Major Evolutionary Transitions in the Voxelbuild Virtual Sandbox Game

In: Proceedings of the European Conference on Artificial Life (ECAL 2017). Cambridge, MA: MIT Press.

Justin K. Pugh, L. B. Soros, Rafaela Frota, Kevin Negy, and Kenneth O. Stanley

University of Central Florida, Orlando, FL 32816-2362
jpugh@cs.ucf.edu, lisa.soros@gmail.com, rafaelalfrota@gmail.com, kevinnegy@gmail.com, kstanley@cs.ucf.edu

Abstract

Developing a comprehensive theory of open-ended evolution (OEE) depends critically on understanding the mechanisms underlying the major evolutionary transitions; such periods of rapid innovation, such as the Cambrian explosion, have resulted in exactly the kind of diversity and complexity deemed the hallmarks of strong OEE. This paper introduces a new domain for studying major transitions in an evolutionary robotics context. Inspired by the popular Minecraft video game, the new Voxelbuild domain centers on agents that evolve the capacity to build arbitrarily complex block structures with minimal objectives. Initial experiments demonstrate both the rich expressive potential of the new domain and, intriguingly, the occurrence of major evolutionary transitions in at least some runs, thereby providing a unique opportunity to probe how and why such transitions occur or fail to occur in different runs of the same system.

Introduction

The emergence of complexity is arguably the most impressive and elusive feature of natural evolution. What *exactly* spurs the growth of functional organization from chaos and simplicity? Szathmáry and Maynard Smith (1995) identify *major transitions* to help describe and explain the general increase in complexity observed in biological evolution. These transitions, including the development of eukaryotes from prokaryotes and later the development of human societies bound by language, deal primarily with changes in the encoding and transmission of information, leading eventually to increased complexity at the phenotypic level (i.e. tightly-bound aggregate entities composed of simpler parts). This phenomenon of complexity-increasing evolutionary mechanisms is central to the idea of open-ended evolution (OEE), which roughly corresponds to indefinitely scalable evolutionary processes. In fact, the presence of a major evolutionary transition delineates *weak* from *strong* OEE (de Vladar et al., 2017). While artificial life worlds have demonstrated the capacity for at least *some* features of OEE, understanding major evolutionary transitions, and the mechanisms and conditions that enable them, remains an important open problem for OEE (Bedau et al., 2000; Taylor et al., 2016).

Over 20 years have passed since Szathmáry and Maynard Smith's publication on the topic, and in that time the artificial life and OEE communities have undergone substantial change. Two important perspective shifts have been gaining momentum in recent years: (1) there may be many different kinds or degrees of open-ended evolution, and (2) open-ended evolution (of some kinds) may be observed in non-biological domains (such as some artificial life worlds and even high-level real world systems such as technological innovation) as well as in traditional biologically-inspired systems (Taylor et al., 2016). In their original formulation of the major transitions, Szathmáry and Maynard Smith focus on heredity because it is the biological means for information to be stored and transmitted. To study artificial life systems, the concept of the major transitions can thus be beneficially defined more broadly and may even take forms deviating largely from that of biological evolution. For this reason, it makes sense to broaden our interpretation of major transitions to admit mechanisms leading to indefinitely increasing complexity in non-biological domains as well. That is, it is conceivable that in other, non-biological domains, the major transitions that lead to an inevitable increase in complexity might be defined in a different manner.

Accordingly, this paper follows the alternative model of major transitions proposed by Koonin (2007) wherein major transitions are characterized as "brief bursts of innovation", or a rapid diversification of novel and complex forms. This formulation turns out useful for studying evolution in a cross-disciplinary context (which is ideal for artificial life) because it admits comparisons of complexity-increasing transitions in systems that are both biologically inspired and not.

One obstacle to reproducing the phenomenon of major transitions in biological systems is the sheer amount of time required. The Cambrian explosion, for instance, occurred over 3 billion years after the development of the first cell on Earth (Marshall, 2006). Another difficulty arises from the lack of controllability of complex biological systems. Artificial life systems, in contrast, allow us to perform both tractable and controllable experiments and thereby test with rigor scientific theories about major open problems in evolutionary theory.

However, not all simulations have the capacity to express the levels of functional organization considered to be the hallmarks of complexity. This paper introduces a novel domain called Voxelbuild that, when coupled with an effective evolutionary framework (in this paper, *quality diversity* (Pugh et al., 2015, 2016a,b)) does exhibit a major transition in at least some runs. Interestingly, not *all* runs exhibit a major transition, resulting in a promising environment for studying both the success and failure of hypothesized prerequisites for such transitions.

