The EvCA Project: A Brief History

Wim Hordijk
SmartAnalytiX.com
wim@WorldWideWanderings.net

Abstract

This article presents a brief history of the Evolving Cellular Automata (EvCA) project. In the EvCA project, a genetic algorithm was used to evolve cellular automata to perform certain (non-trivial) computational tasks, in an effort to gain more insight into the question: "*How does evolution produce sophisticated emergent computation in systems composed of simple components limited to local interactions?*" Next to providing many interesting results and useful insights, the EvCA project seems to have spawned a whole research area of its own. Here, a brief overview is given of how it all started, developed, and inspired further work.

1 The Pre-History

1.1 Computation at the edge of chaos

In 1990 a paper appeared with the provocative title *"Computation at the Edge of Chaos"* [16]. This paper, written by Chris Langton (who was later considered to be the founder of the field of Artificial Life), examined cellular automata and their ability to perform computations. *Cellular automata* (CA) are mathematical models of spatial dynamical systems consisting of many simple individual components (laid out in a regular grid) with only local interactions, yet they are capable of producing complex global patterns and even performing sophisticated computations. The paper's main conclusion was that "*we can locate computation within the spectrum of dynamical behaviors at a phase transition here at the 'edge of chaos'*."

This "spectrum of dynamical behaviors" in cellular automata was represented by a single variable, called the λ (lambda) parameter, which is actually calculated from the (static) CA update rule, and not from the resulting dynamical behavior itself. Langton does show in his paper that there seems to be a correlation (at least qualitatively) between a CA's λ value and its actual dynamical behavior, but it is only a weak one, with much variability between individual cases, even for similar λ values. Furthermore, "computational ability" was not very precisely defined either, but represented by a small number of general statistics of a CA's dynamics, which Langton claimed are directly correlated with a capability for computation (but again, only in a weak and general sense).

Despite this fairly weak support for its main conclusion, the paper had a big impact, and the phrase 'edge of chaos' certainly stuck in people's minds. Even more so since similar ideas (and phrases) were published around the same time by others (see, e.g., [5, 15]).

1.2 Adaptation toward the edge of chaos

At around the same time, Norman Packard performed the first experiments on *evolving* cellular automata with a genetic algorithm [22]. A *genetic algorithm* (GA) is a stochastic search method modeled after natural evolution, and can be used to find good (approximate) solutions to difficult optimization problems. The algorithm maintains a population of candidate solutions, which evolves over time by (probabilistic) selection of the best (most "fit") solutions in the current population and (re)combining them in a way similar to genetic crossover and mutation, producing offspring individuals that constitute the population at the next generation. GAs are also often used as simple models of actual evolutionary processes.

Packard's work was inspired by a hand-designed, one-dimensional, two-state CA known as the GKL rule [9, 8], which was invented to be robust against small errors ("noise") in the CA's dynamical update process. However, this CA also turned out to be capable of performing a computational task known as *density classification*, where the CA has to decide whether there are more zeros or more ones in the initial configuration (IC), by eventually settling down to an all-zeros or all-ones configuration, respectively. It was shown that the GKL rule performs this task with a high degree of accuracy for most initial "densities" (i.e., number of ones in the IC). Only for densities close to 0.5 (i.e., roughly an equal number of zeros and ones), does its performance ("fitness") degrade somewhat, to about 70-80% correct classifications.

The GKL rule provides an interesting example of *emergent computation* in a CA, as the density classification task requires global information processing whereas the components in the CA (the individual cells) are limited to local interactions only. Using the density classification task as a concrete example of "useful computation", and a genetic algorithm to evolve CA rules to perform this task, Packard wanted to test two hypotheses: (1) CA rules able to perform complex computations are most likely to be found near "critical" λ_c values (i.e., near the phase transition, or 'edge of chaos'); and (2) when rules are evolved to perform a complex computation, evolution will tend to select rules near such critical λ_c values. Packard's results seemed to support these hypotheses, as his evolved CA rules did indeed cluster around critical λ values, and thus showed adaptation toward the edge of chaos.

