Association for Computing Machinery
Advancing Computing as a Science & Profession

Ubiquity, an ACM publication
November, 2012

Ubiquity Symposium

Evolutionary Computation and the Processes of Life

Evolutionary Computation as a Direction in Nature-inspired Computing

by Hongwei Mo

Editor's Introduction

In this article evolutionary computation (EC) is considered as a kind of nature-inspired computing (NIC) paradigm. EC not only has great effect on the development of computing methods from structure to process, but also has great effect on many aspects of our society as a ubiquitous or general computational thinking. EC is still one of the best choices for problem solving among all methods when people face more and more complex problems.

Editor

Ubiquity Symposium

Evolutionary Computation and the Processes of Life

Evolutionary Computation as a Direction in Nature-inspired Computing

by Hongwei Mo

Computation is a complex concept, which has been discussed for nearly a century. Our understanding of computation is not only drawn from mathematics, but also from the development of computer science, electronic science, and even physics and biology.

In recent years, we are stilling trying to reveal the secret of computation. According to Burgin computation is a process of information transformation, which is controlled by an algorithm, while an algorithm is a system of rules for a computation [1].

Peter Denning gives the following definition of computation [2]: A representation is a pattern of symbols standing for something. A representation that stands for a method of evaluating a function is called an algorithm. A computation is an information process in which the transitions from one element of the sequence to the next are controlled by a representation.

Based on this definition of computation, in a broad sense, evolutionary computation (EC) is an information transformation process controlled by rules of evolutionary algorithms, while evolutionary algorithm is a system of rules inspired by biology evolution for computation.

The main difference between evolutionary computation and computation lies in that the former is a kind of computation technology inspired by biology evolution. Although there are many kinds of technologies from mathematics and computer science, researchers pay more attention to EC since it is simple and efficient in solving many different kinds of problems in different fields.

EC as a Direction of NIC

Generally speaking, EC includes genetic algorithm (GA) [3], evolutionary programming [4], evolutionary strategies [5], genetic programming [6], differential evolution [7] and estimation of distribution algorithm [8]. GA is based on genetic science and natural selection and it attempts to simulate the phenomenon of natural evolution at genotype level while evolutionary strategies and evolutionary programming simulate the phenomenon of natural evolution at phenotype level. So, it is no doubt that EC is a kind of nature inspired computing technology and also the earliest one. Now we should put EC in a bigger picture in order to understand it more deeply.

Natural computing is a field of developing new computational tools—in software, hardware or "wetware'"—based on or inspired by nature mechanisms for problem solving [9]. It includes three branches: The first one is nature inspired computing (NIC), which uses nature principles or laws as a source of inspiration or metaphor for problem solving. The second one is nature modeling, which is the emulation or simulation of nature systems by computer for researching nature or livings. The third one is natural materials based computing, which is for developing novel computer system by means of biology, physics, and chemical.

NIC can be further classified into bio-inspired computing, physics-inspired computing, chemistry-inspired computing, and social-inspired computing according to different sources of inspiration [10]. Bio-inspired computing is a kind of computational systems or algorithms based on biological phenomena, processes, or theoretical models. Similarly, physics-inspired computing is based on physics phenomena, processes, or theoretical models. Chemistry-inspired computing is based on chemical processes; and social-inspired computing is based on culture and socio-cognition of human society. So it is clear that EC is a class of bio-inspired computing based on the biological evolution or biological behavior of members in a population.

Since the 1980s, more and more NIC algorithms were developed following EC. Among them, evolutionary algorithms (EAs) and swarm intelligence (SI) algorithms are two important classes of population-based optimization algorithms. SI algorithms share many common characteristics with EAs and are also regarded to belong to the algorithms family of EC, including ant colony optimization [11] and particle swarm optimization [12]. As well as immune algorithm, artificial bees colony optimization, artificial fish swarm optimization, bacterial algorithms, biogeography-based optimization algorithm [13], and so on. Besides EC mentioned above, some other NIC approaches including simulating annealing, culture algorithm, photosynthetic algorithm, membrane computing [14], intelligent water drops algorithm, and chemical-reaction optimization [15] have also been developed. Their inspiration sources can be found directly from their names.