Background

Achieving open-ended evolution has challenged the artificial life community since the field's inception. Open-endedness was initially viewed as a detectable property of a certain class of evolutionary systems; a system can exhibit a capacity for ongoing innovation on the order of biological evolution (which has historically been the inspiration for artificial open-ended systems) or something less productive. This perspective is reflected in the activity statistics test that aims to detect signatures of open-endedness (Bedau et al., 1998; Channon, 2003). In recent years, however, a more pluralistic view of OEE has also emerged, admitting a wider variety of *kinds* of open-endedness (Taylor et al., 2016).

Whatever perspective one takes, it seems that no artificial evolutionary system to date has displayed the capacity for open-endedness observed in biological and physical systems (de Vladar et al., 2017). Digital evolution systems such as Tierra (Ray, 1992) and Avida (Ofria and Wilke, 2004), in which code-based lifeforms compete for computational resources on a minimal virtual computer, have proved fruitful for the purposes of studying evolutionary dynamics in a non-biological context. For example, Lenski et al. (2003) demonstrate how digital organisms in Avida can learn to perform complex logic functions only when some of the simpler functions that comprise them provide some evolutionary advantage. However, systems such as Tierra and Avida have not yet produced definitive explosions of complexity that might be expected of OEE. Embodied systems such as the cell-based world Chimera (which in some configurations exhibits phase transitions to primitive multicellularity) (Solé and Valverde, 2013) and the block-creature worlds of Division Blocks (Spector et al., 2007), Evosphere (Miconi and Channon, 2005), and the seminal work of Karl Sims (Sims, 1994) offer great expressive potential through the individuals and the environment they inhabit. Yet there too a definitive open-ended result remains elusive.

It is also difficult to assess most systems' potential for major transitions simply because most published experiments run for a relatively short amount of time (at least compared to evolution on Earth). The next section introduces a novel experimental platform designed to support definitive transitions that can be identified and tracked.

Voxelbuild

For the purpose of studying evolutionary transitions and long-term/open-ended evolution in the context of evolutionary robotics, this paper introduces a new experimental platform called *Voxelbuild*, inspired by the Minecraft [1] video game that is well-known for facilitating seemingly boundless creativity. Voxelbuild, like Minecraft, serves as a sandbox in which embodied agents can build structures by placing and removing blocks in a discrete three-dimensional grid-based world. However, the Voxelbuild world is greatly simplified from that of Minecraft to facilitate faster evaluation and a smaller genetic encoding. Additionally, the world imposes the following physical constraints: (1) agents are subject to gravity (i.e. they cannot fly), (2) agents cannot walk through solid blocks, (3) agents may jump on top of blocks, but jump height is limited to one block, (4) blocks can only be placed adjacent to other blocks (however, there are no constraints on block removal and blocks are not subject to gravity, so it is possible to create "floating" structures by building upwards and then removing some of the blocks underneath)[2], and (5) block placement is limited to a small radius around the agent (in this paper, the radius is 5). Subject to these constraints, building all but the simplest structures is a non-trivial task; especially difficult is building upwards because the only way to reach heights outside the block placement radius is to stand on previously placed blocks (i.e. *scaffolding*).

In this paper the world size is bounded for computational efficiency to $21 \times 21 \times 12$ blocks. At the beginning of each evaluation, the environment contains only a single layer of grassy blocks. The agent is placed in the center of this environment and allowed up to 10,000 ticks to perform combinations of six discrete actions: (1) turn left 90 degrees, (2) turn right 90 degrees, (3) move forward, (4) move backward, (5) place a crate block, or (6) remove a grassy block or crate block. Importantly, a trial is terminated early if the agent requests an illegal move or enters a loop (e.g. moving back and forth or spinning in a circle); in this way, actual trial length varies widely according to how well each agent obeys the physical constraints of the world.

Agent Configuration

The discrete nature of time and space in Voxelbuild facilitates controlling agents through artificial means. In this paper, agents are controlled by evolved neural networks. Agents sense the world around them with an $11 \times 11 \times 11$ array of block sensors (1,331 sensors total), which extends as far as the maximum distance for block placement and removal (distance 5). Sensor values are determined according to the type of block present at the corresponding location (1 – crate or grassy block, 0 – boundary block, -1 – no block). The

[1]Copyright (c) 2011 Mojang

[2]Unlike in Minecraft, block placement and removal is not restricted by line-of-sight, although this restriction can be added to increase problem difficulty.

control network takes the block sensor values as input and outputs six action selector values where the highest value determines which action is taken by the agent. Ties between action selector values are resolved in order of the following priority: move forward, move backward, remove block, place block, turn left, turn right. In the case of the *remove block* and *place block* actions, the block location is decided by an $11 \times 11 \times 11$ output array (ties are resolved deterministically with preference for closer locations). The exact network configuration including hidden layers is depicted in figure 1.

