Genet Program Evolvable Mach (2017) 18:363–367
DOI 10.1007/s10710-017-9290-3

COMMENTARY

Probing the axioms of evolutionary algorithm design: Commentary on "On the mapping of genotype to phenotype in evolutionary algorithms" by Peter A. Whigham, Grant Dick, and James Maclaurin

Lee Altenberg[1,2]

Published online: 24 February 2017

Abstract Properties such as continuity, locality, and modularity may seem necessary when designing representations and variation operators for evolutionary algorithms, but a closer look at what happens when evolutionary algorithms perform well reveals counterexamples to such schemes. Moreover, these variational properties can themselves evolve in sufficiently complex open-ended systems. These properties of evolutionary algorithms remain very much open questions.

Keywords Fisher's geometric model · 1/5 rule · Evolution of evolvability

The astonishing and unmatched capabilities of organisms—and in particular their information processing capabilities—inspire us to emulate the ways they achieve these capabilities, an effort called "bio-inspired computing". But there is a hazard in blindly copying what we think nature is doing when we design representations and operators for evolutionary algorithms. This is the caution that Whigham et al. [24] ask us to heed. Indeed, unless we are simulating the physics and chemistry of actual living things (the object of one thread of computational biology), we will be abstracting what we think is the relevant essence of a particular biological phenomenon, and here we are vulnerable to imposing an artifactual framework on the biological reality that may miss the mark. Whigham et al. [24] propose one particular example of this, *grammatical evolution*, inspired by the translation step of the genetic code, in which a system that generates programs via a grammar is pre-

This comment refers to the article available at doi:10.1007/s10710-017-9288-x.

✉ Lee Altenberg
altenber@hawaii.edu

1 Information and Computer Sciences, University of Hawai'i at Mānoa, Honolulu, HI, USA

2 Konrad Lorenz Institute for Evolution and Cognition Research, Klosterneuburg, Austria

appended with a another system that maps arbitrary strings to grammars. The problem with the pre-appended string-to-grammar map, Whigham et al. point out, is that small changes in the strings map to haphazard changes in the programs. Whigham et al. seek out the “axioms” for designing representations that would guide us away from such unproductive constructions, and draw upon nine conditions (C1–C9) described by Sterelny [20]: “inheritance mediated by a replication system satisfying C1–C9 would be evolutionarily potent; it would be highly evolvable.” The conditions relevant to genetic programming are C6, C8, and C9—they address the variational properties [1] of the genotype–phenotype map:

> “**C6** The replicator/organization map should be robust.
> **C8** A small change in the replicator set should generate a small change in biological organization.
> **C9** The generation of biological organization from the replicator set should be modular ...designed so that they make a distinctive contribution to the generation of one or a few traits, and relatively little distinctive contribution to others.”

Whigham et al. [24] use these conditions to compare Grammatical Evolution (GE), with another representation for executable structures, Context-Free Grammar-Guided GP (CFG-GP). They note that an extra layer of representation in GE, a binary or integer encoding of grammar rules, only serves to scramble the neighborhood relations between grammars that are similar, whereas CFG-GP changes the neighborhood relations in a more orderly fashion.

In the context of evolutionary computations, all of Sterelny’s conditions are “sort of” right. But they are, first of all, not stated with the mathematical precision needed to evaluate them, and secondly, they are several steps removed from the actual criteria by which evolutionary algorithms are evaluated.

In evolutionary algorithms, evolvability is *everything*. By evolvability, I mean how quickly the sampling of the search space finds desirable points (how ever we may wish to define ‘desirable’, e.g. the global optimum, an approximation of the global optimum, an approximation of the global optimum objective function, Pareto-optimal, etc.). In living organisms, however, “survivability is more fundamental than evolvability” [16], in that species survival is the precondition for everything else. In evolutionary algorithms, populations do not actually go extinct—selection is merely the choice of which prior samples to apply the variation operators to.

In evolutionary algorithms, it is not the average fitness of new samples is that matters; rather, it is how quickly the iteration of sampling finds the objective. Let us consider what actually happens when an evolutionary algorithm works well. A sequence of samples of the search space is generated by applying variation operators to prior samples such that a path is produced that leads to desired points in the search space. Typically, the selection operator focuses the application of the variation operator on points with the best objective function values among those previously sampled, which means that what is needed are *short paths* of improving objective function values to the desired points in the space.