2 The History

2.1 Revisiting the edge of chaos

Intrigued by this idea of computation at the edge of chaos and the possibility that evolution drives complex systems toward it under selective pressure, computer scientist Melanie Mitchell (already known for her earlier work on genetic algorithms), aided by graduate student Peter Hraber, set out to replicate Packard's original experiment. Soon, they also enlisted the help of physicist Jim Crutchfield, who had already worked extensively on cellular automata and dynamical systems theory. Thus was born the "*Evolving Cellular Automata* " (EvCA) group, led by Mitchell and Crutchfield, and based at the Santa Fe Institute (SFI) in the high desert of New Mexico, USA.

However, the group was not able to replicate Packard's results. In fact, their results clearly *contradicted* those of Packard, as their evolved CA rules clustered around λ values *away* from the critical ('edge of chaos') λ_c values. After constructing a theoretical argument for why this should be so for the density classification task, and personal discussions with Packard, their only remaining conclusion was that Packard's original results must have been due to mechanisms in his particular GA implementation that were not reported in his paper.

This, together with the mentioned shortcomings in Langton's work, led the EvCA group to be rather critical of Langton's and Packard's earlier work. It did not necessarily *disprove* the 'edge of chaos' hypothesis, but it certainly did not support it, and pointed out that more precise definitions and positive experimental results were required to support the hypothesis. In their own words: "The results presented here do not disprove the hypothesis that computational capability can be correlated with phase transitions in CA rule space. [...] We have shown only that the published experimental support cited for hypotheses relating λ_c and computational capability in CAs was not reproduced." [21, 19]

But this was not the end of the story. In fact, it was only the beginning...

2.2 Emergent computation in evolved cellular automata

Instead of only looking at the λ values of the evolved CA rules, the EvCA group decided to investigate in more detail the actual dynamical behavior of the evolved CAs, and also their evolutionary history. This brought up a whole plethora of additional insights. For example, they identified what they called "*epochs*" in the evolutionary process. During these epochs, the fitness levels in the GA population are relatively stable, until a sudden "jump" to a higher fitness level occurs, followed by another epoch of relative stability, and so on. Furthermore, these jumps clearly correlated with significant changes (caused by GA-induced mutations in the CA update rules) in the dynamical behavior of the CAs, which gave rise to a higher fitness value, i.e., increased ability to solve the density classification task [20].

However, as often in science, these additional insights also brought up many new questions. And so, after recruiting another SFI graduate student, Rajarshi Das, the group decided to perform more experiments and analyses. In these new experiments, it turned out that a small percentage of the GA runs actually produced CA rules very similar to the GKL rule that had originally inspired Packard, with a similar performance on the density classification task (see Figure 1 for an example). Using a mathematical framework called *computational mechanics* as applied to CAs, developed by Crutchfield and Hanson [10, 3], the dynamical behavior of these high-performance evolved CAs was analyzed in terms of "regular domains", "particles", and "particle interactions" [7, 4].

These mathematical formalizations of the emergent patterns in a CA's dynamical behavior can be used to provide a concise description of the CA's global dynamics, abstracting away from the underlying individual cells, and eliciting a CA's "*intrinsic computation*". It was argued that these emergent patterns (in particular the particles and their interactions, as shaped and adapted by the evolutionary process), were the features that really allowed the CA to perform the necessary global information processing to solve the density classification task. Furthermore, additional experiments on a different computational task known as *global synchronization*, resulted in evolved CAs with very similar particle-based "computational strategies" [6] (an example is shown in Figure 2).

Using the computational mechanics framework this way, another newly-recruited graduate student in the group, Wim Hordijk, constructed so-called "*particle models*" of some of the evolved CAs, where their global dynamics is modeled at the emergent level of the particles, abstracting away from the underlying individual cells in the CA. He then showed that these particle models can be used to accurately (and efficiently) predict the actual CA's dynamical behavior and, moreover, its performance on the given computational task [11, 12]. These results thus provided a formal and detailed understanding of emergent computation in evolved cellular automata.