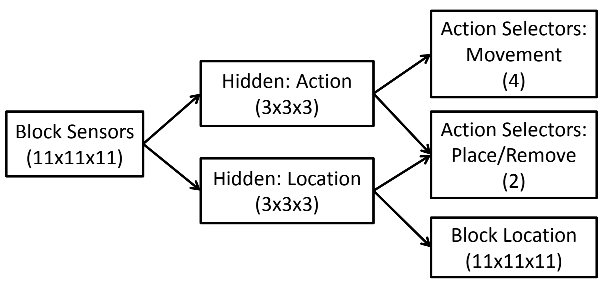

Figure 1: **Network configuration.** All connected layers are fully connected: there are 108,027 connections in total. Connection weights are evolved with HyperNEAT. Neurons in each neuron group are spread evenly across three dimensions bounded between -1 and 1 on each axis.

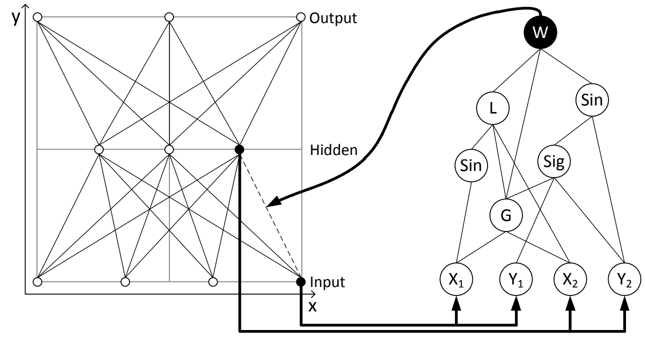

Figure 2: **HyperNEAT example.** Neurons in the example substrate (left) are assigned (x, y) coordinates. For each connection in the substrate, the NEAT-evolved CPPN (right) takes the coordinates of the source and target neuron as inputs and outputs the connection weight. While this example substrate is two-dimensional, HyperNEAT can easily be extended to support three or more dimensions by adding two additional coordinate inputs per extra dimension (e.g. substrates in this paper are three-dimensional and thus have six coordinate inputs). To express regularities such as symmetry and repetion, neurons in the evolved CPPN can have activation functions other than the usual sigmoid (Sig) such as Gaussian (G), linear (L), or sine (Sin).

Approach

The approaches in this section combine in Voxelbuild to allow a diversity of block placement strategies to evolve.

HyperNEAT

Agent control networks in this paper are evolved by HyperNEAT (Stanley et al., 2009; Gauci and Stanley, 2010), which is itself an extension of the popular NEAT (Neuroevolution of Augmenting Topologies) (Stanley and Miikkulainen, 2002) method for evolving neural networks. NEAT networks begin with minimal complexity and gradually increase in complexity over time by adding neurons and connections through random mutations. Unlike NEAT, HyperNEAT is an *indirect encoding* wherein connection weights in a *substrate* are decided by a *compositional pattern producing network* (CPPN) (Stanley, 2007) that is itself evolved by NEAT. Figure 2 depicts an example of how HyperNEAT determines connection weights in the substrate. Through its indirect encoding, HyperNEAT is able to evolve large networks with hundreds of thousands or even millions of connections, making it ideal for evolving Voxelbuild controllers, which have over 100,000 connections. HyperNEAT neurons are embedded in a three-dimensional substrate at locations that correspond to the relative locations of each of their respective sensors and effectors on the agent. In this way, HyperNEAT takes advantage of problem geometry (e.g. the geometric relationship between block sensors) and evolved networks display biologically important regularities such as symmetry and repetition.

In this paper, HyperNEAT networks (figure 1) are embedded in a multi-spatial substrate (Pugh and Stanley, 2013), which means that each functionally distinct neuron group exists in a separate three-dimensional coordinate space and connection weights between each connected pair of neuron groups are determined by a separate CPPN output. Multi-spatial substrates allow multiple sensor and effector modalities to coexist in the same substrate without imposing geometric relationships between them.

Novelty Search with Local Competition

While traditionally evolutionary algorithms are driven by incremental progress on a heuristic fitness function, it is well-known that, through a phenomena called *deception*, following the path of increasing fitness does not always lead to the best possible performance (Goldberg, 1987). Even when evolution is pursued more organically as a competition among interacting individuals for scarce resources (a practice common in artificial life systems), a deceptive fitness gradient may be imposed. An alternative evolutionary algorithm called *novelty search* (NS) (Lehman and Stanley, 2008, 2011a) attempts to circumvent deceptive fitness landscapes by ignoring the fitness function altogether, instead searching for novelty in a space of possible behaviors (a *behavior* is a vector of numbers that captures some aspects of how an agent acted during its trial). Novelty search proceeds by maintaining an archive of previously-encountered behaviors, and new genomes are rewarded according to how different their behavior is from those stored in the archive, with the most novel genomes receiving the highest scores.

NS alone is powerful for exploring a space of possible behaviors (e.g. finding many different kinds of structures in Voxelbuild). However, because it ignores fitness altogether, NS does not necessarily find the best possible structures at each region of structure space.

