Association for
Computing Machinery
Advancing Computing as a Science & Profession

Ubiquity, an ACM publication
December, 2013

Ubiquity Symposium

Evolutionary Computation and the Processes of Life

Perspectives and Reality of Evolutionary Computation

Closing Statement

by Mark Burgin and Eugene Eberbach

Editor's Introduction

In this final piece of the evolutionary computing symposium, the editors provide an overview on what they learned from the invited authors, whose contributions explored new ideas for evolutionary computation and its application, the current state of the evolutionary computation field, and trends in evolutionary computation. The editors close out the symposium with a discussion on what they think will be the future of evolutionary computing.

Peter Denning,
Editor-in-Chief

Ubiquity, an ACM publication
December, 2013

Ubiquity Symposium

Evolutionary Computation and the Processes of Life

Perspectives and Reality of Evolutionary Computation

Closing Statement

by Mark Burgin and Eugene Eberbach

In the Opening Statement to this symposium, we discussed the history of evolutionary computation (EC). Our participants were from industry, academia, and research, representing six countries from four continents. Now we can offer some future perspectives based on what the participants of this symposium have said. The area of evolutionary computation has three main parts:

1. **Study of biological systems and exploring natural life.** (Several participants examined the impact of computing on biology.)

2. **Imitation of natural life, creating and exploring artificial life.** (Several participants examined the impact of biology on computing.)

3. **Solution of optimization and other practical problems.** (The remaining participants examined the role of evolutionary computation in information processing technology, artificial intelligence and computer science.)

Our participants' explorations are of three kinds:

1. **Creative.** They suggested new ideas for evolutionary computation and its application. The papers of Yao, Sekanina, Sipper, Riofrio, Roglic, Zenil and Marshall are in this category.

2. **Analytical.** They evaluated the current situation in evolutionary computation. The papers of Michalewicz, Fogel, Schwefel, Wolpert, Mo, and Garzon are in this category.

3. **Predictive.** They examined trends in evolutionary computation. The papers of Sekanina, Yao, Michalewicz, Wolpert, Sipper, and Roglic are in this category.

Some of the authors contributed to more than one of these categories.

Now let us look at the main ideas and results of each participant.

- Hans-Paul Schwefel describes his life lessons from simulations concentrating on evolution strategies. He provides historical roots of Rechenberg's 1/5 success rule (the ratio of successful mutations to all mutations), claimed to be derived empirically and used in self-adaptation. It is used to speed up searches over the fitness function. Schwefel shows it is provable for two specific search fitness spaces: an inclined ridge and sphere. This explains why Rechenberg's rule does not always work for other fitness landscapes—where it may lead to local optima instead of global optima, stagnation, or even regression. Schwefel offers unexpected lessons: The failure rate is as high as 80 percent or even higher for optimal progress; the evolutionary process may include stagnation and recession phases instead of being gradual; the second, third, and lesser fit individuals are as important for the whole as the overall best; and self-adaptation of internal parameters and rules works only with large populations that have variety, and many more introns than exons. Note that introns are segments of DNA that are supposedly not needed to create proteins, whereas exons provide code for proteins. Introns are believed to contain only a "junk" code. However, recent results indicate that introns might provide some useful information for proteins. It is possible to observe similar analogies in genetic programming.

- Xin Yao presents an "essence of evolutionary computation." He overviews the most important basic concepts of EC and finds surprising gaps where much more research is needed. He stresses the close relation of EC with AI generate-and-test search techniques; but EC is different from AI search because EC is population-based and is stochastic in nature. He thus disagrees with Russell and Norvig who, in their popular AI textbook, claim EC is simply a special case of local search inspired by natural evolution. Yao points out EC theoretical results are spotty, although some progress on approximating capabilities of evolutionary algorithms has been reported recently. He also points out biological evolution, unlike evolutionary computation, works without defined fitness function; its power is based on interaction and co-evolution. This observation parallels a conclusion of the previous Ubiquity symposium on computation, where participants concluded that interactive computation is different from Turing machine computation. Yao does not explore this topic because too few theoretical results in co-evolution are available. At the same time, it is proved in (Burgin, 2003) that interactive computation can be more powerful than Turing machine computation, while in (Burgin, 2007), all sources of higher computational power of interactive computation are described.