With all this, the EvCA group had come a long way in answering (at least partly) the more general question: "*How does evolution produce sophisticated emergent computation in systems composed of simple components limited to local interactions?*" Typical examples of emergent computation in biological (i.e., evolved, distributed, and decentralized) systems include foraging and nest building strategies in social insects, sensory input processing in the brain, and quorum sensing among bacteria, which are all still highly relevant research topics. Of course the EvCA framework is a (perhaps overly) simplified model of such biological systems, but as often in science, it is useful to start with the simplest model that still captures the essential properties and behaviors of interest. And in that sense, the EvCA framework has certainly been very valuable.

2.3 Extensions of the EvCA framework

Furthermore, in trying to answer this general question, several additional ideas and extensions of the EvCA framework were pursued as well. Yet another graduate student and group member, Erik van Nimwegen, developed a mathematical theory that accurately predicts the fitness levels of the evolutionary epochs observed during the GA evolution of the CAs, and also the speed of evolutionary innovations between them [28, 29]. This work can also be directly related to ideas about neutral evolution.

Work on co-evolution was done with undergraduate intern Justin Werfel and postdoctoral researcher Ludo Pagie. In these experiments, the GA contained two populations: the first one being the population of CA rules, as before, and the second one being a population of initial configurations (ICs) on which the CA rules are tested. The idea is that these two populations co-evolve, the fitness of a CA rule (from the first population) being the fraction of ICs (from the second population) it classifies correctly (using the density classification task), and the fitness of an IC being the fraction of CA rules that misclassify it [30, 23].

And finally, two-dimensional CAs were also evolved for the density classification task [13].

3 The Future

As all good things must come to an end, so did the EvCA project. The Santa Fe Institute does not offer permanent positions, and over time the various group members moved elsewhere to pursue their individual research projects and academic careers. However, in the mean time other researchers had picked up on the idea of using genetic algorithms (and variants thereof) to evolve cellular automata, and produced their own results. Some of these early efforts extended the framework to non-uniform CA rules (where each cell has its own update rule), see e.g. [2, 27]. Others turned it into a competition, trying to find even better CA rules to perform density classification [1], although this was never the intention of the original project, as the EvCA group was mostly interested in trying to answer the more fundamental, scientific questions. Also the idea of using co-evolution was continued by others [14], and the EvCA framework was later even used as a possible constructive bridge in the computation versus dynamics debate in cognitive science [24].

By now it has become impossible to keep track of everything that is published on the EvCA topic. In fact, it seems to have grown into its own research area. More recent work has, for example, introduced automatic filters to detect coherent structures (such as particles and their interactions) in a CA's behavior, and has shown that these are indeed the most complex ("autonomous") structures in its dynamics

[25]. Further work has provided measures to quantify information storage and transfer in distributed computation, leading to direct quantitative evidence in support of the earlier conjectures about the information processing roles of these emergent structures [17]. However, it also appears that various micro-level analyses (i.e., based on measures of the update rule itself, as opposed to emergent structures in the global dynamics) provide useful and additional insights into the overall behavior and computational ability of evolved CAs and similar distributed systems [26, 18]. So, there clearly are still many issues to investigate and questions to be answered. It is indeed a pleasure (and also an honor) to see that the original questions posed, and insights provided, by the EvCA group many years ago are still considered relevant and interesting today.

In conclusion, what started out as a standard scientific exercise in trying to replicate the results of other researchers, turned into a whole new research program of its own. Even though the original EvCA project is not active anymore, its (former) website is still maintained by Jim Crutchfield, and contains a list of all the papers published by the group (including links to downloadable copies): `http://csc.ucdavis.edu/∼evca/`. And with the continuing efforts and progress by others today, hopefully we can truly achieve a full understanding of emergent computation and its evolution in cellular automata (from a scientific point of view), which could then also lead to useful practical applications (from an engineering point of view). May the results of the original EvCA project continue to inspire new efforts and insights!

Acknowledgments

The author would like to thank Melanie Mitchell, Jim Crutchfield, and the Santa Fe Institute for having provided such a stimulating and inspiring working environment all those years, and the other EvCA group members for their invaluable contributions to the project. Thanks also go to two reviewers for providing valuable feedback on more recent work.

