

Distilling the salient features of natural systems: Commentary on “On the mapping of genotype to phenotype in evolutionary algorithms” by Whigham, Dick and Maclaurin

Michael O’Neill¹ · Miguel Nicolau¹

Published online: 28 February 2017
© Springer Science+Business Media New York 2017

Abstract Here we comment on the article, “On the mapping of genotype to phenotype in evolutionary algorithms”, by Peter A. Whigham, Grant Dick, and James Maclaurin. The authors present a critical view on the use of genotype to phenotype mapping in Evolutionary Algorithms, and how the use of this analogy can be detrimental for problem solving. They examine a grammar-based approach to Genetic Programming (GP), Grammatical Evolution (GE), and highlight properties of GE which are detrimental to effective evolutionary search. Rather than use loose analogies and metaphors, we suggest that a focus should be (and has been in GE and other approaches to GP) on addressing one of the most significant open issues in our field, i.e., What are the sufficient set of features in natural, genetic, evolutionary and developmental systems, which can translate into the most effective computational approaches for program synthesis?

1 Analysis

Although the article title suggests a critical view of the use of molecular biology as inspiration for designing evolutionary algorithms, the first four pages seem to draw a much broader criticism, of the actual use of evolution to solve problems. On their initial view of the relationship between genetic operators and the mapping process employed, the authors support the idea that “genetic operators and the analogues of schemata should be defined in the space of phenotypes”, as a view against the use of GPM. However, to this day, there is still no definite theory nor proof that the

This comment refers to the article available at doi:[10.1007/s10710-017-9288-x](https://doi.org/10.1007/s10710-017-9288-x).

✉ Michael O’Neill
m.oneill@ucd.ie

¹ School of Business, University College Dublin, Dublin, Ireland

propagation of schemata exists in GP-like systems, even when genetic operators are applied directly to phenotypic structures, so their criticism seems to encompass most GP-like systems.

The departure from GPM criticism and the further broadening to the whole field of Evolutionary Computation (EC) is further exacerbated with the criticism of the “assumption that biological (fitness) landscapes are closely analogous to problems encountered in the application of GP”. To support this idea, a list highlighting the sub-optimality of biological evolution is produced, suggesting that the EC field as a whole is based on a suboptimal process. This list, however, is mostly disconnected from the reality of common use of EC algorithms:

- point 1 is only a potential criticism within the context of artificial evolution in dynamic environments;
- points 2 and 4 are dealt with in artificial evolution with higher rates of application of variation operators, and possibly lower selection pressure;
- point 3 is only a potential criticism within the context of multi-objective artificial evolution;
- point 6 is only a potential criticism within the context of co-evolutionary systems.

Core to the critical views presented throughout the paper is the assumption that biological evolution (not just GPM) is used to guide the development of search algorithms on the basis of it being an optimal or at least best available search mechanism. This seems disconnected with the reality of the EC field; in 2016, it is common knowledge that stochastic search algorithms such as evolutionary algorithms possess both advantages and disadvantages. Not only is the search process not optimal, but also the solutions provided are not necessarily optimal.

The 1990s view that EC methods are useful for optimality search has been dropped from most EC literature. Measures such as Cumulative Frequency of Success are not accepted as success measures anymore, and the variability (and hence sub-optimality) of the evolutionary search process is acknowledged in the requirement for several runs, and appropriate statistical analysis of those. This is not a recent view; seminal papers highlighting these issues appear as far back as the early 2000s (see for example the work of Luke and Panait [8] or Keizjer [6]).

The authors introduce the analysis of Sterelny [21] and using the Grammatical Evolution (GE) system as an example of how using biological evolution (and the GPM process in particular) as a basis for search algorithm design is not a recommended way to proceed. The criticism of GE is based mostly on the original instantiations of GE development [16], which again seems somewhat distanced from current literature: improvements made to GE since (which are acknowledged by the authors) have addressed most of GE’s original shortcomings, much in the same way that vast improvements have been made to Koza’s original GP system [7].

An example of the problems of the GPM process in GE is given, in which “a single point mutation may change all of the mappings that occur after this mutation”. The authors however failed to indicate the probability of this event in

GE's linear representation, with the typical mutation rates used in the literature. Also note that the same effect can occur in the derivation tree structures employed by systems such as CFG-GP, and in fact a very similar effect can occur in GP systems not employing a GPM process: the mutation of a node close to the root of the tree.