These inspiration sources are so diverse that they cover nearly all nature systems from biology to chemical, from macroscopic worlds to microscopic ones. While EC is the earliest bio-inspired computing or nature inspired computing, which is inspired by Darwin evolution theory. As we know, as early as 1962, Holland and his collaborators developed GAs. He was the first to use crossover and recombination strategies for modeling adaptive system. During the same period, Ingo Rechenberg and Hans-Paul Schewefel developed evolutionary strategy. In 1966, Lawrence J. Fogel together with A.J. Owen and M.J. Walsh developed evolutionary programming. All of these have evolved into the much wider discipline of evolutionary computation.

Although there have been so many NIC approaches inspired by different sources from nature, it is no doubt that EC had been leading the direction of NIC for the past half century. In other words, it has been EC that inspired people to gradually develop various NIC approaches during the past two decades.

At first, generally speaking, we can classify optimization algorithms into heuristics and meta-heuristics. Heuristic means to “find” or to “discover by trial and error.” Meta-heurstic means “beyond” or “higher level” heuristics because meta-heuristics generally perform better than simple heuristics. On one hand, heuristics are different from meta-heuristics in that the former are tailor-made for specific problems. They may be able to solve some problems very well but may give poor solutions to others. On the other hand, well-designed meta-heuristics can be applied to a broader range of problems and results in good performance, such as EC. In fact, EA is the earliest and the most important meta-heuristic algorithm. As we know, it has many advantages and disadvantages. Most of the NIC approaches try to overcome the disadvantages of EC and beat it. We can see that the synergy between exploration and exploitation has been a prominent and central issue of all NIC approaches for improving their performance. So there is the rise of memetic algorithms [16], a category of optimization techniques, which feature the explicit exploration-exploitation coordination. To some extent, we can say it is just the disadvantages of EA that become the power of developing more effective optimization algorithms for researchers and promote the development of NIC in the past few decades.

Secondly, EA is also the earliest population-based optimization algorithm. Population-based optimization algorithms find near-optimal solutions to the difficult optimization problems by motivation from nature. A common feature of them is the population consisting of possible solutions to the problem is modified by applying some operators on the solutions depending on the information of their fitness. Hence, the population is moved toward better solution areas of the search space. In the EA, there is competition among agents on survival of the fittest. These paradigms consist of simple elements that can solve complicated problems of the real world when working together. For example, the algorithms like SI have a cooperative nature in comparison with GA. The most important similarity between SI and the GA is that they have the interactive population.

Thirdly, EC inspires people to design new and efficient algorithms. One hand, nature selection is the basic principle on which EC is based. For the design of new NIC algorithms—although there are many ways of achieving a good formulation of them—two good and successful ways are still based on two basic ways of natural selection: explore new strategies and inherit the fittest strategies [17]. Therefore, the first way of developing new NIC algorithms is to design new ones using discoveries. The second way is to formulate by hybrid and crossover. So the basic principles of EC still work in the designing of new NIC algorithms.

On the other hand, fitness is one of the most important concepts in the field of NIC, but no one would doubt that it is firstly used in EC. And in most developed NIC algorithms, people would like to use fitness or similar concepts to measure the quality of solutions in their algorithms, though they have different processes, steps and nature sources of inspiration. Even in the first paper of PSO, which is also a very famous NIC approach, it said that each particle keeps track of its coordinates in hyperspace, which are associated with the best solution (fitness) [12].

In one word, EC has great effect on the development of NIC in many aspects from the idea of nature inspiration to structure and process.

