Selecting Continuous Life-Like Cellular Automata for Halting Unpredictability: Evolving for Abiogenesis

Q. Tyrell Davis  
qdavis@uvm.edu  
University of Vermont  
Burlington, Vermont, USA

Josh Bongard  
jbongard@uvm.edu  
University of Vermont  
Burlington, Vermont, USA

ABSTRACT
Substantial efforts have been applied to engineer CA with desired emergent properties, such as supporting gliders. Recent work in continuous CA has generated a wide variety of compelling bioreminscent patterns, and the expansion of CA research into continuously-valued domains, multiple channels, and higher dimensions complicates their study. In this work we devise a strategy for evolving CA and CA patterns in two steps, based on the simple idea that CA are likely to be complex and computationally capable if they support patterns that grow indefinitely as well as patterns that vanish completely, and are difficult to predict the difference in advance. The second part of our strategy evolves patterns by selecting for mobility and conservation of mean cell value. We validate our pattern evolution method by re-discovering gliders in 17 of 17 Lenia CA, and also report 4 new evolved CA and 1 randomly evolved CA that support novel evolved glider patterns. The CA reported here share neighborhood kernels with previously described Lenia CA, but exhibit a wider range of typical dynamics than their Lenia counterparts. Code for evolving continuous CA is made available under an MIT License.1

KEYWORDS
cellular automata, evolution, complexity, artificial life

1 INTRODUCTION
The anthropic principle has many variants in two categories: those predicated on the universe being somehow fine-tuned to support life (famously espoused by Barrow and Tipler [1]), and the incorporation of selection bias into reasoning about the universe, described originally by Carter [4]. We can consider similar perspectives when it comes to examining life-like capabilities in cellular automata (CA).

1https://github.com/rivesunder/yuca

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO ’22 Companion, July 9–13, 2022, Boston, MA, USA  
© 2022 Copyright held by the owner/author(s).  
ACM ISBN 978-1-4503-9268-6/22/07.  
https://doi.org/10.1145/3520304.3529037

2 BACKGROUND
2.1 Growth, Decay, and Complexity
Significant efforts have gone into categorizing or finding specific characteristics of CA systems that underlie complexity and universal computation. These efforts seek to define what makes a complex system “interesting” and to predict whether they are capable of universal computation by developing objective metrics or practical heuristics

Wolfram proposed a classification scheme for 1D CA with subjective criteria intended to capture complexity and universality [29], also applied to 2D CA [19]. Class I CA progress to a homogeneous state, typically cells either all become quiescent or all active with a uniform value. Class II CA settle into static or oscillating pattern equilibria. Class III CA continuously generate chaotic patterns,

1In this work we use the term glider to refer to mobile patterns in general, also called spaceships, as well as the original reflex glider from Life.
We use vanishing and growth as the basis for fitness in our evolving systems like Life and, as implied in the name, the processes of life have much in common with engineered computation. As the authors of Winning Ways put it, with basic logic gates and wires defined:

“From here on it’s just an engineering problem to construct an arbitrarily large finite (and very slow!) computer. Our engineer has been given the tools—let him finish the job!” [3]

Computation is an interesting and useful characteristic of complex systems like Life and, as implied in the name, the processes of life have much in common with engineered computation [17].

Living systems must sense, act on, and copy internal and/or external information in order to be successful. Modern computational resources have facilitated large scale simulation of continuous CA that produce evocative bioreminiscent patterns, at the expense of being somewhat more complicated than their discrete antecedents.

2.3 Continuous Cellular Automata

In addition to Life, CA systems have been developed with larger neighborhoods [12, 21], higher dimensions [2, 6], and many other extensions. While continuous CA have been developed and applied for modeling tasks for several decades [22, 24, 25], recent work has produced a plethora of bioreminiscent and aesthetically pleasing dynamic patterns, particularly in the Lenia framework [5, 6].