References

[1] D. Andre, F. H. Bennett III, and J. R. Koza. Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem. In J. R. Koza, editor, *Proceedings of the First Annual Conference on Genetic Programming*, pages 3–11. MIT Press, 1996.

[2] M. S. Capcarrère, M. Sipper, and M. Tomassini. Two-state, r=1 cellular automaton that classifies density. *Physical Review Letters*, 77(24):4969–4971, 1996.

[3] J. P. Crutchfield and J. E. Hanson. Turbulent pattern bases for cellular automata. *Physica D*, 69:279–301, 1993.

[4] J. P. Crutchfield and M. Mitchell. The evolution of emergent computation. *Proceedings of the National Academy of Sciences*, 92(23):10742–10746, 1995.

[5] J. P. Crutchfield and K. Young. Computation at the onset of chaos. In W. Zurek, editor, *Entropy, Complexity, and Physics of Information*, pages 223–269. Addison-Wesley, 1990.

[6] R. Das, J. P. Crutchfield, M. Mitchell, and J. E. Hanson. Evolving globally synchronized cellular automata. In L. J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 336–343. Morgan Kaufmann, 1995.

[7] R. Das, M. Mitchell, and J. P. Crutchfield. A genetic algorithm discovers particle-based computation in cellular automata. In Y. Davidor, H.-P. Schwefel, and R. Manner, editors, *Parallel Problem Solving from Nature—PPSN III*, pages 344–353. Springer-Verlag, 1994.

[8] P. Gacs. Nonergodic one-dimensional media and reliable computation. *Contemporary Mathematics*, 41:125, 1985.

[9] P. Gacs, G. L. Kurdyumov, and L. A. Levin. One-dimensional uniform arrays that wash out finite islands. *Problemy Peredachi Informatsii*, 14:92–96, 1978.

[10] J. E. Hanson and J. P. Crutchfield. The attractor-basin portrait of a cellular automaton. *Journal of Statistical Physics*, 66(5/6):1415–1462, 1992.

[11] W. Hordijk, J. P. Crutchfield, and M. Mitchell. Embedded particle computation in evolved cellular automata. In T. Toffoli, M. Biafore, and J. Leão, editors, *Proceedings of the Conference on Physics and Computation*, pages 153–158. New England Complex Systems Institute, 1996.

[12] W. Hordijk, J. P. Crutchfield, and M. Mitchell. Mechanisms of emergent computation in cellular automata. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature–V*, pages 613–622. Springer-Verlag, 1998.

[13] F. Jimenez-Morales, J. P. Crutchfield, and M. Mitchell. Evolving two-dimensional cellular automata to perform density classification: A report on work in progress. *Parallel Computing*, 27(5):571–585, 2001.

[14] H. Juillé and J. B. Pollack. Coevolving the "ideal" trainer: Application to the discovery of cellular automata rules. In J. R. Koza, editor, *Proceedings of the Third Annual Conference on Genetic Programming*, 1998.

[15] S. A. Kauffman. *The Origins of Order*. Oxford University Press, 1993.

[16] C. G. Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D*, 42:12–37, 1990.

[17] J. T. Lizier, M. Prokopenko, and A. Y. Zomaya. Information modification and particle collisions in distributed computation. *Chaos*, 20(3):037109, 2010.

[18] M. Marques-Pita, M. Mitchell, and L. M. Rocha. The role of conceptual structure in learning cellular automata to perform collective computation. In *Unconventional Computation: 7th International Conference. Lecture Notes in Computer Science*, volume 5204, pages 146–163. Springer-Verlag, 2008.

[19] M. Mitchell, J. P. Crutchfield, and P. T. Hraber. Dynamics, computation, and the "edge of chaos": A re-examination. In G. A. Cowan, D. Pines, and D. Melzner, editors, *Complexity: Metaphors, Models, and Reality*, pages 497–513. Addison-Wesley, 1994. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings Volume 19.

[20] M. Mitchell, J. P. Crutchfield, and P. T. Hraber. Evolving cellular automata to perform computations: Mechanisms and impediments. *Physica D*, 75:361–391, 1994.

[21] M. Mitchell, P. T. Hraber, and J. P. Crutchfield. Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex Systems*, 7:89–130, 1993.