Advantages of EC

According to the “No Free Lunch Theorem,” every meta-heuristic has equal performance on average on solving computational problems [18]. No one algorithm can always, on average, surpass the others in all possible optimization problems. So we do not need an algorithm that is average over all possible functions for a given optimization problem. Our research focuses on finding the best and most efficient algorithm for a given specific problem. However, the “spectrum” of problems is too huge to find the best match for each of them. So how to decide which one is more suitable to the specific problem? It should be answered by the principle of Occam’s razor, that is, if an algorithm is simply enough to find a satisfying or suitable solution (not to be the optimal) for a specific problem, it is the best one for the problem. One hand, in general, for simple analytic problem, the stationary conditions, extreme points, and calculus-based algorithms can be used at first. For simple and complex analytical function optimization problems, if analytical methods don’t work well, NIC algorithms can be used. While in these algorithms, EAs can be used at first if they are suitable for the problems since they are simple, mature, and robust. If EAs don't work, then we can try the other NIC algorithms suitable for the problems solved. On the other hand, for large scale, nonlinear, global optimization problems, we tend to use meta-heuristic algorithm.

So EC is the first choice and the most competent approach among all the NIC approaches in problem solving. Some statistics show that a vast majority of Fortune 500 companies are now using EC to solve hard combinatorial optimization problems such as planning, data fitting, and scheduling. Till now, no other NIC algorithms could surpass EC in applications. So it is also the best one in practice. Ultimately, we hope that a kind of intelligent evolutionary algorithm, which can evolve and optimally adapt to solve NP-hard optimization problems efficiently and intelligently, will appear in the future.

EC as Computational Thinking

About what is computing thinking, Priami states "the basic feature of computational thinking is abstraction of reality in such a way that the neglected details in the model make it executable by a machine [19]."

Aho has discussed computing and computational thinking. He said "As we shall see, finding or devising appropriate models of computation to formulate problems is a central and often nontrivial part of computational thinking [20]."

Although EC and the artificial neural network were born in last century, at that time, we still know very little about computing thinking. In fact, both of them are some of the earliest BIC approaches. EC and the other NIC approaches have found wide applications in many engineering fields. NIC algorithms are always based on some particular mechanisms of the natural world. So the development of NIC is based on a deep view into nature by researchers. Now, formulation of appropriate computation models of natural processes for solving problems had become general thinking in the intersection of nature science and computer science. As discussed above, it is no doubt that EC has great influence on the development of NIC, more importantly it had become an ubiquitous computation idea by which people understand the process of nature, life, mind, society—even art, management, economy and problems solving in these areas. Today, EC is the most popular intelligent approach for solving different kinds of problems, not only for optimization problem. We have seen many new fields based on evolution computing grow in the past three decades, including evolution music, evolution design [21], evolution art [22], evolution neural network [23], evolution robotics, and so on. And it is even used to invent new things as an inventing way. Till now, no other NIC approach has influence as great as that of EC on our society from so many aspects, though such influence is potential, not as explicit as that of the invention of the telephone and the car. EC does change our lives at the back. To give some quantifiable indication of this impact, on June 22, 2011, Holland's book attracted 24,597 citations (according to Google Scholar), while David E. Goldberg's more recent account had 38,706 citations [24]. These citations come from varied fields of engineering and design, and there are hundreds of books written in different fields.

Conclusion

According to Burgin and Eberbach: "In a broad sense, evolutionary algorithms are rules that control any kind of evolutionary computations [25]." Here, this viewpoint can be extended: Evolutionary algorithms are the meta-inspiration rules that inspire any kind of nature inspired computing. Although they have different inspiring sources, the objective of them is similar. Since we meet more and more complex problems in the development of society, what we should do is to continuously find the right inspiration sources for developing right algorithms for correct problem solving. But among all the competitive NIC approaches, we should never forget that EC is still one of the best choices for any potential complex problems.

Acknowledgement

Thanks very much to Prof. Mark Burgin for the invitation for the Evolutionary Computation Symposium. And Thanks very much to the reviewers for their good suggestions. The work is supported by National Nature Science Foundation of China under grant No. 61075113, the Excellent Youth Foundation of Heilongjiang Province of China and the Fundamental Research Funds for the Central Universities No.HEUCFZ1209.

References

[1] Burgin, M. *Super-recursive Algorithms*, Springer, Berlin, 2005.

[2] Denning, P. What is Computation: Opening statement. *Ubiquity* November (2010); http://ubiquity.acm.org/article.cfm?id=1880067.

[3] Holland, J.H. *Adaption in Natural and Artificial Systems*. MIT Press, Cambridge, 1975.

