2 Introduction to Evolutionary Computing

'*Owing to this struggle for life, variations, however slight and from whatever cause proceeding, if they be in any degree profitable to the individuals of a species, in their infinitely complex relations to other organic beings and to their physical conditions of life, will tend to the preservation of such individuals, and will generally be inherited by the offspring. The offspring, also, will thus have a better chance of surviving, for, of the many individuals of any species which are periodically born, but a small number can survive. I have called this principle, by which each slight variation, if useful, is preserved, by the term Natural Selection*'. (Darwin, 1859 [127], p. 115)

Biological evolution performs as a powerful problem-solver that attempts to produce solutions that are at least *good enough* to perform the job of survival in the current environmental context. Since Charles Darwin popularised the theory of Natural Selection, the driving force behind evolution, molecular biology has unravelled some of the mysteries of the components that underpinned earlier evolutionary ideas. In the twentieth century molecular biologists uncovered the existence of DNA, its importance in determining hereditary traits and later its structure, unlocking the key to the genetic code. The accumulation of knowledge about the biological process of evolution, often referred to as neo-Darwinism, has in turn given inspiration to the design of a family of computational algorithms known collectively as *evolutionary computation*. These evolutionary algorithms take their cues from the biological concepts of natural selection and the fact that the heritable traits are physically encoded on DNA, and can undergo variation through a series of genetic operators such as mutation and crossover.

A. Brabazon et al., *Natural Computing Algorithms*, Natural Computing Series,
DOI 10.1007/978-3-662-43631-8_2

2.1 Evolutionary Algorithms

Evolutionary processes represent an archetype, whose application transcends their biological root. Evolutionary processes can be distinguished by means of their four key characteristics, which are [57, 92]:

i. a population of entities,
ii. mechanisms for selection,
iii. retention of fit forms, and
iv. the generation of variety.

In biological evolution, species are positively or negatively selected depending on their relative success in surviving and reproducing in the environment. Differential survival, and variety generation during reproduction, provide the engine for evolution [127, 589] (Fig. 2.1).

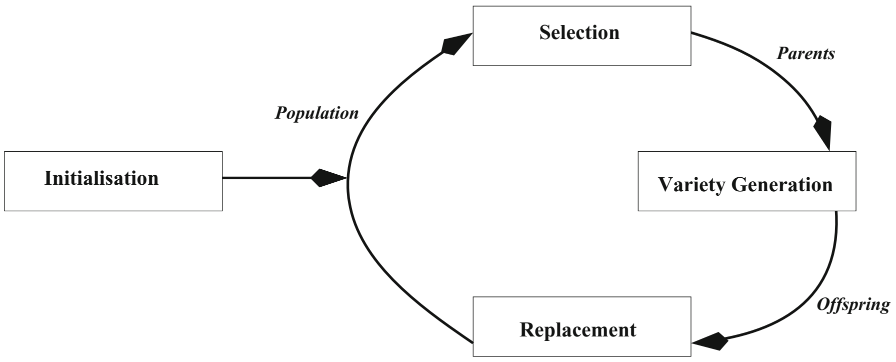

Fig. 2.1. Evolutionary cycle

These concepts have metaphorically inspired the field of evolutionary computation (EC). Algorithm 2.1 outlines the evolutionary meta-algorithm. There are many ways of operationalising each of the steps in this meta-algorithm; consequently, there are many different, but related, evolutionary algorithms. Just as in biological evolution, the selection step is a pivotal driver of the algorithm's workings. The selection step is biased in order to preferentially select better (or 'more fit') members of the current population. The generation of new individuals creates *offspring* or *children* which bear some similarity to their parents but are not identical to them. Hence, each individual represents a trial solution in the environment, with better individuals having increased chance of influencing the composition of individuals in future generations. This process can be considered as a 'search' process, where the objective is to continually improve the quality of individuals in the population.

Algorithm 2.1: Evolutionary Algorithm

```
Initialise the population of candidate solutions;
repeat
    Select individuals (parents) for breeding from the current population;
    Generate new individuals (offspring) from these parents;
    Replace some or all of the current population with the newly generated
    individuals;
until terminating condition;
```

Evolutionary Computation in Computer Science

The idea of a (computer) simulated evolutionary process dates back to the very dawn of digital computing, being introduced in the writings of Alan Turing in 1948–1950 [636, 637]. One of the earliest published works on the implementation of an evolutionary-like algorithm was by Friedberg in 1958 [203] and followed by [171, 204]. In the earlier of these studies, random and routine changes were made to binary strings representing machine code, with the performance of individual instructions being monitored in a credit assignment form of learning. Friedberg's studies were subsequently compiled into an edited collection providing a snapshot of the seminal papers that gave rise to the field of Evolutionary Computation [192]. Friedberg's work represents the origin of what is now known as Genetic Programming, which was later popularised via the work of Cramer [119], Dickmans et al. [157] and Koza [339] in the 1980s.

During the 1960s and 1970s two significant, independent, lines of research developing evolutionary algorithms were undertaken in Europe and the US. The Europeans (Rechenberg and Schwefel) developed Evolution Strategies [530, 531, 560, 561] and the Americans (Fogel et al. and Holland) developed Evolutionary Programming [194] and Genetic Algorithms [281]. More recently, Storn and Price have added Differential Evolution [600] to the family of evolutionary algorithms.

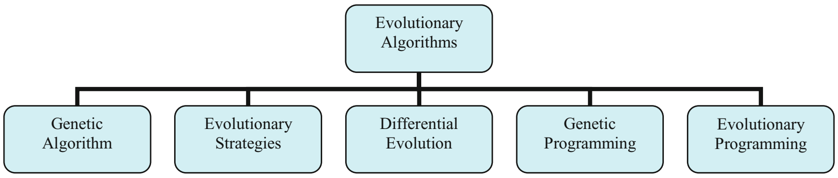

Fig. 2.2. Main branches of evolutionary computation

Although the above strands of research were initially distinct, with each having their own proponents, the lines between all of these evolutionary inspired approaches is becoming blurred with representations and strategies being used interchangeably between the various algorithms. As such, today it is common to use the term evolutionary algorithm to encompass all of the above approaches [144, 175]. Figure 2.2 provides a taxonomy of the more common branches of evolutionary computation.

In the following chapters in Part I of the book, we will introduce three of the main families of evolutionary inspired algorithms in turn, namely, the genetic algorithm (Chaps. 3 and 4), evolutionary strategies (Chap. 5), and differential evolution (Chap. 6). Following these, genetic programming is then discussed in Chap. 7.