A new class of evolutionary algorithms, called *quality diversity* algorithms, attempt to find a wide diversity of highly performing individuals, which requires a careful balance between the drive for novelty and standard fitness pressure. One such QD algorithm, called *novelty search with local competition* (NSLC) (Lehman and Stanley, 2011b) achieves this balance by allowing fitness competition only within local niches, while simultaneously exploring as broadly as possible across the space of possible behaviors. Specifically, NSLC is a multiobjective formulation where one objective is novelty and the other objective is performance relative to a local behavioral neighborhood. In this way, NSLC discovers strong candidates in all regions of the behavior space, *even those regions that are lower-performing than others*.

Grid-free QD Collection

While NSLC encounters a diversity of high-performing individuals over the course of evolution, it is often infeasible to simply return everything that is discovered (e.g. over a long run that evaluates hundreds of thousands of individuals). Instead, a set of results called the *QD collection* should be returned to the user, where the QD collection represents the best performers across the entire behavior space. However, there are different ways to best choose which individuals to include in this collection.

When the dimensionality of behavior vectors is low (under 10), it is possible to divide the entire behavior space into discrete bins and simply remember the best performing individual encountered in each bin (Pugh et al., 2015, 2016b) (this idea is central to a QD algorithm called MAP-Elites (Mouret and Clune, 2015; Cully et al., 2015)). However, for higher dimensional behavior vectors (such as those featured in this paper), this strategy is impractical or even impossible because the number of bins grows exponentially with the dimensionality of the behavior vector.

Several strategies have been explored in the literature for returning a QD collection without discretizing the behavior space. The simplest such strategy is to return the population or the novelty archive at the end of the run (Lehman and Stanley, 2011b). However, the population or archive at any given point in a run (including the last generation) is not necessarily representative of everything that has been explored over the entire run.

An alternate strategy is to pare down the dimensionality of the behavior vectors post-hoc via principal component analysis (PCA) (Szerlip and Stanley, 2013). However, PCA does not always work well for high dimensional vectors (Johnstone and Lu, 2009); additionally, this strategy requires saving all individuals encountered during a run which may be prohibitive because of memory or disk space limitations.

Cully and Mouret (2013) introduces a variant of NSLC's novelty archive wherein archive members are continually replaced by better-performing individuals as they are encountered during search. One concern with this approach is that polluting the archive with fitness pressure may leave NSLC subject to deception and thus interfere with its ability to explore the behavior space.

This paper modifies the original idea of the behavioral repertoire so that the QD collection does not interfere with evolution. This process, called *grid-free QD collection*, proceeds by maintaining a small QD collection on the side that will serve as the set of results to return to the user. As individuals are encountered in evolution, they participate in *insertion tournaments* to decide whether they enter the QD collection. For every new individual encountered during evolution, a tournament is held between the new individual and a number of randomly selected members of the QD collection. The tournament participants each calculate their novelty scores against the QD collection in the same way that NS computes novelty against the archive: by calculating their behavioral distance against all members of the QD collection then summing the 20 smallest distances. Then, the participant with the lowest novelty score compares its fitness against its closest neighbor and the individual with higher fitness is allowed into the QD collection while the lower fitness individual is removed or discarded. In this way, the QD collection continually expands its coverage of the behavior space while simultaneously *locally* improving fitness. This approach is similar to the behavioral repertoire in Cully and Mouret (2013) except that the QD collection here runs in the background and does not interfere with evolution in any way.

Experiment

The world of Voxelbuild is designed to facilitate experiments in open-ended creativity through the evolution of intelligent agents that construct diverse and complex structures. Of particular interest in this paper is the phenomenon of *evolutionary transitions* wherein skills are acquired by the gene pool that enable the production of fundamentally different and more complex artifacts. To study such transitions, it is necessary that the task contains a substantial level of difficulty such that major transitions are observed in some, but not all, evolutionary runs. That way successful runs can be compared against unsuccessful runs to identify how and why these transitions occur. Thus, building in Voxelbuild is designed to be nontrivial – agents are bound by the physical constraints of the world and must learn how to move and act within the world without violating its rules.

While Voxelbuild supports the construction of an effectively endless array of unique structural artifacts, not all structures are possible without first acquiring certain fundamental skills, each of which corresponds to a major evolutionary transition. During preliminary runs, it was observed that agents have a difficult time building vertically; more accurately, some runs find agents that learn vertical building and thus achieve better results, while agents in other runs continue building only at ground level. Considering only the physical constraints of the world, two major evolutionary transitions in Voxelbuild are conceivable, each of which opens the door to building increasingly complex structures. The first such transition is *vertical building*. Before learning vertical building, agents effectively build two-dimensional (ground-level) artifacts. Once agents learn to place blocks at heights above ground level, new types of structures become possible that utilize all three dimensions. The second major evolutionary transition stems from agents' inability to place blocks outside of a small local radius. Faced with this restriction, it is impossible for agents to build structures taller than their block placement radius without first moving on top of a previously placed block. The process of placing a block and then moving on top of it is called *scaffolding*, a major evolutionary transition that enables agents to build structures of unlimited height. Because blocks must be placed adjacent to other blocks and agents cannot jump higher than one block at a time, climbing via scaffolding is an advanced behavior requiring careful coordination. Besides vertical building and scaffolding, there may be still more major transitions in Voxelbuild that enable the building of different kinds of structures not possible in the pre-transition landscape.