 Parsimony suggests we should prioritize minimally-complicated CA systems that still fulfill desired objectives. Life is by some measures the simplest 2D CA in its class [20], and is Turing complete [3, 23], as is the 1D elementary CA 110 [8]. Against the simplicity and complexity of these precedents, do continuous CA offer novel capabilities, or merely appeal to human pareidolia and aesthetic sensibilities? There may indeed be a payoff to the complicated-ness of continuous CA and their successors: self-organizing intelligent agents, fully-embodied in self-consistent simulation.

Recent work nascently demonstrated such self-organizing agents. Authors used gradient descent to train continuous CA update rules, generating robust mobile patterns that survive interactions with immutable obstacles [15]. Simulated environments are typically distinct from agents in reinforcement learning and evolutionary optimization, even when the environment and the agent are both modeled as different types of CA as in [9]. Future work stemming from continuous CA may find systems that exhibit autopoietic selection and robustness (and eventually learning) with no externally imposed evolution or gradient-based optimization.

3 METHODS

3.1 Glaberish Framework

We use a continuous CA framework called Glaberish [10]. Based on Lenia, Glaberish extends Lenia’s update function by splitting the single growth function $G$ from Lenia into genesis and persistence functions $G_{\text{gen}}$ and $P$, respectively, analogous to the Birth and Survival rules in Life-like CA [13]. The Glaberish update is shown in Equation 1.

$$A_{t+dt} = \rho \left( A_t + dt \cdot \left[ (1 - A_t) \cdot G_{\text{gen}}(n) + A_t \cdot P(n) \right] \right)$$

Where $A_t$ is the grid state at time $t$, $n$ is the result of a neighborhood convolution $K * A_t$, $dt$ is the step size, and $\rho$ is a squashing or clipping function that keeps cell values between 0 and 1.

We also consider several CA from the Lenia framework. Lenia differs from Glaberish in that the update is a single growth function, regardless of current cell value. The equation for the Lenia update is shown in Equation 2.

$$A_{t+dt} = \rho \left( A_t + dt \cdot G(n) \right)$$
Figure 2: Selecting for halting unpredictability to search for complex systems capable of supporting life-like patterns.

3.2 Evolving Halting Unpredictability in Continuously-Valued Cellular Automata

Our approach evolves CA update rules (while keeping a static neighborhood) based on the inability of a trio of convolutional networks to accurately predict whether a CA grid state will settle to a quiescent state (all cells have value 0) or remain active after a number of update steps. To achieve this, we wrapped training/validation of convolutional neural networks in a covariance matrix adaptation evolutionary algorithm [16]. The (negative) average accuracy of 3 trained models is the fitness (Equation 3).

\[
\max_\theta E \left( -\frac{1}{N} \sum_n (J(f_{\theta_n}(x), \hat{y})) \right)
\] (3)

Where \(J\) is a function that returns halting prediction accuracy for halting predictions \(f_{\theta_n}(x)\) with respect to the final grid states \(\hat{y}\). A cartoon representation of halting unpredictability evolution is shown in Figure 2.

We also implemented a simple version of CA halting evolution. Simple halting evolution has no inner prediction training loop and fitness is based simply on mean-squared error between the proportion of end-point CA grids with nonzero cell values and a target proportion of 0.5.

3.3 Evolving Glider Patterns Under Cellular Automata Rule Sets

We evolved a population of compositional pattern-producing networks (CPPNs) [27] as synthesis patterns. Fitness is designed to reward motility, homeostasis, and survival, and is composed of a positive reward for displacement in center of mass (motility), penalized for changes in average cell values (homeostasis), and severely penalized for patterns that disappear entirely (survival). Figure 3 shows a graphic representation of pattern evolution.

![Figure 3: Pattern evolution with mobility-based fitness. Solid line is fitness, dashed line is the motility component, and dotted line is the homeostasis component.](image)

Table 1: Example CA supporting CPPN-mediated evolved gliders. Update functions are Gaussians with peaks at \(\mu\) and width \(\sigma\). Lenia CA have a single update function, while CA evolved in this project are based on the Glaberish framework with update functions split into genesis and persistence (\(g\) and \(p\)). * indicates *Hydrogeminium natans* neighborhood kernel, other CA use *Orbium* kernel parameters [5].

