Chapter IV
Evolutionary Algorithms

Pierre Collet
Université du Littoral Côte d'Opale, France

ABSTRACT

Evolutionary computation is an old field of computer science that started in the end of the 1960s nearly simultaneously in different parts of the world. Each paradigm has evolved separately, apparently without knowledge of what was happening elsewhere, until people finally got together and shared their experience. This resulted in strong trends that still survive, even though it is now possible to outline a generic structure for an evolutionary algorithm that is described in this chapter.

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, the 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 as follows:

- **Evolutionary Strategies:** By Rechenberg and Schwefel, best described in Rechenberg (1973) and Schwefel (1995).

Copyright © 2007, Idea Group Inc., distributing in print or electronic forms without written permission of IGI is prohibited.

- **Genetic Algorithms:** By Holland, later popularized by Goldberg on the U.S. East Coast (Michigan) (Holland, 1975; Goldberg, 1989).
- **Evolutionary Programming:** By Lawrence Fogel and later David Fogel on the U.S. West Coast (Fogel, Owens, & Walsh, 1966; Fogel, 1992).
- **Genetic Programming:** By Cramer (1985) and later developed by Koza (1992) (see Chapter V).

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 also started back in 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 them a technique competitive with other (posterior) stochastic optimization paradigms such as simulated annealing (Kirkpatrick, Gellat, & Vecchi, 1983) or tabu search (Glover, 1977, 1989, 1990) (see Chapter III).

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*, which is now widely accepted (Darwin, 1859). 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 *variation* during *reproduction*. Adaptation to the environment is also called *fitness*.

Artificial evolution grossly copies these natural mechanisms in order to optimize solutions to difficult problems. All optimization techniques based on Darwinian principles are *de facto* members of the evolutionary computation paradigm.

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 currents (evolutionary strategies, genetic algorithms, evolutionary programming, genetic programming) all share the same principles copied from natural selection.

Rather than describing each algorithm, this chapter will describe a generic and complete version that can emulate virtually any paradigm, depending on chosen parameters.

Representation of Individuals

Due to the similarities between artificial evolution and natural evolution that was the source of its inspiration, a good part of the vocabulary was borrowed from biologists. In artificial evolution, a potential solution to a problem is called an *individual*.

Using a correct representation to implement individuals is a very essential step that is trivial for some kinds of problems and much less trivial for others. The American trend (genetic algorithms) advocates using a representation that is as generic as possible—for example, a bit string

12 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/evolutionary-algorithms/21119

Related Content

Stellar Mass Black Hole for Engineering Optimization
Premalatha Kandhasamy, Balamurugan Rand Kannimuthu S (2017). *Recent Developments in Intelligent Nature-Inspired Computing (pp. 62-90).*
www.irma-international.org/chapter/stellar-mass-black-hole-for-engineering-optimization/179363

Comprehensive Review on Deep Learning for Neuronal Disorders: Applications of Deep Learning
Vinayak Majhi, Angana Saikia, Amitava Datta, Aseem Sinhaand Sudip Paul (2020). *International Journal of Natural Computing Research (pp. 27-44).*
www.irma-international.org/article/comprehensive-review-on-deep-learning-for-neuronal-disorders/241940

A Model for a Heterogeneous Genetic Network
Ângela T.F. Gonçalvesand Ernesto J.F. Costa (2010). *Handbook of Research on Computational Methodologies in Gene Regulatory Networks (pp. 523-545).*
www.irma-international.org/chapter/model-heterogeneous-genetic-network/38250

Identification of Essential Proteins by Detecting Topological and Functional Clusters in Protein Interaction Network of Saccharomyces Cerevisiae
Kaustav Sengupta, Sovan Saha, Piyali Chatterjee, Mahantapas Kundu, Mita Nasipuriand Subhadip Basu (2019). *International Journal of Natural Computing Research (pp. 31-51).*
www.irma-international.org/article/identification-of-essential-proteins-by-detecting-topological-and-functional-clusters-in-protein-interaction-network-of-saccharomyces-cerevisiae/219800

Composition of Local Normal Coordinates and Polyhedral Geometry in Riemannian Manifold Learning
Gastão F. Miranda Jr., Gilson Giraldi, Carlos E. Thomazand Daniel Millàn (2015). *International Journal of Natural Computing Research (pp. 37-68).*
www.irma-international.org/article/composition-of-local-normal-coordinates-and-polyhedral-geometry-in-riemannian-manifold-learning/126482