Investigating the presence of and circumstances surrounding evolutionary transitions requires several runs of evolution where some observe transitions and others do not. In this experiment, 10 runs of evolution are given the same amount of CPU time over a period of two weeks on a 160-core computing cluster. Due to the large variance in trial duration (trials may last up to 10,000 ticks, but most terminate in under 200 ticks due to agents attempting illegal actions such as moving through a wall), there is a corresponding variance in the number of evaluations completed by each run.

Experiment Parameters

Voxelbuild agents are evolved by HyperNEAT[3] with the following CPPN mutation rates: 5% add connection, 5% delete connection, 1% add neuron, 1% delete neuron. NSLC population size is 128 with a batch size of 32, where the lowest Pareto rank fitness individuals are deleted from the population after each batch. Maximum novelty archive size is 512, enforced by drawing random tournaments of size 3 and deleting the participant with the lowest novelty score. The QD collection holds 128 individuals and is dumped to file every 1,000 epochs (32,000 evaluations), allowing pre-transition collections to be compared to those after a transition.

[3]The HyperNEAT implementation in this paper is a modified version of SharpNEAT 1.0 (Green, 2006).

Fitness and Behavior Characterization

As a QD algorithm, NSLC requires both a *fitness function* to measure the quality of individuals and a *behavior characterization* that describes which aspects of an individual should be considered when calculating novelty. Fitness is calculated as follows, where b_{net} is the total number of blocks placed minus the number of blocks removed and h_{max} is the maximum height at which a block is placed:

$$\text{fitness} = b_{net} \times h_{max} \ .$$

This formulation rewards larger structures while emphasizing those that utilize the z-axis so that taller structures correspond to substantially higher scores. This approach makes it easy to identify when vertical building or scaffolding transitions occur but does not necessarily reflect the *quality* of structures. Behaviors are characterized by a vector of 5,292 values for each location in the final state of the world, where a value of 1 means a block exists at the corresponding location and 0 means no block exists at that location.

Results

To illustrate when major evolutionary transitions occur, Figure 3 graphs the best fitness over the course of ten separate runs of evolution, where transitions correspond to abrupt increases in fitness caused by agents learning to build vertically. In two runs, vertical building is learned early (before the first QD collection dump at 32,000 evaluations). However, in two other runs, vertical building is learned later, allowing the state of the QD collection to be examined both before and after the evolutionary transition occurs. Six runs do not successfully transition in the given time frame, indicating the difficulty of transitioning.

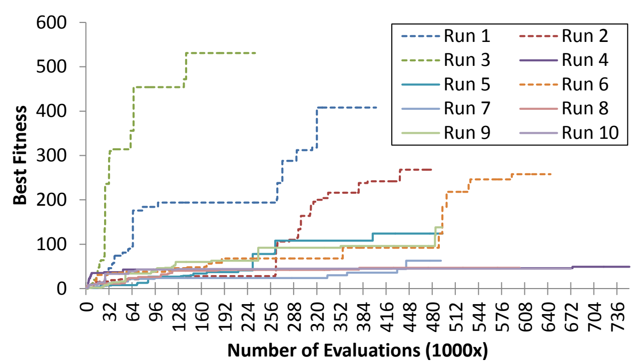

Figure 3: **Best fitness over time.** Graphing the best fitness over time for each run allows isolating when major transitions occur. High scores (above 200-250) correspond to achieving the vertical building transition. Out of the four successful runs (denoted by dashed lines), two runs transition early, while two other runs transition later. Tick marks on the x-axis indicate when the QD collection is dumped to file.

Upon examination of the QD collections before and after major transitions are observed, a clear pattern of progression emerges, consistent across all runs.

Agents first learn to build *chaotic structures* (figure 4), where blocks are placed haphazardly. These structures are often quite small, suggesting that the lack of organization makes it difficult for agents to survive very long without performing an illegal move. Unsuccessful runs never progress beyond chaotic structures.

Runs that eventually succeed in learning vertical building (a major evolutionary transition) first learn to place blocks in an organized manner (figure 5). In these structures, blocks are placed in straight lines or rectangles, with fewer gaps and less apparent randomness.

