Evolvability and Robustness in Artificial Evolving Systems: Three Perturbations [*]

June 2, 2014

Guest Editor: Lee Altenberg, Senior Fellow, The KLI, Klosterneuburg, Austria
gpem@dynamics.org

This special issue focuses on two emergent properties of artificial evolutionary systems: evolvability and robustness. The three papers presented here expand the subject in distinct directions—two dealing with robustness, and one with evolvability.

Short- and long-term evolution depends on the variational properties of the systems: how changes to their structure map to changes in their behavior. Robustness and evolvability are key variational properties that themselves show evolutionary dynamics. This journal's focus is on artificial evolutionary systems that are *active*—which take inputs from their environment and act on them to produce their behavior. Active systems are an especially rich domain for the evolution of robustness and evolvability because the size and structure of their data objects are often open-ended.

Evolvability and robustness were never part of the classical 'Modern Synthesis' which melded Darwin's theory of evolution by natural selection with Mendel's theory of heredity. The reason that evolvability was not an issue was because it was presumed at the outset: empirical experiences with quantitative genetics and selective breeding produced a consensus that phenotypic variation was effectively like a gas which could flow into any selective bottle; e.g. as Francisco Ayala wrote in 1978, "The fact that artificial selection works almost every time it is attempted indicates there is genetic variation in populations for virtually every characteristic of the organism" [3]. It was with the advent of evolutionary computation—the attempt to solve engineering problems by mimicking Darwinian evolution and Mendelian genetics—that it became widely clear that evolvability could not be taken for granted, but was in fact the primary design challenge for evolutionary algorithms. Perhaps it was for this reason that the evolutionary computation literature was more welcoming to early theory on the evolution of evolvability [1] than the evolutionary biology literature.

Robustness was also not a part of the Modern Synthesis, for the reason that there was no empirical experience of robustness as part of the evolutionary process. Robustness as a concept for *physiological* functioning was introduced as early as the 1920s by

[*] In press in *Genetic Programming and Evolvable Machines*

Cannon [5] under the term ‘homeostasis’. But the first ideas about *mutational* robustness appear in the 1940s work of Schmalhausen [15] and Waddington [19].

Early pioneers in the modern theory of the evolution of evolvability and robustness include Riedl [14], and Conrad [6, 7], who described the ‘mutation-absorption model of the enzyme’, and the ‘self-facilitation’ of evolution. Two papers that helped initiate the growth in interest in mutational robustness are Nimwegen *et al.* [12] and Ancel and Fontana [2]. Today evolvability and robustness are active areas of research, with some 170 papers published in the last year that mention the “evolution of evolvability”, while some 250 papers mention “mutational robustness” [Google Scholar]. Masel and Trotter [11] provide a good recent review that documents the ubiquity of mutational robustness in organisms.

The two papers here that address robustness take the subject in novel directions. In their paper, “**Software mutational robustness**” [16], **Eric Schulte and coauthors** set out to quantify the extent of mutational robustness in ‘naturally evolved’ artificial systems, which is to say, software produced for commercial and production use. The idea that real-world software exhibits evolutionary, statistical-mechanical ‘laws’ that emerge independently of, and often contrary to, the intentions of software engineers, was given the name ‘software evolution’ by Lehman [9]. This line of research has been vigorous [10]. ‘Evolution’ in that literature is conceived broadly without the specifically Darwinian character of the field of evolutionary computation. But Schulte *et al.* wonder whether there might be an underlying similarity between these two modes of software creation:

> The tools, design patterns and codes that we have today are those that have proven useful and were robust to software developers edits, hacks and accidents, and those that survived the economic pressures of the marketplace. We hypothesize that these evolutionary pressures have caused software to acquire mutational robustness resembling that of natural systems.

By ‘natural systems’ here, Schulte *et al.* mean *life*.

They apply three kinds of mutation operators to real-world software, and develop a methodology for determining the extent to which these mutations leave program behavior undamaged. The mutation operators they use are deletion, duplication, and reordering. They find remarkably high levels of program robustness under these mutations.

One kind of mutation, however, is not examined: substitution—changing $+$ to $-$, `and` to `or`, bit-flipping in the binary executable, etc.. This would obviously be an important followup study to perform. While real-world software engineering makes plenty of use of code deletion, duplication, and reordering, substitution operations are much more restricted. And so if the source of mutational robustness found by Schulte *et al.* is perhaps a deep similarity between their mutation operators and the operations used in software engineering, we should expect that this robustness is less under substitution mutations. One can hope that their paper will stimulate such further investigations.

