Book Review

Genetic Programming III - Darwinian Invention and Problem Solving

John R. Koza **Forest H. Bennet III** **David Andre** **Martin A. Keane**

San Francisco, CA: Morgan Kaufmann Publishers, Inc.
1999; hardbound, ISBN 1-55860-543-6.

Reviewed by
Peter Nordin
Chalmers University

The quest for automatic Programming is the holy grail of artificial intelligence. The dream of having computer programs write other useful computer programs has haunted researchers since the nineteen fifties. In *Genetic Programming III - Darwinian Invention and Problem Solving (GP3)* by John R. Koza, Forest H. Bennet III, David Andre, and Martin A. Keane, the authors claim that the first inscription on this trophy should be the name Genetic Programming (GP). GP is about applying evolutionary algorithms to search the space of computer programs. The authors paraphrase Arthur Samuel of 1959 and argue that with this method it is possible to *tell the computer what to do without telling it explicitly how to do it.*

The main hypothesis of the book is that GP is not only the first instance of true automatic programming but also creative to such an extent that it competes with humans in solving very hard problems, and, therefore, the solutions produced by GP can sometimes be called "inventions"—thus the name Darwinian Invention Machine. The book starts by listing sixteen proposed attributes of any automatic programming system. The attribute list begins with obvious properties such as, the ability to produce entities that can run on a computer. It continues by describing components of full computer programs and ends by expressing fuzzier concepts such as, applicability, scalability, and competitiveness with human-produced results. The authors argue that GP definitely possesses most of the sixteen attributes and, at least to some extent, possesses the remaining few. The last attribute, human competitive results, is in turn defined by a list of eight properties where each of them gives enough evidence to conclude competitiveness to results produced by the intellect of a human. This list includes concepts such as whether the results are patentable, publishable in scientific journals, or better than best known human solutions. GP3 reports fourteen experiments by the authors where they claim that results produced by GP fulfill one or more of these properties and, thus, are competitive with that of a skilled human such as an engineer, mathematician, designer, or programmer. Examples of results with the "Darwinian invention quality" include sorting networks, analog electrical circuit synthesis and creation of motifs for protein family detection. Pointers are also given to human competitive solutions evolved by other researchers.

The volume is Koza's third book in the series on genetic programming where the first book actually sparked the creation of the research field seven years ago. In line with the format of the previous volumes, this book consists of a description of the state of the art of genetic programming exemplified by applications and problems performed by John Koza and colleagues. Also, like the previous books, it is an impressive brick volume of 10 parts, 64 chapters, and more than 1100 pages. Koza was the sole author of the two previous books, but in this book he is joined by his colleagues Forest H. Bennet III, David Andre, and Martin A. Keane. Following the outline of its predecessors, the book starts with a thorough background and introduction to genetic algorithms, LISP programming, basic GP, and pointers to other information about GP and related fields. To achieve true automatic programming, it is important that the user supply as few parameters as possible. Part 3 deals with what the authors call "architecture altering operations" which can be seen as a set of high level genetic operators that manipulate subroutines, iteration structures, recursion, and internal memory. The use of these operations frees the user from the requirement to supply size, shape, and type of the expected program solutions. Instead, these properties can evolve just like the rest of the solution program. Some of these architecture altering operations are inspired by biology, such as OhnoÓs gene duplications, while other operations seem more artificial. The authors evaluate these GP additions on a wide range of test problems ranging from standard Boolean problems to innovative work on genome analysis. Part 4 introduces a beast called the Genetic Programming Problem Solver (GPPS)—a system that combines architecture altering operations with a fixed function set to create an automatic programming blackbox capable of solving hard problems with minimum human interaction. GPPS and evolution creates programs containing variable numbers of inputs, subroutines, iteration structures, recursion elements, and internal memory storage. The last chapter of Part 4 evaluates GPPS on a handful of standard problems.

The next large section of the book spans almost 600 pages dealing with the automatic synthesis of analog electronic circuits. Here, the objective is not to induce a computer program, but it is more of a recipe for how to obtain an electronic circuit performing a certain function. The method is inspired by *cellular encoding* as introduced by Fredric Gruau. The fitness function consists of a standard circuit simulator called SPICE—a very complex software system developed over the past thirty years at Stanford University. Using GP in this way is claimed to be the first fully automatic method for circuit design, and target designs range from simple passive filters to complex active amplifying circuits. Some of the evolved circuits even infringe on old patents! Encouraged by the success in complex domains such as this, the authors have just completed one of the largest computers in the world—a 1000 processor parallel machine, that, among other things, will be used for the induction of very complex analog circuits. (Note: the quality of the book layout is uneven. For instance, in this part, some of the circuit diagrams are printed with too thin lines making them difficult to read.)

I find chapter 57 on evolvable hardware highly interesting. Rapidly reconfigurable gate arrays are used to implement and speed up fitness evaluations. These field programmable gate arrays (FPGAs) are programmable digital circuits which can be reprogrammed on-the-fly providing one of the fastest ways to evaluate simpler fitness functions such as Boolean problems.

The remainder of the book is concerned with two areas where the authors can show human competitive performance: the discovery of cellular automata rules and the discovery of motifs for molecular biology. Overall, there is no question that this is an important book and places the spotlight on one of the peak performing and most promising candidates for the general AI prize. There is no doubt that this book belongs in the standard library of all GP researchers or practitioners. This voluminous book is a bit heterogeneous, probably stemming from the fact that it is comprised of a number of previously published papers and some new material. On the other hand is the volume important documentation of innovative work done by John Koza and his colleagues. In many places, numerous pointers to work by other researchers are given, but, in the end, I believe that the book would have a stronger case for presenting the GP state of the art by including more references to similar research by other research groups.

However, the most important and intriguing thing about this book is the provocative questions raised concerning definitions and claims of human competitive performance, "Darwinian invention", and artificial intelligence, particularly, whether we have already passed an important milestone in the history of AI automatic programming.