Four out of ten runs eventually gain the ability to build vertically. Structures at this point continue to exhibit a high degree of organization, facilitating the construction of larger and more complex structures with features such as repeated motifs and some blocks placed above ground level (figure 6). Sometimes, agents employ primitive forms of scaffolding when placing blocks above ground level (e.g. agents build a large sheet at ground level, then stand on top of it to continue building on the next level up).

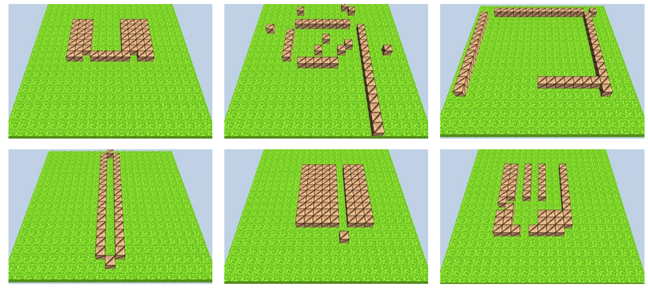

Figure 5: **Organized structures.** Primitive organization is marked by placing blocks in straight lines or rectangular shapes. This building strategy is the precursor for more advanced structures.

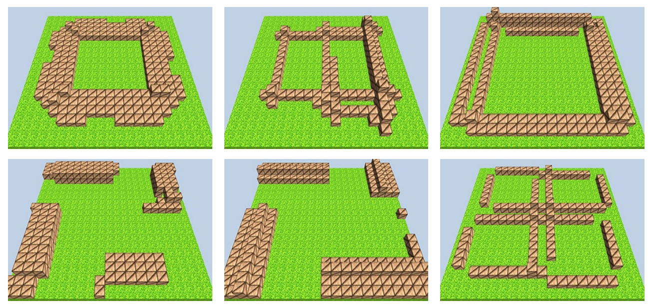

Figure 6: **Post-transition structures.** After learning organized block placement, larger and more complex structures become possible, including the ability to build vertically. In these examples, some blocks are placed at heights 2 and 3.

Discussion

Initial experiments reveal the first major evolutionary transitions in the world, the ability to place blocks vertically, preceded by an apparently necessary yet unanticipated preliminary transition: the discovery of organization. While vertical building is specific to Voxelbuild, its precursor, organized building, captures a principle that may generalize to other open-ended simulations and artificial life worlds: *organization precedes complexity*. Chaos in such simulations is sometimes celebrated as a sign of complexity (when we do not understand exactly what is going on in a system, we may assume there are advanced strategies at play), but it is possible that in many cases chaos is actually a sign of evolutionary stagnation. In Voxelbuild, all runs begin with agents that place blocks haphazardly and no evolutionary trajectories are observed where chaotic building leads directly into vertical building, suggesting that organized behavior is an important *stepping stone* to more advanced behaviors. Similarly, deliberate and principled behavior, while sometimes less exciting to observe, may eventually lead to higher levels of complexity in other open-ended domains.

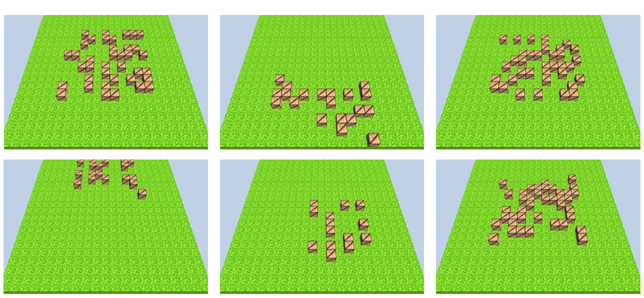

Figure 4: **Chaotic structures.** Chaotic structures are marked by haphazard, disorganized block placement and are the first types of structures that agents learn to build.

In Voxelbuild, agents must operate within the physical constraints of their world, which includes the stipulation that blocks must be placed adjacent to other blocks. Placing blocks at ground level easily satisfies this constraint because every location at height 1 is adjacent to the blocks that constitute the ground (at height 0). However, all positions at height 2 are initially illegal, requiring first placing a block nearby (usually underneath). Agents that place blocks seemingly at random (i.e. chaotic placement) have a difficult time building upwards because they have not learned how to satisfy the adjacency constraint. On the other hand, organized placement is characterized by contiguous lines and rectangles, signifying that agents have learned to place blocks adjacent to each other at ground level and thus they also have an easier time satisfying adjacency in the vertical direction.

The principle of organization preceding complexity can be observed in nature. For example, early forms of life consisted of *prokaryotes* – single-celled organisms whose components are not membrane-bound but rather all float together in the cytoplasm. Later in Earth's evolutionary timeline, *eukary-*

otes (organisms with cells whose specialized components are clearly separated into organelles by thin membranes) began to appear (Ridley, 2004). The separation of cellular components into functionally distinct organelles represents an advancement in organization that perhaps opened the door to multicellular organisms and in turn to the diversity of complex life that we observe today.

