Editorial Introduction

Welcome to the second issue of volume 2 of *Evolutionary Computation*! As I noted in the previous issue, we are in the process of expanding the journal in several directions. We've completed our first special issue (V2N1) and have proposals for several more. In addition to the long technical papers, we are planning to include short technical notes, book reviews, and summaries of special workshops. I'm pleased to draw your attention to Thomas Bäck's review of the ICGA-93 Proceedings in this issue. By the time you are reading this, the 1994 summer conferences and workshops will have taken place. I hope others will follow Thomas's example and provide the readership with summaries and reviews of these activities.

Without any planning on my part, this issue has turned out to be very GA-oriented. It contains a collection of papers from the GA community that provide a nice view of current issues and applications. The first paper, by Aizawa and Wah, summarizes and extends our methodologies for applying GAs in a noisy environment. In a nutshell, the issue here is in deciding how many resources should be spent getting reliable estimates of fitness. The standard wisdom is that, since GAs are relatively insensitive to noisy feedback, it's not worth spending much effort improving fitness extimates. Typical applications use an estimation procedure involving averaging the values of a small fixed number of evaluations. Aizawa and Wah bring some resource scheduling theory to bear on the problem, and show that significant performance improvements can be obtained by dynamically scheduling the GA resources used to improve feedback reliability.

The area of combinatorial optimization has provided a continuing source of examples relating to the importance of choosing good GA representations and operators. GA representation theory still lags behind the GA practitioner, leaving room for a good deal of clever design when a new and difficult class of problems is attacked. The Falkenauer paper provides a nice summary of this process for what he calls grouping problems (which include the familiar bin packing problems). He carefully shows why standard bit string and other common representations are not effective. Using our present understanding about representation, he then develops and tests a new representation that is considerably more effective over a range of sample grouping problems.

Because of the practical importance of many NP-hard problems, considerable effort continues to be put into finding good (low polynomial time) heuristics for them. The paper by Maini et al. provides another example of the potential for using GAs to achieve these goals. In their case, the problem is one of decoding linear block codes used heavily in communications systems. Using a domain-specific crossover operator, they are able to perform as well as or better than the current state of the art with respect to error rates, and significantly better with respect to memory requirements.

As a field I think it's fair to say that we haven't really succeeded in adapting and using visualization techniques as a tool for analyzing the dynamical behavior of our evolutionary algorithms. The final technical paper in this issue is an exception in this regard and involves a fascinating study by Juliany and Vose of some of the properties of the "G" function developed in Vose's earlier work on modeling simple GAs as Markov processes. The G function maps a given population into the "expected" next population that would be produced by a GA on average. Hence, given an initial population, G can be used to explore the expected population

trajectories produced by GAs. Since G inverse also exists, one can explore these trajectories both forward and backward in time.

An immediate consequence of this is to think of GAs as a dynamical system, and study the fixed points of G, basins of attraction, and convergence velocities. According to Juliany and Vose, the good (bad?) news is that nothing too interesting happens within the region of feasible populations (i.e., the GA is a well-behaved dynamical system). However, if we apply G to infeasible regions (populations containing a negative number of instances of some strings), then we begin to see several interesting features, including fascinating fractals involving convergence velocities.

Enjoy!

Kenneth De Jong
George Mason University
Fairfax, VA 22030 USA
E-mail: ecj@cs.gmu.edu