Schulte *et al.* go beyond merely documenting the extent of software mutational robustness: they propose how to exploit it. It is interesting to note that the two great phenomena that Darwin sought to explain were the adaptedness of life, and life’s great diversity. Evolutionary computation has focused almost exclusively on how to produce adaptedness—i.e. optimization. That other branch of Darwinism, diversity, is largely unexplored. The importance of software diversity is coming into greater focus now as people realize that real-world computing has become an ecosystem, with arms races between malware, cracking, and security engineering. Schulte *et al.* touch on the ways in which the software mutational robustness they document might be employed to generate software diversity. Their method has already been employed and cited during the time interval between the online publication of their paper and the printed version [4].

Our second paper on robustness returns to the original context in which robustness was conceived: homeostasis, the organism as a robust dynamical system. The paper "**Self-repair ability of evolved self-assembling systems in cellular automata**" [13] by **Can Öztürkeri** and **Colin Johnson** investigates self-repair, a form of robustness in which systems are "able to reconfigure themselves following disruptions to bring them back into a defined normal state." The evolution of the robustness of development is what motivated the early work of Schmalhausen [15] and Waddington [19].

Öztürkeri and Johnson evolve the rules of a 2-D cellular automaton-like system so that it develops from an initial state of all zeros to a target geometric pattern. They then disrupt the pattern in various ways, and find that with remarkably high probability it develops back to the original evolved pattern, even though there was never any direct selection for such self-repair ability. This they call 'self-repair for free', in analogy to Kauffman's 'order for free' [8].

Another way to describe their approach is that they use evolutionary search through the space of cellular automaton rules (which is to say, through their space of dynamical systems) to find systems where

1. their target geometric pattern is the system's *attractor*, and where
2. their initial state of all-zeros is within the *domain of attraction* of the target pattern.

The size of the set of disruptions that repair themselves corresponds to the size of the domain of attraction. They find that the domain of attraction in their system is large, so that with high probability their disruptions repair themselves back to the target pattern.

Their result prompts the following question: is there something about evolutionary search that causes it to find dynamical systems with large domains of attraction? This hope is instantly dashed by another of their interesting findings: self-repair occurs in their system when the grid of automata cells is bounded by four edges; if they eliminate the edges by making the boundaries periodic, self-repair disappears: the domains of attraction are very small. Somehow, the symmetry-breaking caused by the edges gives positional information which is an essential feature for self-repair. One wonders whether this discovery points to phenomena that await a larger theory.

Their paper addresses *developmental* robustness, whereas the more common focus in the literature on evolved dynamical systems is upon *mutational* robustness, for example [17]. Ancel and Fontana [2] make the pivotal discovery that these two may be tied together—that robustness of the attractor of an evolved dynamical system may be intrinsically tied to its mutational robustness, a phenomenon they call 'plastogenetic congruence'. If this phenomenon holds widely, one might expect that the developmental cellular models studied by Öztürkeri and Johnson would show a correspondence between their self-repair capability and their mutational robustness; hence, mutational robustness would be higher in the model with edges, and lower in the model with periodic boundaries. This poses yet another direction for further investigation.

Our third paper is "**On evolvability and robustness in the matrix-GRT model**" [18] by **Uwe Tangen**, and deals primarily with evolvability. Tangen's goal is that 'holy grail' of artificial life research: to produce a system with open-ended evolution or growing complexity in time. To this end he follows the primary heuristic of computational intelligence: emulate nature.

Tangen reviews past efforts to create *in silico* artificial life systems with unbounded evolution and complexity with their different pitfalls. He designs a system based on the separate roles of RNA and proteins—with his virtual RNA to contain the information that builds proteins, and also engage in catalysis, with virtual proteins that carry out further catalytic transformations in the molecule-like system. He incorporates formal digital analogs of our current understanding of cellular biology: replication, transcription, translation, a 'physics' of symbolic protein folding, tRNAs, ligation, restriction enzymes, proteases, methylation, and even chaperones, and embeds these in a spatial grid structure. Needless to say, his system is complicated and complex.

A fundamental question after one has built such a system is methodology: How can you tell when you have open-ended evolution or increase in complexity? —especially when the system is complex to begin with? Tangen employs various visualization techniques and activity measures to give a picture of how his system is evolving. From these he is able to categorize the kinds of behavior into three general patterns: Phase 0, in which the system rapidly optimizes its resource use and ceases to undergo further major organizational changes; Phase I, which begins in a state of turbulent change and eventually settles into the gradual optimizing mode; and Phase II, which can begin with Phase 0 or I behavior, but continues to undergo major organizational changes. Tangen notes that the apparently stable states reached by Phase 0 and I systems may be merely metastable over very long time spans. He describes a fortuitous discovery made when a Phase 0 system was accidentally left running: it appeared to get stuck in a stable state around generation 500, but at generation 65,400 suddenly underwent a radical organizational change.