[22] N. H. Packard. Adaptation toward the edge of chaos. In J. A. S. Kelso, A. J. Mandell, and M. F. Shlesinger, editors, *Dynamic Patterns in Complex Systems*, pages 293–301. World Scientic, 1988.

[23] L. Pagie and M. Mitchell. A comparison of evolutionary and coevolutionary search. *International Journal of Computational Intelligence and Applications*, 2(1):53–69, 2002.

[24] L. M. Rocha and W. Hordijk. Material representations: From the genetic code to the evolution of cellular automata. *Artificial Life*, 11(1–2):189–214, 2005.

[25] C. R. Shalizi, R. Haslinger, J-B. Rouquier, K. L. Klinkner, and C. Moore. Automatic filters for the detection of coherent structure in spatiotemporal systems. *Physical Review E*, 73:036104, 2006.

[26] I. Shmulevich and S. A. Kauffman. Activities and sensitivities in boolean network models. *Phyical Review Letters*, 93:048701, 2004.

[27] M. Sipper. *Evolution of Parallel Cellular Machines: The Cellular Programming Approach*. Springer-Verlag, 1997.

[28] E. van Nimwegen, J. P. Crutchfield, and M. Mitchell. Finite populations induce metastability in evolutionary search. *Physics Letters A*, 229(2):144–150, 1997.

[29] E. van Nimwegen, J. P. Crutchfield, and M. Mitchell. Statistical dynamics of the royal road genetic algorithm. *Theoretical Computer Science*, 229:41–102, 1999.

[30] J. Werfel, M. Mitchell, and J. P. Crutchfield. Resource sharing and coevolution in evolving cellular automata. *IEEE Transactions on Evolutionary Computation*, 4(4):388–393, 2000.

Figures

Figure 1: An example of an evolved cellular automaton (CA) for the density classification task. Space is horizontal (100 cells), and time goes down the page (100 iterations). In the figure on the left the CA eventually settles down to an all-white configuration, while in the figure on the right it settles down to an all-black configuration, depending on the (random) initial configuration. The regular domains in the dynamical behavior of this CA are the all-white, all-black, and checkerboard patterns. The particles are the boundaries between these regular domains.

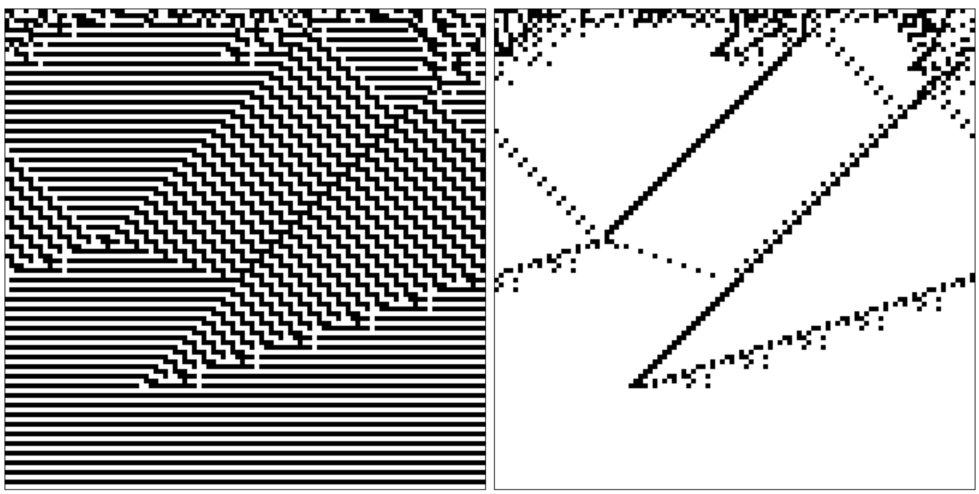

Figure 2: An example (left) of an evolved CA for the global synchronization task, where the CA eventually settles down to a synchronized oscillation between an all-white and an all-black configuration. The figure on the right shows the same space-time diagram, but with the regular domains (the locally synchronous oscillations and the 'zig-zag' pattern) filtered out, clearly showing the particles and their interactions.