The authors then refer to their 2015 publication [22], in which they claim that the CFG-GP system is more stable and robust than GE for the problems tested, as further evidence of the shortcomings of the GE design. Yet the original GE design is used in that study, without any of the improvements presented in the literature since (as an example, GE is randomly initialised, whereas CFG-GP uses a ramped-half-and-half initialisation).

Interestingly, Sterelny's condition 4 points to the requirement that a genotype-phenotype map exists. This is not present in representations such as CFG-GP.

Towards the end of the article a number of principles that should be followed when designing the representation for a particular problem are provided. Some of these are criticisms of the whole GP field, whereas others are in fact addressed to some extent in the GE literature:

1. The idea of building-block preservation is still largely absent from the GP literature to this day, particularly in symbolic regression applications. Yet we refer to two examples of it being used in GE literature, in the evolution of behaviour trees for platform games [19] and the evolution of parameterisation sequences for a Cayley graph visualiser [14]; in both cases, grammar-controlled crossover locations allow for the definition of effective behaviour ([19] or parameter [14]) building blocks. Their definition and propagation is only possible through the GPM employed by GE.
2. This point emphasizes the idea that genetic operators must be designed taking into account the encoding used. In grammar-based systems, however, the encoding is not merely the structure used to represent an individual (linear numerical genomes in GE), but also (and very importantly) the grammar used. Grammars can (and should) be designed taking into account the encoding and operators used, as shown in the literature [3, 15]. By having a grammar-based mapping process, there is more flexibility to adapt to different problems, not only in terms of syntactic specification, but also in deliberate representation adaptation by the end-user.
3. The idea that the representation is stable reverts back to the hierarchical representations used, not only in systems such as GE, but indeed most GP-like algorithms. There is some effort made in the GP community to address this issue, such as the work on Semantic GP [9], and its application to GE [10] (conveniently done through grammar design).
4. The modularity of functions does not necessarily suffer from the use of a GPM process, and in fact, can be controlled through grammar design [13].
5. This point stresses the need for careful considerations when designing a mapping. Again in the context of GE, since its early years the influence of grammar design on the mapping process has been stressed [12].

2 Conclusions

We agree that analogies with molecular biology and other natural systems have been adopted in the broader literature on natural computing, and in our opinion, their use is sometimes tenuous and at worst undermine serious research efforts. A recent keynote by Sörensen on Metaphors in Metaheuristics at EvoStar 2016 and a related article reinforces this perspective [20]. Rather than use loose analogies, in GP we think there is actually a greater emphasis on asking what are the necessary features that enable, for example, biological evolution to operate effectively, and can we attempt to distill the salient features of natural systems (e.g., evolution, genetic, biochemical and developmental) which might be useful for program synthesis, and test their impact in GP. We firmly believe there is still a great deal that we can learn from the natural sciences, and this has been in part the motivation behind some of our research (e.g., [1, 2, 11, 17]), and more generally identifying appropriate representations for GP is considered a significant open issue in the community [18].

On distillation of the salient features of nature, the authors refer to the analysis of Sterelny [21]. While interesting, this is but one perspective, and the authors do not consider the many other significant contributions towards uncovering the general principles of computation in nature (e.g., Kauffman on the pivotal role of self-organisation [5], Hofstadter on strange loops and tangled hierarchies [4], and Wagner on Robustness and Evolvability [23]) and subsequently how they might be leveraged in GP.

We disagree with the authors statement that their analysis “*should not be considered a catalyst for future work on improved variants of GE*”. On the contrary this is exactly what analysis like this should lead to, i.e., improved forms of GP. And indeed this may, or may not, necessitate the involvement of GE, or even additional learnings from the natural world, rather whatever representation (i.e., encoding and search operators) is most effective for program synthesis.

We agree that the design of evolutionary algorithms should not be based on the optimality of biological systems, and there is evidence in the literature that this view is widespread. Evolution is not an optimal process; it is, however, a successful process. Likewise, GE is not an optimal system; it is, however, a successful system. Systems such as GE (and many other EC systems) do not intend on copying biological processes based on their optimality; instead, their design is influenced by the success of such processes, adapting (and vastly simplifying) concepts with the view of solving real-world problems.

In conclusion, as a community we should throw out analogies and metaphors (except in the case of attempting to articulate a concept in simple, understandable terms) and instead focus on asking one of the most significant open issues in our field [18]: What are the sufficient set of features of natural, genetic, evolutionary and developmental systems, which can translate into the most effective computational approaches for program synthesis?

