmental in embryos. In fact, differences in HOX gene regulation explain a great deal of arthropod and tetrapod diversity (Carroll 1995). Amores et al. (1998) explain that since invertebrates have a single HOX cluster while vertebrates have four, cluster duplication must have significantly contributed to elaborations in vertebrate body-plans. The additional HOX genes took on new roles in regulating how vertebrate anterior-posterior axis develops, considerably increasing body-plan complexity. Although Martin (1999) argues that the additional clusters can be explained by many single gene duplications accumulating over generations, as opposed to massive whole-genome duplications, researchers agree that gene duplication contributed significantly to important body-plan elaboration.

A detailed account of how duplicate genes can take on novel roles was given by Force et al. (1999): Base pair mutations in the generations following duplication *partition* the initially redundant regulatory roles of genes into separate classes. Thus, the embryo develops in the same way, but the genes that determine overall body-plan are confined to more specific roles, since there are more of them. The partitioning phase completes when redundant clusters of genes are separated enough that they no longer produce identical proteins at the same time. After partitioning, mutations within the duplicated cluster of genes alter different steps in development than mutations within the original cluster. In other words, duplication creates more points at which mutations can occur. In this way, developmental processes complexify.

In order to implement this idea in artificial evolutionary systems we are faced with two major challenges. First, such systems evolve variable-length genomes, which can be difficult to cross over without losing information. For example, depending on when duplications occurred in the ancestral histories of two different genomes, the same gene may exist at different positions. Conversely, different genes may exist at the same position. Thus, artificial crossover may lose essential genes through misalignment. Second, it may be difficult for a variable-length genome GA to find innovative solutions; Optimizing many genes takes longer than optimizing only a few, meaning that more complex genotypes may be eliminated from the population before they have a sufficient opportunity to be optimized.

How has nature solved these problems? First, nature has a mechanism for aligning genes with their proper counterparts during crossover, so that data is not lost nor obscured. This alignment process has been most clearly observed in *E. coli* (Radding 1982; Sigal and Alberts 1972). A special protein called *RecA* takes a single strand of DNA and aligns it with another strand by attaching the strands at genes that express the same traits, which are called *homologous genes*. The process by which RecA protein aligns homologous genes is called *synapsis*. In experiments in vitro, researchers have found that RecA protein does not complete the process of synapsis on fragments of DNA that are not homologous (Radding 1982). Second, organisms with significantly different genomes never mate because they are in different species. Speciation also protects innovation, because it allows organisms to compete primarily within their own niches, instead of with the population at large.

It turns out complexification also possible in evolutionary computation if abstractions of synapsis and speciation are made part of the genetic algorithm. The NEAT method (section 3) is an implementation of this idea: the genome is complexified by adding new genes which in turn encode new structure in the phenotype, as in biological evolution.

Complexification is especially powerful in open-ended domains where the goal is to continually generate more sophisticated strategies. Competitive coevolution is a particularly important such domain, as will be reviewed in the next section.

2.2 Competitive Coevolution

In competitive coevolution, individual fitness is evaluated through direct competition with other individuals in the population, rather than through an objective fitness measure. In other words, fitness signifies only the relative strengths of solutions; an increased fitness in one solution leads to a decreased fitness for another. Ideally, competing solutions will continually outdo one another, leading to an "arms race" of increasing sophistication (Dawkins and Krebs 1979; Rosin 1997; Van Valin 1973).

In practice, it is difficult to establish an arms race. Evolution tends to find the simplest solutions that can win, meaning that strategies can switch back and forth between different idiosyncratic yet uninteresting variations (Darwen 1996; Floreano and Nolfi 1997; Rosin and Belew 1997). Several methods have been developed to encourage the arms race (Angeline and Pollack 1993; Ficici and Pollack 2001; Noble and Watson 2001; Rosin and Belew 1997). For example, a "hall of fame" can be used to ensure that current strategies remain competitive against strategies from the past. Recently, Ficici and Pollack (2001) and Noble and Watson (2001) introduced a promising method called *Pareto coevolution*, which finds the best learners and the best teachers in two populations by casting coevolution as a multiobjective optimization problem.

Although such techniques improve the performance of competitive coevolution, they do not directly encourage *continual coevolution*, i.e. creating new solutions that maintain existing capabilities. For example, no matter how well selection is performed, or how well competitors are chosen, if no better solution exists in the fixed solution space, elaboration is impossible since the global optimum has already been reached. Moreover, it may occasionally be easier to escape a local optimum by adding a new dimension of freedom to the search space than by jumping back out into the same space that lead to the optimum in the first place.

Complexification is an ideal technique for establishing a coevolutionary arms race. Complexification naturally elaborates strategies by adding new dimensions to the search space. Thus, progress can be made indefinitely long: even if a global optimum is reached in the search space of solutions, new dimensions can be added, opening up a higher-dimensional space in which even higher optima may exist.

To test this idea experimentally, we chose a robot duel domain that combines predator/prey interaction and food foraging in a novel head-to-head competition (Section 4). We use this domain to demonstrate how NEAT uses complexification to continually elaborate on strategies. The next section describes the NEAT neuroevolution method, followed by a description of the robot duel domain and a discussion of the results.

3 NeuroEvolution of Augmenting Topologies (NEAT)

The NEAT method of evolving artificial neural networks combines the usual search for appropriate network weights with complexification of the network structure. This approach is highly effective: NEAT outperforms other neuroevolution (NE) methods, e.g. on the benchmark double pole balancing task by a factor of five (Stanley and Miikkulainen 2002b,c,d). The NEAT method consists of solutions to three fundamental challenges in evolving neural network topology: (1) What kind of genetic representation would allow disparate topologies to crossover in a meaningful way? Our solution is to use historical markings to line up genes with the same origin. (2) How can topological innovation that needs a few generations to optimize be protected so that it does not disappear from the population prematurely? Our solution is to separate each innovation into a different species. (3) How can topologies be minimized *throughout evolution* so the most efficient solutions will be discovered? Our solution is to start from a minimal structure and grow only when necessary. In this section, we explain how NEAT addresses each challenge.