- Max Garzon thoroughly analyzes evolutionary models in computing, describing the connections between evolutionary computation, natural computation, and current definitions of computer science. He starts with an important observation that although energy is important in natural evolution, there is a shift in social concerns from energy to information for evolutionary computation. The primary aim of the paper is to formulate a more coherent, comprehensive, and modern definition of computation that may address the problems articulated in (Denning, 2010) concerning the proper definition of computer science. The overarching problem is the definition of computer science requires rethinking so as to be acceptable to scientists in other fields by comparison to theirs. The author proceeds in three stages. First, he characterizes the type of phenomena that goes by the name of computation today. It is necessary to do because, at least historically, establishing a science has required the determination of an aspect and scope of natural phenomena it is interested in, usually at the exclusion of others. The thorough analysis he makes, brings Garzon to the conclusion the subject matter of computer science is information. Thus, information theory, such as the general theory of information (Burgin, 2010), is the foundation for computer science. After this, Garzon provides a general characterization of how computer science has addressed the issues of information processing. A major problem appears to be that current definitions are so restrictive as to exclude a lot of what one would intuitively deem to be "computing." The new focus reflects more closely the great many flavors of what is considered computation today, stimulate a tighter integration between them, and hopefully drive the science to a deeper definition—more embedded in natural phenomena and much more interactive with other sciences, yet with a more characteristic scope by comparison to them. In this respect, Garzon observes the classical von Neumann architecture—a processor, memory, and I/O—was utterly ignored in conventional models of computations, as there are no specific I/O structures in Turing machines, finite automata, or formal grammars. The absence of this component resulted in missed opportunities for models of computation. One of these opportunities is realized in such a super-recursive model as inductive Turing machine, which is more powerful than conventional Turing machine being able to solve the halting problem for Turing machines and giving result after the finite number of computational steps, i.e., in finite time (Burgin, 2005). Finally, Garzon discusses how an integration of important features from evolutionary and natural computing, such as nonsymbolic memory and complexification components, might provide a more distinct, sharper, and encompassing definition of computer science.

- Hongwei Mo considers evolutionary computation as a subarea of nature-inspired computing (NIC). According to the contemporary understanding, NIC has its roots in nature and include three directions. The first one uses nature principles and/or laws as a source of inspiration or metaphor for computational problem solving. In the second one, nature systems are modeled by computer emulation or simulation. In the third one, novel computer systems are built based on existing in nature biological, chemical, and physical systems. As a practical and scientific area, NIC consists of bio-inspired computing (BIC), physics inspired computing (PIC), chemical inspired computing (CIC), and social-inspired computing (SIC). BIC is a kind of computational systems or algorithms based on biological phenomena, processes, or theoretical models. PIC is based on physics phenomena, processes, or theoretical models; CIC is based on chemical processes; SIC based on culture and socio-cognition of human society. In discussing the advantages of evolutionary computation, Mo sees EC not only as a direction of NIC, but also as a part of computational thinking leading to new computational models.

- Zbigniew Michalewicz demonstrates much has yet to be done in EC in order to claim the true success story for real-world applications. There is a mismatch between the theory of EC, the toy applications to illustrate the theory, and real-world applications. According to Michalewicz, after 35 years of research, tens of thousands of papers written, hundreds of dedicated conferences (e.g., GECCO, IEEE CEC, PPSN), specific journals (e.g., *Evolutionary Computation Journal*, *IEEE Transactions on Evolutionary Computation* or *International Journal of Swarm Intelligence & Evolutionary Computation*), special sessions and special tracks on most AI-related conferences, special sessions on real-world applications, and more—but it still is not easy to find EC applications in the real world. There are no applications in daily use in some business or industry. The main reasons for the invisibility of EC-based enterprise software applications are: experiments focus on silo problems or on global optima, theory does not support practice, the research community generally dislikes business issues, and EC-based companies are rather scarce. Michalewicz hopes his contribution will be a wake-up call for the EC community.

- Walter Riofrio relates problems of evolutionary computation to new findings in the theory of biological evolution. A growing body of scientific evidence indicates biological evolution is not only vertical (Darwinian) but massively horizontal—the so-called Horizontal Gene Transfer (HGT) (Vetsigian et al. 2006). To imitate natural biological processes using nature's way to discover adaptive pathways, Riofrio conjectures, HGT could be a new way of monitoring and observing the behavior of evolutionary computations. HGT is based on the cellular origin of prebiotic evolution (Morowitz 1992;

Riofrio 2007). Riofrio introduces conceptual and theoretical aspects that one needs to take into account when exploring the essence and characteristic properties of evolutionary computation for adaptive biological processes.