Interestingly, de Vladar et al. (2017) characterize innovations as "the population stepping into pre-existing but unoccupied dimensions of the embedding trait space", echoing Kauffman's theory of the ever-expanding *adjacent possible* (Kauffman, 2000). They further suggest that innovations may occur concomitantly with "new ways of interacting with the environment." The observed evolved behavior of constructing straight lines and platforms (the latter of which sometimes functions as a primitive form of scaffolding) to build larger and more complex structures clearly constitutes a new way of interacting with the environment, thereby qualifying as an innovation in this sense. Furthermore, the rapid diversification and complexification of structures built following this particular innovation indicates that it signifies a major evolutionary transition, opening the door for new kinds of behaviors that were previously difficult or impossible.

As in Voxelbuild, the evolution and complexification of life on Earth is often characterized by a stepwise (not gradual) pattern (Eldredge and Gould, 1972; Schuster, 2016). In fact, de Vladar et al. (2017) argues that such steps, each corresponding to major transitions, are impossible to define completely a priori because many innovations arise from *exaptations* (traits that are evolved for one purpose but later used for another purpose). In Voxelbuild, the appearance of primitive scaffolding is itself an exaptation because it is not evolved for any particular purpose but will be necessary later for building structures more than six blocks tall. Framed another way, the nondeterminism and nonalgorithmic nature of evolution may make it impossible to guarantee any particular outcome in sufficiently high-dimensional spaces (though it is also argued by de Vladar et al. (2017) that we can get close enough for practical purposes). Given this inherent lack of strict order, it is not surprising that only some runs exhibited the organized style of building that is prerequisite to the vertical building major transition.

In open-ended evolution, which strives for an effectively endless progression of meaningful complexification, major transitions represent evolutionary bottlenecks where progress may seem to stall. However, these bottlenecks are to be expected and may even be necessary for many of the most important innovations. In Voxelbuild, as on Earth, the road to more advanced behaviors and artifacts is marked not by incremental progress, but rather by sudden bursts of development following the discovery of important traits that introduce fundamentally new ways of interacting with the world. With this in mind, it is likely inappropriate to drive OEE solely by some concrete measure of fitness; while chasing ever-increasing fitness may be suitable for adaptation, it does little to encourage exaptations – the impetus for true innovation. Instead, chasing adaptations towards a predefined target may actually take search in the wrong direction, where it can become trapped in an evolutionary dead end (a phenomenon commonly known in the evolutionary computation community as *deception*). In fact, the very presence of a fitness function generally assumes that there is an end goal in mind. However, true OEE should continuously generate interesting new artifacts and behaviors beyond what can be conceived of a priori. Just as a silent observer of Earth four billion years ago would not have been able to imagine the eventual development of modern plant and animal life, open-ended evolution should not constrain the search process with preconceptions about what is possible or desirable, but rather should leave evolution free to explore and discover, not strictly pursuing what is regarded as "better" but instead pursuing what is different.

In this initial study into evolutionary transitions in Voxelbuild, some runs succeed in transitioning in the allotted time frame, while others do not. This raises an important question: *given more time, will unsuccessful runs eventually reach a major transition, or are they stuck forever in the space of simple behaviors?* Relatedly: *given more time, will successfully transitioned runs continue on to cross the threshold of still more major transitions, thus producing increasingly complex and interesting behaviors?* To answer these questions, we must consider that evolutionary runtime is limited by available computation power. However, in this experiment as in many others, the bulk of computational power is spent running multiple rounds of evolution in parallel to be sure that at least some successful runs are observed, severely limiting the duration of each individual run. Studies like this one of the circumstances surrounding evolutionary transitions, especially of the factors that either promote or impede innovation, are therefore important because they may inspire new approaches that can more reliably achieve such milestones. Then it becomes more realistic to devote with confidence all available computation to a single evolutionary run, allowing substantially deeper exploration into the hidden possibilities of open-ended domains such as Voxelbuild.

Conclusion

This paper introduces a new domain for open-ended evolution (OEE) called Voxelbuild, a world in which embodied agents move and build structures out of blocks, similarly to in the popular Minecraft video game. Initial experiments in this domain reveal the presence of major evolutionary transitions where agents learn fundamentally new ways of interacting with the world, thus unlocking more complex behaviors than were previously possible. Evolutionary transitions can be found in Earth's evolutionary timeline – most notably in the Cambrian explosion, where primitive multicellular life rapidly developed into the vast diversity of complex species observable today – and are likely an important hallmark of strong OEE. This work offers insight into such transitions and their evolutionary precursors, including some principles that may be shared across other OEE domains, such as the importance of organized behavior. A greater understanding of evolutionary transitions in OEE will potentially inform more sophisticated approaches that more reliably transition into progressively more complex behaviors, opening the door for longer and more interesting OEE experiments in the future.

