Editorial Introduction

Welcome to the first issue of Volume 5 of *Evolutionary Computation*. As the new Editor-in-Chief of the journal, I owe a debt to Ken De Jong, the journal's Editor-in-Chief for the first four years, for creating a solid foundation for *Evolutionary Computation*; Ken has also continued to be helpful with Volume 5 as I learn on the job.

Evolutionary Computation came about because different research communities interested in computational models based on principles of evolution came together with the goal of having a single journal that could bring together researchers and promote the general interests of the field. I was part of the committee that helped to initiate and launch the journal in 1991, 1992, and 1993. Since that time the field has seen an explosion of new activity and a large increase in the number of researchers interested in the field of evolutionary computation. There is now at least one other journal dedicated to evolutionary computation and numerous conferences, and the field has even spawned a new subdiscipline more or less within the last four years: genetic programming.

I don't expect there to be major changes in *Evolutionary Computation* in the near future, but there will be some growth as the journal adds new Associate Editors and Editorial Board Members.

In the first issue of *Evolutionary Computation*, Ken De Jong outlined the topics to be covered by this journal:

- mathematical foundations of evolutionary computation
- biological foundations of evolutionary computation
- characterization of problems suitable for evolutionary computation approaches
- parallel computation of evolutionary algorithms
- implementation issues of evolutionary algorithms
- evolutionary approaches to optimization
- evolutionary approaches to machine learning
- evolutionary approaches to artificial life
- evolution of neural networks
- evolution of emergent properties
- applications of evolutionary computation to problems in science, engineering, economics, etc.

This list remains as fresh and relevant today as it was four years ago, and I believe that it still reflects the focus of the journal as well as the interests of the evolutionary computation community. In particular, I want to emphasize that the journal should not neglect applications. In the past year or so I have seen several examples of ground-breaking applications that use evolutionary computation methods, such as the work by Falkenauer on bin packing problems. In applications as well as theory, *Evolutionary Computation* should be the home of work that redefines the state of the art.

In the current issue are four papers that touch on several themes central to the journal.

The paper by Christopher Rosin and Richard Belew looks at "coevolution" in the context of evolutionary computation. Coevolution involves two or more distinct non-interbreeding populations that coevolve such that the fitness of one population depends on some characteristic of the other population. An early example of the use of coevolution in evolutionary computation was the coevolution of test problems in conjunction with the evolution of sorting networks by Hillis (1991); in this case the fitness of the sorting networks was determined by how well they did against the evolving test problems, and the fitness of the population of test problems was determined by how well they were able to find flaws in the sorting networks. Rosin and Belew provide new experiments working with games, as well as new techniques for competitive coevolution and new mathematical insights into the use of coevolutionary methods.

The paper by Christopher Houck, Jeffery Joines, Michael Kay, and James Wilson is an empirical investigation that looks at how evolutionary methods might be combined with more traditional local search methods. In particular, if local search is used to "improve" an individual produced via reproduction, should the improvements only change the fitness of the individual (i.e., the phenotypic behavior), or should the improvements change the genotype of the individual in a "Lamarckian" sense so that these improvements might be directly passed along to offspring? This work suggests that partial Lamarckianism appears to yield the best results. The paper also relates to a recent special issue of *Evolutionary Computation* on the Baldwin effect and, like the Rosin and Belew paper, should also be of interest to researchers concerned with the biological foundations of evolutionary computation and in artificial life.

The paper by Shigeyoshi Tsutui, Yoshiji Fujimoto, and Ashish Ghosh entitled "Forking Genetic Algorithms: GAs with Search Space Division Schemes" presents some ideas that might be used for parallelizing an evolutionary algorithm, but the main theme of the paper really relates to representation issues. "Forking" in this case doesn't just generate a new subpopulation, that subpopulation can in fact be specialized to use a different representation or to change the precision with which parameters are represented. In this sense, the paper has similarities with dynamic parameter encoding (Schraudolph and Belew, 1992), as well as the ARGOT strategy (adaptive representation genetic optimizer technique) introduced by Craig Schaefer at the Second International Conference on Genetic Algorithms 10 years ago (Schaefer, 1987). The use of a reduced *neighborhood hypercube* is also a central feature of delta coding (Mathias & Whitley, 1994). In the paper on the forking genetic algorithm (GA), different "genotypic" and "phenotypic" schemes for changing the problem representation are discussed and evaluated empirically. Another use of the forking GAs is for multimodal function optimization.

The paper by Cem Hocaoğlu and Arthur C. Sanderson also looks at multimodal function optimization and also has the potential for parallel implementation. The paper examines the use of cluster analysis as well as the minimal representation criterion, which is based on the "minimal complexity of a program that would reproduce the observed data on the output tape when executed on a deterministic Universal Turing Machine (UTM)." The clusters that are produced correspond to "species" exploiting different local optima in the search space. No a priori information about the fitness landscape is required. In additional to being applied to several example problems, the methods are applied to path-planning problems in robotics.

Finally, this issue also contains a report on the International Conference on Evolvable Systems that summarizes current efforts in this relatively new area of research.

Darrell Whitley
Computer Science Department
Colorado State University
Fort Collins, CO 80523
ecj@cs.colostate.edu

References

Hillis, W. D. (1991). Co-evolving parasites improve simulated evolution as an optimization procedure. In S. Forrest (Ed.), *Emergent computation* (pp. 228–234). Cambridge, MA: MIT Press.

Mathias, K., & Whitley, D. (1994). Changing representations during search: a comparative study of delta coding. *Evolutionary Computation, 2(3)*, 249–278.

Schaefer, C. (1987). The ARGOT strategy: adaptive representation genetic optimizer technique. In J. J. Grefenstette (Ed.), *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications* (pp. 50–55). Hillsdale, NJ: Lawrence Erlbaum Associates.

Schraudolph, N. N., & Belew, R. K. (1992). Dynamic parameter encoding for genetic algorithms. *Machine Learning*, *9*,9–21.