- Moshe Sipper proposes "Darwinian software engineering" to help struggling conventional software engineering. Following Alan Perlis, he claims "programming is an unnatural act"; software is laborious to write, arduous to maintain, buggy, and constantly out of date. Darwinian software engineering— evolving software based on principles of natural evolution—may be the golden solution. He proposes this evolutionary approach, in which human-written programs are improved and optimized by evolutionary computations, could revolutionize software engineering. He discusses preliminary experience from evolving Java bytecode in the experimental FINCH system. He forecasts that in 50 years it will be possible to program computers by evolution. The editors hope that after 50 years the above claim will be verified in a follow-up *Ubiquity* symposium.

- David Fogel describes evolutionary game theory as a way to better understand evolution. He says evolutionary computation can produce more accurate predictions than evolutionary game theory. He stresses the game-theory notion of stable equilibria does not seem to work for evolutionary games—it is not observable in simulation experiments. The original game theory (pioneered by Steinhaus, Ulam, von Neumann, and Morgenstern work) is quite old and many of its results are obsolete, just like Turing machines in computation theory. A number of possibilities might explain this. Perhaps, the problem is not with the original von Neumann game theory, but rather with its extension to evolutionary game theory by Maynard, Smith, and Price (evolutionarily stable strategies ESS), Thomas (evolutionarily stable set), or Selten (limit evolutionarily stable strategies). These three main approaches are valid only within narrow limits, and too many simplifying assumptions are needed to apply them to reality. Perhaps, simplifications in the abstract game model caused the problem; Fogel mentions the infinity of populations, using expected payoffs instead of real payoffs, and assuming resources are replenished after each round of games (for example, an injured hawk magically recovers and continues unharmed in the next game iteration without extra cost). Perhaps, the equilibrium criterion in ESS should be replaced by some other performance measure such as orbits from chaos theory). It is not clear at this point how the evolutionary game theory should be modified, to better reflect observed phenomena. New research is needed for that.

- Lukas Sekanina's paper deals with evolvable hardware as an alternative to the top-down conventional engineering design. Some forms of evolvable hardware (unconstrained evolution directly on a chip without a circuit simulator) can be difficult or impossible to analyze or to model. Thus evolution becomes a "black box" approach, in the style of neural networks, to configure materials without understanding their internal structures. The author concludes that we do not know how and why an evolved solution works—the mapping between an abstract computational model and an evolved physical implementation remains unknown. Evolution is capable of discovering new physical behaviors that appear to be physically impossible from the point of view of current physical theories. If so, we are very close to Turing's oracle "black box" capable to solve Turing unsolvable problems. In fact, the author suggests the reconfigurable chip may implement an endless sequence of mappings F1, F2, F3, … from binary input strings to binary output strings—each potentially performing super-Turing computation. At the moment, this is only an interesting hypothesis requiring further research and experiments. If this hypothesis proves true, evolutionary computation might be more expressive than conventional computation. The author suggested in conclusions that "EC can explore 'dark corners' of design spaces that humans have left unexplored." Perhaps, this may lead to more realistic approaches to autonomous self-adaptation and self-repair, as in new autonomic grids and clouds. John von Neumann theoretically explored some of these problems for his cellular automata many years ago (von Neumann, 1966).

- Hector Zenil and James Marshall consider computational aspects of evolution and life, comparing work of biological systems to the Turing machine model. They hypothesize biological evolution is computational and should be driven by the same laws that govern information processing. They claim biology is computationally universal in the sense of Universal Turing Machine and DNA is essentially a programming language that computes the organism and its functioning. They argue under reasonable assumptions, it is possible to use evolutionary computation for inferring interesting conclusions for behavioral evolution. Zenil and Marshall consider two important features of life, robustness and fitness, connecting these features to algorithmic probability and the thermodynamics of computation. They attempt to employ Kolmogorov algorithmic complexity to measure robustness of biology in the sense of preserving by an organism its phenotype in spite of random mutations or transitional environmental changes. The authors employ computational dynamics to derive from energy, via by thermodynamics equations, information in the form of the fitness function. They suggest these disciplines may be capable of modeling key features of living organisms, as well as for developing new algorithms for evolutionary computation. However, despite the interesting analogies and insights, the paper stops short of defining a more complete theory of biological computation.

- Darko Roglic starts with an observation that a group of computer scientists and biologists (Banzhaf, *et. al.*., 2006) have recently presented limitations of EC with respect to such biological concepts as complexity and robustness. Roglic demonstrates EC encounters essential difficulties in projecting these biological concepts by conventional algorithms bounded by the Church-Turing thesis are used. The reason is such models give an oversimplified description of biological processes. For instance, Longo and Tendero (2007) explain the programming paradigm based on the Turing machine model cannot capture the relations that should link the genome to the phenotype. Roglic believes super-recursive algorithms—which are, by their definition (Burgin, 2005), more powerful than Turing machines and other recursive algorithms—can eliminate these limitations and provide adequate models of biological processes. For instance, super-recursive algorithms can naturally model systems made of parts that continually turn over, renewing themselves. This conjecture correlates well with theoretical results, which show super-recursive algorithms provide more possibilities for applications of EC (Burgin and Eberbach, 2008; 2009; 2011).