Due to the computational resources needed for the system, Tangen here provides what could be characterized as an exploration rather than a comprehensive characterization of his system. It therefore leaves ample avenues for further investigation. While Tangen sought to minimize the set of operators emulating the biology of the cell, one wonders whether these could be further whittled down to a 'minimal cell' that still exhibits his Phase 0, I, and II, and long-term metastability.

And then, there is the question of a 'theory' for what is happening in Tangen's system: is there a mathematical way of capturing the system—a description simpler than the system itself—from which its behaviors can be derived? Or is the system itself its own shortest description of its behavior?

I will leave it the reader to answer for themselves—hopefully with papers prompted by Tangen's work—how close he has come to reaching the 'holy grail' of artificial life research: open-ended evolution and increase in complexity.

Acknowledgements

I thank all those who submitted papers and the anonymous reviewers who gave of their time and thoughts which led to helpful revisions. I thank the journal Editor-in-Chief, Lee Spector, for proposing this special issue and invited me to edit it. Thanks to Stephanie Forrest and Eric Schulte for a reference suggestion. I gratefully acknowledge the support of The KLI, and of the Mathematical Biosciences Institute through National Science Foundation Award #DMS 0931642, and the generous hospitality of Daniel and Erika Laubacher.

References

[1] Lee Altenberg. The evolution of evolvability in genetic programming. In Kenneth E. Kinnear, editor, *Advances in Genetic Programming*, chapter 3, pages 47–74. MIT Press, Cambridge, MA, 1994.

[2] L. W. Ancel and W. Fontana. Plasticity, evolvability and modularity in RNA. *Journal of Experimental Zoology (Molecular and Developmental Evolution)*, 288:242–283, 2000.

[3] Francisco J. Ayala. The mechanisms of evolution. *Scientific American*, 239(3):56–69, 1978.

[4] Benoit Baudry, Simon Allier, and Martin Monperrus. Tailored source code transformations to synthesize computationally diverse program variants. *arXiv preprint arXiv:1401.7635*, 2014.

[5] Walter B Cannon. Organization for physiological homeostasis. *Physiological Reviews*, 9(3):399–431, 1929.

[6] Michael Conrad. Evolutionary adaptability of biological macromolecules. *Journal of molecular evolution*, 10(1):87–91, 1977.

[7] Michael Conrad. Mutation-absorption model of the enzyme. *Bulletin of Mathematical Biology*, 41:387–405, 1979.

[8] Stuart A. Kauffman. The sciences of complexity and “origins of order”. *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, Volume Two: Symposia and Invited Papers:299–322, 1990.

[9] Meir M Lehman. The programming process. Technical Report RC2722, IBM Research Centre, Yorktown Heights, NY, September 1969.

[10] Meir M Lehman and Juan C Fernández-Ramil. Background and approach to development of a theory of software evolution. In Nazim H. Madhavji, Juan C. Fernández-Ramil, and Dewayne E. Perry, editors, *Software Evolution and Feedback: Theory and Practice*, pages 339–357. John Wiley and Sons, Ltd, 2006.

[11] Joanna Masel and Meredith V Trotter. Robustness and evolvability. *Trends in Genetics*, 26(9):406–414, 2010.

[12] Erik van Nimwegen, James P. Crutchfield, and Martijn Huynen. Neutral evolution of mutational robustness. *Proceedings of the National Academy of Sciences U.S.A.*, 96:9716–9720, 1999.

[13] Can Öztürkeri and Colin G Johnson. Self-repair ability of evolved self-assembling systems in cellular automata. *Genetic Programming and Evolvable Machines*, 2014.

[14] Rupert J. Riedl. *Die Ordnung des Lebendigen: Systembedingungen der Evolution*. Parey, Hamburg and Berlin, 1975.

[15] I. I. Schmalhausen. *Factors of Evolution: The Theory of Stabilizing Selection*. University of Chicago Press, Chicago, 1949. 1986 edition.

[16] Eric Schulte, Zachary P Fry, Ethan Fast, Westley Weimer, and Stephanie Forrest. Software mutational robustness. *Genetic Programming and Evolvable Machines*, 2013.

[17] Mark L. Siegal and Aviv Bergman. Waddington's canalization revisited: Developmental stability and evolution. *PNAS*, 99(16):10528–10532, 2002.

[18] Uwe Tangen. On evolvability and robustness in the matrix-GRT model. *Genetic Programming and Evolvable Machines*, 2014.

[19] Conrad H. Waddington. Canalization of development and the inheritance of acquired characters. *Nature*, 150:563–565, 1942.