[4] Fogel, L.J., Owens, A.J., and Walsh, M.J. *Artificial Intelligence Through Simulated Evolution*. Wiley, 1966.

[5] Beyer, H.G. *The Theory of Evolution Strategies*. Springer. Berlin, 2001.

[6] Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, 1992.

[7] Storn, R., and Price, K. Differential Evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 4 (1997), 341-359.

[8] Pelikan, M. Linkage problem, distribution estimation, and bayesian networks. *Evolutionary Computation* 8, 3 (2006), 311-340.

[9] De Castro, L. N. *Fundamentals of Natural Computing*. Champman & Hall/CRC, Florida, 2006, 3-20

[10] Mo, H.W. Research Development on Nature Inspired Computing. *Journal of Intelligence Systems* (In Chinese) 6, 6 (2011), 11-13.

[11] Dorigo, M., Manianiezzo, V., and Colorni, A. The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Trans. Sys. Man and Cybernetics* 26, 1(1996), 1-13.

[12] Eberhart, R. C. and Kennedy, J. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micromachine and Human Science* (Nagoya, Japan). 1995, 39-43.

[13] Simon, D. Biogeography-based optimization. *IEEE Trans on Evolutionary Computation* 12, 6 (2008), 702-713.

[14] Paun, A., and Paun, G. The power of communication: P systems with symport/antiport. *New Generation Computing* 20, 3 (2002), 295-305.

[15] Lam, A. Y. S., and Li, V. O.K. Chemical-reaction-inspired metaheuristic for optimization. *IEEE Trans on Evolutionary Computation* 14, 3 (2010), 381-400.

[16] Chen, X. S., Ong, Y. S., Lim, M. H., and Tan, K. C. A multi-facet survey on memetic computation. *IEEE Trans on Evolutionary Computation* 15, 5 (2011), 591-607.

[17] Yang, X.-S. *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2008.

[18] Wolpert, D.H., and Macready, W.G. No free lunch theorems for optimization. *IEEE Trans on Evolutionary Computation* 1, 1 (1997), 67-82.

[19] Priami, C. Computational thinking in biology. In *Transactions on Computational Systems Biology VIII.* Springer, Berlin, 2007, 63-76.

[20] Aho, V.A. Computation and computational thinking. *Ubiquity* January (2011); http://ubiquity.acm.org/article.cfm?id=1922682

[21] Bentley, P. J. *Evolutionary Design by Computers*. Morgan Kaufman, San Francisco, 1999.

[22] Fernandez, T. Evolutionary Art Gallery; http://www.cse.fau.edu/~thomas/GraphicApDev/ThomasFernandezArt.html

[23] Yao, X. Evolving artificial neural networks. *Proceedings of IEEE* 87, 9 (1999), 1423-1447.

[24] Harman, M. Software engineering meets evolutionary computation. *Computer* 44, 10 (2011), 31-39.

[25] Burgin, M., and Eberbach, E. Evolutionary Computation and the Processes of Life, Opening Statement, Ubiquity Symposium. *Ubiquity* August (2012); http://ubiquity.acm.org/article.cfm?id=2351437

About the Author

Hongwei Mo is Professor of Automation College at Harbin Engineering University, China. He received his Ph.D. at the same university in 2005. He was a visiting Scholar of UC Davis from 2003 to 2004. His interests include nature inspired computing, artificial immune systems, evolutionary computation, machine learning, data mining and robotics. He has published 50 papers on artificial immune systems and nature inspired computing in international journals and conferences. Mo was the guest editor of Special issue on Nature inspired computing and applications for *the Journal of Information Technology Research*. He was the author of three books in Chinese and he is the editor of *'Handbook of Artificial Immune Systems and Nature inspired computing: Applying Complex Adaptive Technologies*'. Mo is a member of IEEE Computing Intelligence Society, IEEE Robotics and Automaton Society. He was also the program committee member in more than 15 international conferences. He serves as the member of the editorial review board of the *Journal of Information Technology Research* and *International Immune Computation, Journal of Man, Machine and Technology, Progress in Intelligent Computing and Applications*.

DOI: 10.1145/2390009.2390011