Chapter 2
A Quick Presentation of Evolutionary Computation

Pierre Collet
Université de Strasbourg, France

ABSTRACT

Evolutionary computation is an old field of computer science, that started in the 1960s nearly simultaneously in different parts of the world. It is an optimization technique that mimics the principles of Darwinian evolution in order to find good solutions to intractable problems faster than a random search. Artificial Evolution is only one among many stochastic optimization methods, but recently developed hardware (General Purpose Graphic Processing Units or GPGPU) gives it a tremendous edge over all the other algorithms, because its inherently parallel nature can directly benefit from the difficult to use Single Instruction Multiple Data parallel architecture of these cheap, yet very powerful cards.

INTRODUCTION AND HISTORY

The development of evolutionary algorithms almost dates back to the dark ages of computers. To put back everything in perspective, Computer Science really started when John von Neumann designed the EDVAC (Electronic Discrete Variable Automatic Computer) in 1945, but the first prototype was actually implemented in 1949 with Wilkes’ EDSAC (Electronic Delay Storage Automatic Calculator). Then, for a while, the only commercially available machines used valves and were therefore not that reliable (IBM 650 in 1953). A quantum leap was made when transistors became available around the 1960s and finally, Integrated Circuits in 1964.

By that time, evolutionary computation had about ten independent beginnings in Australia, United States and Europe, starting in 1953, traced by David Fogel’s excellent Fossil Record (Fogel, 1998): Alex Fraser had evolved binary strings using crossovers (Fraser, 1957), Friedberg had already thought of self-programming computers through mutations (Friedberg, 1958; Friedberg, Dunham, & North, 1958), and Friedman of how evolution could be digitally simulated (Friedman 1959). However, the main evolutionary trends that survived are:
A Quick Presentation of Evolutionary Computation

- Evolutionary Strategies (continuous optimization), by Rechenberg and Schwefel, best described in Rechenberg (1973) and Schwefel (1995),
- Genetic Algorithms, by Holland, later popularised by Goldberg on the US East Coast (Michigan) (Holland, 1975; Goldberg, 1989),

Evolutionary computation cannot, therefore, be seen as a recent development of computer science, or even classified as artificial intelligence, which is a different concept that can also be traced back to the mid 1950s, with John McCarthy and many others.

However, until the principles of evolutionary computation were clearly understood, these techniques necessitated a larger amount of computer power than was available until the beginning of the 1990s.

Thus, although evolutionary computation really started in the late 1960s it only came of age when computers had enough power to make it competitive with other (posterior) stochastic optimization paradigms such as simulated annealing (Kirkpatrick, 1983) or Tabu Search (Glover, 1977, 1989, 1990).

Now that the field is mature, a second drastic change is taking place with the advent of General Purpose Graphic Processing Units (GPGPUs) which are massively parallel cards developed by the billions of dollars of the gaming industry. Announced for the first quarter of 2010, NVidia’s GeForce GTX395 card based on 40nm Fermi chips should give 5 TeraFlops for less than $1000. This tremendous power is directly usable by evolutionary programs, that share the very same parallel workflow than the graphic pixel and vertex shaders for which these cards have been designed.

SHORT PRESENTATION OF THE EVOLUTIONARY COMPUTATION PARADIGM

The general idea comes from the observation that animals and plants are very well adapted to their environment. Back in 1859, Charles Darwin came with an explanation for this called natural selection, that is now widely accepted (Darwin, 1859) and shown at work in a wonderful recent book by Neil Shubin (2008): Your Inner Fish. The rationale is that individuals that are not well adapted to their environment do not survive long enough to reproduce, or have less chances to reproduce than other individuals of the same species that have acquired beneficial traits through variations during the reproduction stage. Adaptation to the environment is also called fitness.

Artificial evolution grossly copies these natural mechanisms in order to optimise solutions to difficult problems. All optimisation techniques based on Darwinian principles are de facto members of the evolutionary computation paradigm, even though a distinction must be made between two different kinds of algorithms. “Standard” evolutionary algorithms evolve a fixed string of bits or reals that is passed to an evaluation function that assesses the “fitness” of the individual, while genetic programming evolves individuals that are evaluated on a number of fitness cases. This distinction may seem tenuous, but one will see in the end of this chapter that that it is important.

A UNIFIED EVOLUTIONARY ALGORITHM

Kenneth DeJong has been giving a GECCO tutorial on the unification of evolutionary algorithms for several years now and has come up with a recent book on the subject (DeJong 2005). Indeed, the previously quoted trends (Evolutionary Strategies, Genetic Algorithms, Evolutionary Programming,
Related Content

Set Comparison in Relational Query Languages
www.igi-global.com/chapter/set-comparison-relational-query-languages/11215?camid=4v1a

Evaluation of Component-Based Development Methods
www.igi-global.com/chapter/evaluation-component-based-development-methods/23021?camid=4v1a

A Framework for Analyzing Mobile Transaction Models
www.igi-global.com/article/framework-analyzing-mobile-transaction-models/3267?camid=4v1a

Case Base Management Systems: Providing Database Support to Case-based Reasoners
www.igi-global.com/article/case-base-management-systems/51133?camid=4v1a