2

Evolutionary Computing: The Origins

This chapter provides the reader with the basics for studying evolutionary computing (EC) through this book. We begin with a brief history of the field of evolutionary computing, followed by an introduction to some of the biological processes that have served as inspiration and that have provided a rich source of ideas and metaphors to researchers. We then discuss motivations for working with, and studying, evolutionary computing methods. We end with examples of applications where EC was successfully applied.

2.1 The Main Evolutionary Computing Metaphor

Evolutionary computing is a research area within computer science. As the name suggests, it is a special flavour of computing, which draws inspiration from the process of natural evolution. It is not surprising that some computer scientists have chosen natural evolution as a source of inspiration: the power of evolution in nature is evident in the diverse species that make up our world, each tailored to survive well in its own niche. The fundamental metaphor of evolutionary computing relates this powerful natural evolution to a particular style of problem solving – that of trial-and-error.

Descriptions of relevant fragments of evolutionary theory and genetics are given later on. For the time being let us consider natural evolution simply as follows. A given environment is filled with a population of individuals that strive for survival and reproduction. The fitness of these individuals is determined by the environment, and relates to how well they succeed in achieving their goals. In other words, it represents their chances of survival and of multiplying. Meanwhile, in the context of a stochastic trial-and-error (also known as generate-and-test) style problem solving process, we have a collection of candidate solutions. Their quality (that is, how well they solve the problem) determines the chance that they will be kept and used as seeds for constructing further candidate solutions. The analogies between these two scenarios are shown in Table 2.1.

A.E. Eiben, J.E. Smith, *Introduction to Evolutionary Computing*,
Natural Computing Series, DOI 10.1007/978-3-662-44874-8_2

Evolution		Problem solving
Environment	$\longleftrightarrow$	Problem
Individual	$\longleftrightarrow$	Candidate solution
Fitness	$\longleftrightarrow$	Quality

Table 2.1. The basic evolutionary computing metaphor linking natural evolution to problem solving

2.2 Brief History

Surprisingly enough, this idea of applying Darwinian principles to automated problem solving dates back to the 1940s, long before the breakthrough of computers [167]. As early as 1948, Turing proposed "genetical or evolutionary search", and by 1962 Bremermann had actually executed computer experiments on "optimization through evolution and recombination". During the 1960s three different implementations of the basic idea were developed in different places. In the USA, Fogel, Owens, and Walsh introduced **evolutionary programming** [173, 174], while Holland called his method a **genetic algorithm** [102, 218, 220]. Meanwhile, in Germany, Rechenberg and Schwefel invented **evolution strategies** [352, 373]. For about 15 years these areas developed separately; but since the early 1990s they have been viewed as different representatives ('dialects') of one technology that has come to be known as **evolutionary computing** (EC) [22, 27, 28, 137, 295, 146, 104, 12]. In the early 1990s a fourth stream following the general ideas emerged, **genetic programming**, championed by Koza [37, 252, 253]. The contemporary terminology denotes the whole field by evolutionary computing, the algorithms involved are termed **evolutionary algorithms**, and it considers evolutionary programming, evolution strategies, genetic algorithms, and genetic programming as subareas belonging to the corresponding algorithm variants.

The development of scientific forums devoted to EC gives an indication of the field's past and present, and is sketched in Fig. 2.1. The first international conference specialising in the subject was the *International Conference on Genetic Algorithms* (ICGA), first held in 1985 and repeated every second year until 1997. In 1999 it merged with the *Annual Conference on Genetic Programming* to become the annual *Genetic and Evolutionary Computation Conference* (GECCO). At the same time, in 1999, the *Annual Conference on Evolutionary Programming*, held since 1992, merged with the *IEEE Conference on Evolutionary Computation*, held since 1994, to form the *Congress on Evolutionary Computation* (CEC), which has been held annually ever since.