- David Wolpert overviews the no free lunch theorems (NFLs) and clarifies many of their incorrect interpretations. A number of “no free lunch” (NFL) theorems demonstrate that for any algorithm, any elevated performance over one class of problems is offset by performance over another class. In the 1990s, the author and Macready introduced NFLs firstly for machine learning and next for optimization (including EC optimization). No free lunch theorems are important contributions to EC. They showed random ad hoc “improvements” to EC operators could not provide universal improvements compared to existing operators. NFL theorems are interpreted that if one search algorithm outperforms another in a given domain, then in a complementary domain it will be opposite. Thus on average all search algorithms/perform equally bad or well. Wolpert claims hill climbing (e.g., genetic algorithms) without resampling is as effective at optimization as a blind random search without resampling. He gives many other examples of NFL misinterpretation. He concludes with directions for future research on NFLs. For example, NFLs for co-evolution have been only slightly researched, and NFLs averaging over algorithms rather than fitness functions have not been explored at all. NFLs appear to be more important for the EC theory than traditional schema theorems that are more often described in the textbooks and monographs. NFLs deserve more popularization.

Despite the range of topics covered by authors, other EC topics are not covered here. Examples are genetic programming, evolutionary programming, particle swarm optimization, ant colony optimization, multi-objective optimization, evolutionary robotics, artificial immune systems, or evolutionary artificial neural networks.

One of our big conclusions is the EC area suffers from a lack of theory, which is only at the beginning of its development. This makes it very hard to determine the range of validity of specific methods. The successes to date are in limited areas and have not attracted more practical applications.

Despite these limitations, EC was, and is, very popular. It is extending the horizons of computer science research beyond the canon of conventional computing, which is represented by Turing machines and other recursive algorithms (see, e.g., Burgin and Eberbach, 2008; 2009; 2009a; 2011; Eberbach, 2002; 2005; Eberbach and Burgin, 2009, and the 2011 *Ubiquity* symposium on computation). We hope this symposium draws more attention and gets more people interested in studying and utilizing evolutionary computation.

Acknowledgements

The editors of the symposium would like to thank Peter Denning for multiple comments, advice and inspiration.

Bibliography

Banzaph, W., Beslon, G., Christensen, S., Foster, A., Kepes, F., Lefort, V., Miller, F.J., Radman, M., and Ramsden, J. Guidelines: From artificial evolution to computational evolution: a research agenda. *Nature Reviews Genetics* 7, 9 (2006).

Burgin M. From Neural networks to Grid Automata. In *Proceedings of the IASTED International Conference "Modeling and Simulation"* (Palm Springs, CA), 2003, 307-312.

Burgin, M. *Super-recursive Algorithms*. Springer, New York, 2005

Burgin, M. Interactive Hypercomputation. In *Proceedings of the 2007 International Conference on Foundations of Computer Science* (FCS'07), CSREA Press (Las Vegas, NV). 2007, 328-333.

Burgin, M. *Theory of Information: Fundamentality, Diversity and Unification*. World Scientific, New York, 2010.

Burgin, M., Eberbach, E. Cooperative Combinatorial Optimization: Evolutionary Computation Case Study. *BioSystems* 91 (2008), 34-50 .

Burgin, M., Eberbach, E. On Foundations of Evolutionary Computation: An Evolutionary Automata Approach, *Handbook of Research on Artificial Immune Systems and Natural Computing*. IGI Global, 2009, 342-360.

Burgin, M. Eberbach, E. Universality for Turing Machines, Inductive Turing Machines and Evolutionary Algorithms. *Fundamenta Informaticae* 90 (2009a), 1-25.

Burgin M., and Eberbach E. Evolutionary Automata: Expressiveness and Convergence of Evolutionary Computation. *Computer Journal* 2011. (doi:10.1093/comjnl/bxr099)

Denning, P. What is Computation? The AMC's Ubiquity Symposium, *Ubiquity*, 2010 (at http://ubiquity.acm.org/article.cfm?id=1880067)

Eberbach E. On Expressiveness of Evolutionary Computation: Is EC Algorithmic? In *Proc. 2002 World Congress on Computational Intelligence* WCCI'2002 (Honolulu, HI). 2002, 564-569.

Eberbach, E. Toward a theory of evolutionary computation*. BioSystems* 82 (2005),1-19.