| Name          | Origin | \((\mu, \sigma)\) |
|---------------|--------|--------------------|
| *Orbium*      | Lenia  | (0.150, 0.0150)    |
| *P. s. labens*| Lenia  | (0.330, 0.0462)    |
| *S. valvatus* | Lenia  | (0.292, 0.0486)    |
| *D. valvatus* | Lenia  | (0.337, 0.0595)    |
| *H. natans*  | Lenia  | (0.260, 0.0360)    |

s7* Simple evo. (0.0420, 0.00490) \(_g\) (0.261, 0.0292) \(_p\)

s613* Pred. evo. (0.0621, 0.00879) \(_g\) (0.215, 0.0369) \(_p\)

s11* Simple evo. (0.0761, 0.0107) \(_g\) (0.260, 0.0303) \(_p\)

s643* Simple evo. (0.0670, 0.0101) \(_g\) (0.248, 0.0186) \(_p\)

s113* Random evo. (0.266, 0.0382) \(_g\) (0.289, 0.0215) \(_p\)

Table 4 RESULTS & DISCUSSION

We recovered gliders in 17 of 17 select CA available online\(^3\), previously described in the Lenia framework. Table 1 lists 5 examples each of Lenia CA and evolved CA which supported glider evolution.

Table 2 includes metrics based on Epstein’s heuristics [11] and approximate CA classes [19, 29]. Mortality ratio is the proportion of grids with all zero cell values; fertility ratio is the proportion of grids where patterns escaped from a bounding box twice as tall.

\(^3\)https://chakazul.github.io/Lenia/JavaScript/Lenia.html
Evolved CA supporting gliders in Table 1 tend to have more diverse dynamics than their Lenia counterparts, and may help increase the already expansive diversity of biorealistic patterns discovered in Lenia, developed previously via manually and by interactive evolution.

This work was supported by the National Science Foundation under the Emerging Frontiers in Research and Innovation (EFRI) program (EFMA-1830870).