References

1. J. Byrne, M. Nicolau, A. Brabazon, M. O'Neill, An examination of synchronisation in artificial gene regulatory networks. in *Proceedings of IEEE CEC 2014* (IEEE Press, 2014), pp. 2764–2769
2. D. Fagan, E. Hemberg, S. McGarraghy, M. O'Neill, Understanding expansion order and phenotypic connectivity in π GE. in *Proceedings of EuroGP 2013* (Springer, 2013), pp. 37–48
3. R. Harper, GE, Explosive grammars and the lasting legacy of bad initialisation. in *IEEE Congress on Evolutionary Computation—CEC 2010* (IEEE press, 2011) pp. 2602–2609
4. D.R. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid* (Penguin Books, London, 1979)
5. S.A. Kauffman, *The Origins of Order—Self-Organization and Selection in Evolution* (Oxford University Press, Oxford, 1993)
6. M. Keijzer, Improving symbolic regression with interval arithmetic and linear scaling, in *European Conference on Genetic Programming—EuroGP 2003*, ed. by Ryan, et al. (Springer, 2003), pp. 70–82
7. J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Evolution* (MIT Press, Cambridge, 1992)
8. S. Luke, L. Panait, Is the perfect the enemy of the good, in *Genetic and Evolutionary Computation—GECCO 2002*, ed. by Langdon, et al. (Morgan Kaufmann, 2002), pp. 820–828
9. A. Moraglio, K. Krawiec, C.G. Johnson, Geometric semantic genetic programming. in *Parallel Problem Solving from Nature—PPSN XII* (Springer, 2012), pp. 21–31
10. A. Moraglio, J. McDermott, M. O'Neill, Geometric semantic grammatical evolution. in *Proceedings of PPSN 2014 Workshops* (2014)
11. E. Murphy, M. Nicolau, E. Hemberg, M. O'Neill, A. Brabazon, Differential gene expression with tree-adjunct grammars. in *PPSN XII 2012, part 1* (Springer 2012), pp. 377–386.
12. M. Nicolau, Automatic grammar complexity reduction in grammatical evolution. in *Genetic and Evolutionary Computation Workshops—GECCO 2004*, ed by Poli et al. (2004)
13. M. Nicolau, I. Dempsey, Introducing grammar based extensions for grammatical evolution. in *IEEE Congress on Evolutionary Computation—CEC 2006* (IEEE press, 2006), pp. 2663–2670
14. M. Nicolau, D. Costelloe, Using grammatical evolution to parameterise interactive 3D image generation, in *Applications of Evolutionary Computation—EvoApplications 2011*, ed. by Di Chio, et al. (Springer, Berlin, 2011), pp. 374–383
15. M. Nicolau, M. O'Neill, A. Brabazon, Termination in grammatical evolution: grammar design, wrapping, and tails. in *IEEE Congress on Evolutionary Computation—CEC 2012* (IEEE press, 2012)
16. M. O'Neill, C. Ryan, *Grammatical Evolution—Evolutionary Automatic Programming in an Arbitrary Language* (Kluwer, Dordrecht, 2003)
17. M. O'Neill, A. Brabazon, M. Nicolau, S. McGarraghy, P. Keenan, π Grammatical evolution. in *Proceedings of GECCO 2004*, Vol. II (Springer, 2004), pp. 617–629
18. M. O'Neill, L. Vanneschi, S. Gustafson, W. Banzhaf, Open issues in genetic programming. *Genet. Program Evol Mach* **11**(3/4), 339–363 (2010)
19. D. Perez, M. Nicolau, M. O'Neill, A. Brabazon, Reactiveness and navigation in computer games: different needs, different approaches. in *IEEE Conference on Computation Intelligence and Games-CIG 2011* (IEEE press, 2011)
20. K. Sörensen, Metaheuristics—the metaphor exposed. *Int. Trans. Op. Res.* **22**(1), 3–18 (2013)
21. K. Sterelny, Niche construction, developmental systems and the extended replicator, in *Cycles of Contingency: Developmental Systems and Evolution*, ed. by S. Oyama, et al. (MIT Press, Cambridge, 2001)
22. P.A. Whigham, G. Dick, J. Maclaurin, C.A. Owen, Examining the “best of both worlds” of grammatical evolution, in *Genetic and Evolutionary Computation—GECCO 2015*, ed. by Sara Silva (ACM, 2015), pp. 1111–1118
23. A. Wagner, *Robustness and Evolvability in Living Systems* (Princeton University Press, Princeton, 2005)