Here, the "one fifth rule" from evolution strategies is instructive. Rechenberg [17] examined evolutionary search on a landscape much like Fisher's geometric model [11], where a real-valued function is defined on points in $\mathbb{R}^n$, and variation consists of adding a random perturbation to sampled points. Fisher showed that in the limit of small mutation perturbation, the probability of a mutation being advantageous approaches 1/2. But Rechenberg [17] found that the fastest progress to the optimal value of the objective function was attained when the mutation size was increased until only about 1/5 of the mutations were advantageous, and about 80% were deleterious. Orr [15] obtained a similar value of 1/5.6 for the success rate that results from the optimal mutation step size, in a model that included random genetic drift. The reason that adaptation progresses faster with larger steps, even though large steps are less likely to be advantageous, is because the mutations that *were* advantageous took much larger steps toward the optimum.

So we see that minimizing the effect of single mutations on the phenotype may not produce optimal evolvability; what matters is not the average fitness of variants, but the weight in the upper tail of the distribution of fitness effects of the variation operators [1, 3]. Empirical studies of mutations in actual organisms show a typically bimodal distribution of fitness effects of mutation, with one mode at near-lethal mutations, and the other mode at neutral mutations. The large probability that a mutation will be disruptive of the functions of the organism shows that genotype–phenotype map is not smooth all around, but contains many "cliffs". Yet these cliffs, which produce genetic load, are largely irrelevant to evolvability, because with some small but non-vanishing probability, adaptive mutations occur. It is the lack of adaptive mutations, not the presence of lethal mutations, that stymies evolution.

Fisher's geometric model also yields some insights on Sterelny's condition C9, modularity. We do not see improved evolvability in Fisher's geometric model from mutations that affect only a small number of traits. As long as the perturbations of the phenotype are small, there is a 50% chance of a mutation being advantageous regardless of how many traits it affects. The probability of selective advantage is not improved if "they make a distinctive contribution to the generation of one or a few traits." As the dimensionality grows, the probability of a selective advantage drops, but it does not depend on the number of traits affected, merely the total Euclidean distance to the optimal phenotype.

As Orr [15] showed, when the dimensionality of the space of traits is very high, the pace of adaptation will slow to a crawl, because the size of perturbations needed to produce a selective advantage with any likelihood makes the selective advantage so small as to be useless.

In fact, the only way to rescue the adaptive process in very high dimensions is to *focus the variation* in those directions where there is still adaptive opportunity. And these "directions of opportunity" may require changes in many organismal traits. Modularity in and of itself, therefore, does not necessarily improve evolvability—it has to be the right kind of modularity [4]. The right kind of modularity is one that suppresses variation in useless directions, and focuses it in directions of adaptive opportunity. The wrong kind of pleiotropy is variation that spills over into these useless directions.

And this brings us back to the genotype–phenotype map. Whigham et al. [24] are certainly correct when they say that the representation and variation operators acting on it ought to be "aware" of how their interaction searches the solution space. But what kind of awareness will effectively focus the search in adaptive directions?

If we look to nature for the answer to this question, we need to keep in mind that the way genes, physiology, and morphology of organisms represent their life functions may not be so much an optimal system for evolving functions, as the other way around: the functions that organisms have evolved may simply be those that were easiest to discover with the mechanisms that they had (an idea to be found in [21] and [10]).

We must also keep in mind, when thinking about evolutionary algorithms, that evolvability and robustness may not be simple entailments of the designer's choice of representation and variation operators, but may change emergently as a consequence of the evolutionary process itself. Numerous mechanisms have now been proposed and analyzed in which evolvability or robustness can evolve [1, 2, 5–9, 12, 14, 17, 18, 22, 23]. One may conjecture that in any sufficiently complex, open-ended evolutionary computation system, evolvability and robustness are not static properties but evolve as a consequence of the evolutionary dynamics. One of the earliest proposals to explain bloat in genetic programming was that bloat was a "defense against crossover"—a means of producing phenotypic robustness in the face of high recombination rates [19]. Other non-adaptive processes are also involved in bloat [13]. In this phenomenon, the evolutionary system has its own agenda that may depart from the designer's hope for high evolvability.

To conclude, Whigham et al. [24] bring our attention to essential questions about the design of evolutionary algorithms, and their proposals for the properties of representations and variation operators are reasonable—but we see that evolutionary dynamics bring in complications that preclude any simple formulae in terms of "locality", "smoothness", "modularity", and the like. And sufficiently complex systems offer the possibility that their variational properties evolve under their own dynamics, independently of the intentions of the designer. For all these reasons, we can see that the field has a rich set of open questions that invite further investigation.