REFERENCES

[1] Barrow, J. and Tipler, F. (1986). The Anthropic Cosmological Principle. Oxford University Press.
[2] Bays, C. (1987). Candidates for the Game of Life in three dimensions. Complex Syst., 1.
[3] Berlekamp, E. R., Conway, J. H., and Guy, R. K. (2004). Winning Ways for Your Mathematical Plays Volume 4. Second Edition. A K Peters, Wellesley, Massachusetts.
[4] Carter, B. D. (1983). The anthropic principle and its implications for biological evolution. Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences, 310:347 – 363.
[5] Chan, B. W.-C. (2019). Lenia - biology of artificial life. Complex Systems, 28(3):251–286.
[6] Chan, B. W.-C. (2020). Lenia and expanded universe. The 2020 Conference on Artificial Life, pages 221–229.
[7] Contributors, V. (2016-2022). List of the Turing-complete totalistic Life-like CA. https://conwaylife.com/forums/viewtopic.php?f=11&t=2597.
[8] Cook, M. (2004). Universality in elementary cellular automata. Complex Systems, page 46.
[9] Davis, Q. T. (2021). Carle’s game: An open-ended challenge in exploratory machine creativity. In IEEE Conference on Games.
[10] Davis, Q. T. and Bongard, J. (2022). Glaabersh: generalizing the continuously-valued Lenia framework to arbitrary Life-like cellular automata. ALIFE 2022: The 2022 Conference on Artificial Life.
[11] Epstein, D. (2010). Growth and decay in Life-like cellular automata. In Adamatzky, A., editor, Game of Life Cellular Automata, pages 71–100. Springer, London.
[12] Evans, K. M. (2001). Larger than Life: Digital creatures in a family of two-dimensional cellular automata. Discrete Mathematics & Theoretical Computer Science, DMTCS Proceedings vol. AA., pages 227–228.
[13] Gardner, M. (1970). Mathematical games - the fantastic combinations of John Conway’s new solitaire game ‘Life’. Scientific American, 225:120–123.
[14] Gardner, M. (1983). Wheels, Life, and Other Mathematical Amusements. W.H. Freeman, New York, New York.
[15] Hamon, G., Etcheverry, M., Chan, B. W.-C., Moulin-Frier, C., and Oudeyer, P.-Y. (2022). Blog post: Learning sensorimotor agency in cellular automata. https://developmentalsystems.org/sensorimotor-lenbia/.
[16] Hansen, N. (2016). The CMA evolution strategy: A tutorial. https://www.rudyrucker.com/capow/.
[17] Hamon, G., Etcheverry, M., Chan, B. W.-C., Moulin-Frier, C., and Oudeyer, P.-Y. (2022). Blog post: Learning sensorimotor agency in cellular automata. https://developmentalsystems.org/sensorimotor-lenbia/.
[18] Hansen, N. (2016). The CMA evolution strategy: A tutorial. ArXiv abs/1604.00772.
[19] Kempes, C. P., Woffert, D., Cohen, Z., and Perez-Mercader, J. (2017). The thermodynamic efficiency of computations made in cells across the range of life. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 375(2109):20160343.
[20] Pivato, M. (2007). RealLife: the continuum limit of Larger than Life cellular automata. Theoretical Computer Science, 372(1):46–68.
[21] Racket, S. (2011). Generalization of Conway’s “Game of Life” to a continuous domain - SmoothLife. ArXiv.
[22] Rendell, P. W. (2011). A universal turing machine in Conway’s Game of Life. 2011 International Conference on High Performance Computing & Simulation, pages 764–772.
[23] Rucker, R. (2003a). CAPOW. https://github.com/rudyrucker/capow.
[24] Rucker, R. (2003b). Continuous-Valued Cellular Automata in Two Dimensions. Oxford University Press.
[25] Schleicher, D. (2013). Interview with John Horton Conway. Notices of the American Mathematical Society, 60:567–576.
[26] Stanley, R. O. (2007). Compositional pattern producing networks: A novel abstraction of development. Genetic Programming and Evolvable Machines, 8:131–162.
[27] Trevorrow, A., Rokicki, T., Hutton, T., Greene, D., Summers, J., Verver, M., Munafò, R., and Rowett, C. (2016). Golly version 2.8.
[28] Wolfram, S. (1983). Universality and complexity in cellular automata. Physica D: Nonlinear Phenomena, 10:1–35.

5 CONCLUSIONS

We demonstrated evolution of complex continuous CA, validating these by evolving gliders under the new rule sets. Selecting for poor halting prediction performance, and simply evolving CA rules that support both halting and persistent patterns, both yield CA that support gliders, the latter having lower computational and tuning overhead.

and wide as the initialized area \(^4\). Putative classes are subjective, but reflect the diversity in the dynamics exhibited by these CA. The Lenia CA typically generate Turing pattern-like grid states, though not entirely static. \(s7\) usually vanishes, \(s613, s643, \) and \(s11\) exhibit chaotic, dynamics, and \(s113\) generates a Turing pattern-type grid, similar to most Lenia CA. Despite different dynamics, all of these CA support mobile, self-organizing patterns.

| Name     | Fertility ratio | Mortality ratio | Putative class |
|----------|----------------|-----------------|----------------|
| Orbium   | 0.745/0.931    | 0.0377/0.0469   | II             |
| D. valvatus* | 0.856/1.0    | 0.0/0.0         | II             |
| H. natans* | 0.923/1.0    | 0.0/0.0208      | II/IV          |
| P. s. labens | 0.785/1.0   | 0.0/0.701       | II/IV          |
| S. valvatus | 0.870/1.0    | 0.0/0.0         | II             |
| \(s7^*\) | 0.0639/0.0158  | 0.577/0.976     | I              |
| \(s613^*\) | 0.953/0.993   | 0.0/0.0         | III/IV         |
| \(s11^*\) | 0.889/0.933   | 0.0212/0.0234   | IV             |
| \(s643^*\) | 0.937/0.999   | 0.0/0.0         | III/IV         |
| \(s113^*\) | 0.693/0.992   | 0.00615/0.00781 | II             |

Table 2: Metrics and putative CA classes. Metrics are split (x/y) into first (x) and second (y) 512 CA steps, approximating initial and steady-state behavior.