Acknowledgements

This work was supported by the National Science Foundation under grant no. IIS-1421925. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Bedau, M. A., McCaskill, J. S., Packard, N. H., Rasmussen, S., Adami, C., Green, D. G., Ikegami, T., Kaneko, K., and Ray, T. S. (2000). Open problems in artificial life. *Artificial life*, 6(4):363–376.

Bedau, M. A., Snyder, E., and Packard, N. H. (1998). A classification of long-term evolutionary dynamics. In *Artificial life VI*, pages 228–237. Cambridge: MIT Press.

Channon, A. (2003). Improving and still passing the alife test: Component-normalised activity statistics classify evolution in geb as unbounded. *Artificial Life*, 8:173–181.

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, 521(7553):503–507.

Cully, A. and Mouret, J.-B. (2013). Behavioral repertoire learning in robotics. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, pages 175–182, New York, NY, USA. ACM.

de Vladar, H. P., Santos, M., and Szathmáry, E. (2017). Grand views of evolution. *Trends in Ecology & Evolution*.

Eldredge, N. and Gould, S. J. (1972). Punctuated equilibria: an alternative to phyletic gradualism. *Models in paleobiology*, 82:115.

Gauci, J. and Stanley, K. O. (2010). Autonomous evolution of topographic regularities in artificial neural networks. *Neural Computation*, 22(7):1860–1898.

Goldberg, D. E. (1987). Simple genetic algorithms and the minimal, deceptive problem. In Davis, L., editor, *Genetic algorithms and simulated annealing*, pages 74–88, London. Pitman.

Green, C. (2003–2006). SharpNEAT homepage.

Johnstone, I. M. and Lu, A. Y. (2009). On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682–693.

Kauffman, S. A. (2000). *Investigations*. Oxford University Press.

Koonin, E. V. (2007). The biological big bang model for the major transitions in evolution. *Biology Direct*, 2(1):21.

Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In Bullock, S., Noble, J., Watson, R., and Bedau, M., editors, *Proceedings of the 11th International Conference on Artificial Life*, Cambridge, MA. MIT Press.

Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.

Lehman, J. and Stanley, K. O. (2011b). Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pages 211–218. ACM.

Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936):139–144.

Marshall, C. R. (2006). Explaining the cambrian "explosion" of animals. *Annu. Rev. Earth Planet. Sci.*, 34:355–384.

Miconi, T. and Channon, A. (2005). A virtual creatures model for studies in artificial evolution. In *The 2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 565–572. IEEE.

Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Ofria, C. and Wilke, C. O. (2004). Avida: A software platform for research in computational evolutionary biology. *Artificial life*, 10(2):191–229.

Pugh, J. K., Soros, L., and Stanley, K. O. (2016a). Searching for quality diversity when diversity is unaligned with quality. In *International Conference on Parallel Problem Solving from Nature*, pages 880–889. Springer.

Pugh, J. K., Soros, L. B., and Stanley, K. O. (2016b). Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40.

Pugh, J. K., Soros, L. B., Szerlip, P. A., and Stanley, K. O. (2015). Confronting the challenge of quality diversity. In *Proceedings of the 17th Annual Conference on Genetic and Evolutionary Computation*, New York, NY, USA. ACM.

Pugh, J. K. and Stanley, K. O. (2013). Evolving multimodal controllers with HyperNEAT. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, pages 735–742, New York, NY, USA. ACM.

Ray, T. S. (1992). An approach to the synthesis of life. In *Proc. of Artificial Life II*, pages 371–408. Addison-Wesley.

Ridley, M. (2004). *Evolution, 3rd Edition*.

Schuster, P. (2016). Major transitions in evolution and in technology: What they have in common and where they differ. *Complexity*, 21(4):7–13.

Sims, K. (1994). Evolving 3D morphology and behavior by competition. pages 28–39. MIT Press, Cambridge, MA.

Solé, R. V. and Valverde, S. (2013). Before the endless forms: embodied model of transition from single cells to aggregates to ecosystem engineering. *PLoS ONE*, 8(4):e59664.

Spector, L., Klein, J., and Feinstein, M. (2007). Division blocks and the open-ended evolution of development, form, and behavior. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pages 316–323. ACM.

Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines Special Issue on Developmental Systems*, 8(2):131–162.

Stanley, K. O., D'Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based indirect encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212.

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

Szathmáry, E. and Maynard Smith, J. (1995). The major evolutionary transitions. *Nature*, 374(6519):227–232.

Szerlip, P. and Stanley, K. O. (2013). Indirectly encoded sodarace for artificial life. In *Proceedings of the European Conference on Artificial Life (ECAL-2013)*, volume 12, pages 218–225.

Taylor, T., Bedau, M., Channon, A., Ackley, D., Banzhaf, W., Beslon, G., Dolson, E., Froese, T., Hickinbotham, S., Ikegami, T., et al. (2016). Open-ended evolution: perspectives from the oee workshop in york. *Artificial Life*.