Eberbach, E. and Burgin, M. Evolutionary Automata as Foundation of Evolutionary Computation: Larry Fogel Was Right. In *Proc. 2009 Congress on Evolutionary Computation* CEC'2009, Trondheim, 2009, 2149-2156

Longo, G. and Tendero, P.E. The Differential Method and the Causal Incompleteness of Programming Theory in Molecular Biology. *Foundations of Science* 12, 4 (2007), 337-366.

Morowitz, H.J. *Beginnings of Cellular Life*: *Metabolism Recapitulates Biogenesis*. Yale University Press, New Haven, 1992.

Riofrio, W. Informational Dynamic Systems: Autonomy, Information, Function. In C. Gershenson, D. Aerts, and B. Edmonds (eds.), *Worldviews, Science, and Us: Philosophy and Complexity*. World Scientific, Singapore, 2007, 232-249.

Vetsigian K., Woese C., Goldenfeld N. Collective evolution and the genetic code. *PNAS*103, 28 (2006), 10696-10701.

Von Neumann, J. *Theory of Self-Reproducing Automata*, A.Burks (ed.), University of Illinois Press, 1966

About the Editors

Dr. Mark Burgin received his M.A. and Ph.D. in mathematics from Moscow State University and Doctor of Science in logic and philosophy from the National Academy of Sciences of Ukraine. He was a Professor at Institute of Education, Kiev; at International Solomon University, Kiev; at Kiev State University, Ukraine; and the Head of the Assessment Laboratory, Research Center of Science at the National Academy of Sciences of Ukraine. Currently he is working at UCLA. Dr. Burgin is a member of New York Academy of Sciences and a Senior Member of IEEE and of the Society for Computer Modeling and Simulation International. He is the Chief Editor of the journal *Integration*, Chief Editor of the journal *Information*, an Associate Editor of *Ubiquity* and of the *International Journal on Computers and their Applications*. Dr. Burgin is doing research, has publications, and taught courses in mathematics, computer science, information sciences, artificial intelligence, philosophy, methodology of science and logic. His research areas in computer science and artificial intelligence include Theory of Algorithms and Computation, Complexity, Evolutionary Computations, Mathematical Theory of Information Technology, Software Metrics, Software Correctness, Concurrent Computation and Distributed Systems, Computer Modeling and Simulation, Interactive Computation, Computational Schema Theory, Algorithmic Information Theory, General Theory of Information, Mathematical Logic, Knowledge Representation, Knowledge Acquisition and Data Mining, Knowledge and Information Dynamics, Representational Schema Theory, Non-monotonic Reasoning, and Inconsistent Knowledge Systems. His recent publications include such books as "*Theory of Information*" (2010), "*Measuring Power of Algorithms, Computer Programs, and Information Automata*" (2010), and "*Super-recursive Algorithms*" (2005). He originated such theories as the general theory of information, mathematical theory of technology, system theory of time, theory of logical varieties, theory of named sets and neoclassical analysis (in mathematics) and made essential contributions to such fields as foundations of computer science, theory of algorithms and computation, information theory, theory of knowledge, theory of negative probabilities, theory of intellectual activity, and complexity studies.

Dr. Eugene Eberbach is Professor of Practice at Department of Engineering and Science, Rensselaer Polytechnic Institute, Hartford, USA. He has Ph.D. (1982) and M.Sc. and Eng. (1977) degrees both from Warsaw University of Technology, Poland. He held various academic and industrial positions in USA, Canada, United Kingdom and Poland, including University of Massachusetts Dartmouth, Acadia University, University of Memphis, University College London, Rzeszow University of Technology, WSK "PZL-Rzeszow" and Applied Research Lab, Penn State University. In late 1980s he worked on new non-von Neumann 5th Generation Computer Architectures at University College London. In 1990-2000s he worked on distributed autonomous underwater vehicles with support of ONR. In Canada and USA he introduced Calculus of Self-modifiable Algorithms and $-Calculus process algebra for automatic problem solving under bounded resources with support of NSERC and ONR. He proposed two super-Turing models of computation: $-Calculus and Evolutionary Turing Machines. His current work is in the areas of process algebras, resource bounded optimization, autonomous agents and mobile robotics. General topics of interest are new computing paradigms, languages and architectures, distributed computing, concurrency and interaction, evolutionary computing and neural nets. Dr. Eberbach is the author of more than 150 publications in the above areas and he has been a recipient of 17 external research grants. More information about projects, publications, courses taught can be found at http://www.ewp.rpi.edu/~eberbe.

DOI: 10.1145/2555235.2555240