Acknowledgements This work was supported by the Konrad Lorenz Institute for Evolution and Cognition Research, the Mathematical Biosciences Institute through National Science Foundation Award #DMS 0931642, the University of Hawai'i at Mānoa, and the Stanford Center for Computational, Evolutionary and Human Genomics, Stanford University. I thank Marcus W. Feldman for his hospitality during a visit to his group.

References

1. L. Altenberg, The evolution of evolvability in genetic programming, in *Advances in Genetic Programming*, chapter 3, ed. by K.E. Kinnear (MIT Press, Cambridge, MA, 1994), pp. 47–74
2. L. Altenberg, Genome growth and the evolution of the genotype–phenotype map, in *Evolution and Biocomputation: Computational Models of Evolution*, ed. by W. Banzhaf, F.H. Eeckman. Lecture Notes in Computer Science, vol. 899 (Springer, Berlin, 1995), pp. 205–259
3. L. Altenberg, The schema theorem and Price's theorem, in *Foundations of Genetic Algorithms 3*, ed. by D. Whitley, M.D. Vose (Morgan Kaufmann, San Mateo, 1995), pp. 23–49

4. L. Altenberg, Modularity in evolution: some low-level questions, in *Modularity: Understanding the Development and Evolution of Natural Complex Systems*, ed. by W. Callebaut, D. Rasskin-Gutman (MIT Press, Cambridge, 2005), pp. 99–128
5. E. Bornberg-Bauer, H.S. Chan, Modeling evolutionary landscapes: mutational stability, topology, and superfunnels in sequence space. Proc. Natl. Acad. Sci. **96**(19), 10689–10694 (1999)
6. M. Conrad, Molecular information processing in the central nervous system, in *Physics and Mathematics of the Nervous System*, ed. by M. Conrad, W. Güttinger, M. Dal Cin (Springer, Berlin, 1974), pp. 82–107
7. M. Conrad, The geometry of evolution. BioSystems **24**(1), 61–81 (1990)
8. A. Crombach, P. Hogeweg, Chromosome rearrangements and the evolution of genome structuring and adaptability. Mol. Biol. Evol. **24**, 1130–1139 (2007)
9. J. Draghi, G.P. Wagner, Evolution of evolvability in a developmental model. Evolution **62**(2), 301–315 (2008)
10. D.S. Fisher, A few comments (& questions!) from a condensed matter physicist, 2014. Talk presented March 21, 2014 at The Simons Institute, Berkeley, Computational Theories of Evolution, https://simons.berkeley.edu/talks/wrap-up-session-general-discussion
11. R.A. Fisher, *The Genetical Theory of Natural Selection* (Clarendon Press, Oxford, 1930)
12. N. Kashtan, U. Alon, Spontaneous evolution of modularity and network motifs. Proc. Natl. Acad. Sci. USA **102**, 13773–13778 (2005)
13. W.B. Langdon, R. Poli, Fitness causes bloat, in *2nd On-line World Conference on Soft Computing in Engineering Design and Manufacturing (WSC2)* (1997), pp. 1–10
14. L.A. Meyers, M. Lachmann, Evolution of genetic potential. PLoS Comput. Biol. **1**, 236–243 (2005)
15. H.A. Orr, Adaptation and the cost of complexity. Evolution **54**(1), 13–20 (2000)
16. M.E. Palmer, M.W. Feldman, Survivability is more fundamental than evolvability. PloS ONE **7**(6), e38025 (2012)
17. I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution* (Frommann-Holzboog, Stuttgart, 1973)
18. R.J. Riedl, A systems-analytical approach to macroevolutionary phenomena. Q. Rev. Biol. **52**, 351–370 (1977)
19. A. Singleton, N. Keenan, Defense against crossover, in *Discussion in the Genetic Programming Workshop at the Fifth International Conference on Genetic Algorithms* (1993)
20. K. Sterelny, Niche construction, developmental systems, and the extended replicator, in *Cycles of Contingency: Developmental Systems and Evolution*, ed. by S. Oyama, P.E. Griffiths, R.D. Gray (MIT Press Cambridge, MA, 2001), pp. 333–350
21. L. Valiant, *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World* (Basic Books, New York, 2013)
22. E. van Nimwegen, J.P. Crutchfield, M. Huynen, Neutral evolution of mutational robustness. Proc. Natl. Acad. Sci. USA **96**, 9716–9720 (1999)
23. G.P. Wagner, L. Altenberg, Complex adaptations and the evolution of evolvability. Evolution **50**(3), 967–976 (1996)
24. P.A. Whigham, G. Dick, J. Maclaurin, On the mapping of genotype to phenotype in evolutionary algorithms. *Genet. Program. Evolvable Mach.* (2016)