The first European event (explicitly set up to embrace all streams) was *Parallel Problem Solving from Nature* (PPSN) in 1990, which has became a biennial conference. The first scientific journal devoted to this field, *Evolutionary Computation*, was launched in 1993. In 1997 the European Commission decided to fund a European research network in EC, called EvoNet, whose

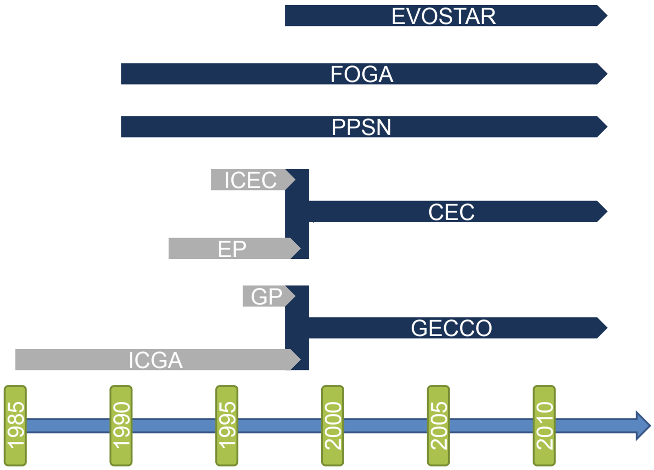

Fig. 2.1. Brief sketch of the EC conference history

funds were guaranteed until 2003. At the time of writing (2014), there were three major EC conferences (CEC, GECCO, and PPSN) and many smaller ones, including one dedicated exclusively to theoretical analysis and development, *Foundations of Genetic Algorithms* (FOGA), held biennially since 1990, and a European event seeded by EvoNet, the annual EVOSTAR conference. There are now various scientific EC journals (*Evolutionary Computation*, *IEEE Transactions on Evolutionary Computation*, *Genetic Programming and Evolvable Machines*, *Evolutionary Intelligence*, *Swarm and Evolutionary Computing*) and many with a closely related profile, e.g., on natural computing, soft computing, or computational intelligence. We estimate the number of EC publications in 2014 at somewhere over 2000 – many of them in journals and conference proceedings of specific application areas.

2.3 The Inspiration from Biology

2.3.1 Darwinian Evolution

Darwin's theory of evolution [92] offers an explanation of the origins of biological diversity and its underlying mechanisms. In what is sometimes called the macroscopic view of evolution, natural selection plays a central role. Given an environment that can host only a limited number of individuals, and the basic instinct of individuals to reproduce, selection becomes inevitable if the population size is not to grow exponentially. Natural selection favours those individuals that compete for the given resources most effectively, in other words, those that are adapted or fit to the environmental conditions best. This phenomenon is also known as **survival of the fittest**.[1] Competition-based selection is one of the two cornerstones of evolutionary progress. The other primary force identified by Darwin results from phenotypic variations among members of the population. Phenotypic traits (see also Sect. 2.3.2) are those behavioural and physical features of an individual that directly affect its response to the environment (including other individuals), thus determining its fitness. Each individual represents a unique combination of phenotypic traits that is evaluated by the environment. If this combination evaluates favourably, then the individual has a higher chance of creating offspring; otherwise the individual is discarded by dying without offspring. Importantly, if they are heritable (and not all traits are), favourable phenotypic traits may be propagated via the individual's offspring. Darwin's insight was that small, random variations – mutations – in phenotypic traits occur during reproduction from generation to generation. Through these variations, new combinations of traits occur and get evaluated. The best ones survive and reproduce, and so evolution progresses. To summarise this basic model, a population consists of a number of individuals. These individuals are the units of selection, that is to say that their reproductive success depends on how well they are adapted to their environment relative to the rest of the population. As the more successful individuals reproduce, occasional mutations give rise to new individuals to be tested. Thus, as time passes, there is a change in the constitution of the population, i.e., the population is the unit of evolution.

This process is well captured by the intuitive metaphor of an **adaptive landscape** or adaptive surface [468]. On this landscape the height dimension belongs to fitness: high altitude stands for high fitness. The other two (or more, in the general case) dimensions correspond to biological traits as shown in Fig. 2.2. The xy-plane holds all possible trait combinations, and the z-values show their fitnesses. Hence, each peak represents a range of successful trait combinations, while troughs belong to less fit combinations. A given population can be plotted as a set of points on this landscape, where each dot is one individual realising a possible trait combination. Evolution is then the process of gradual advances of the population to high-altitude areas, powered by variation and natural selection. Our familiarity with the physical landscape on which we exist naturally leads us to the concept of **multimodal problems**. These are problems in which there are a number of points that are better than all their neighbouring solutions. We call each of these points a **local optimum** and denote the highest of these as the **global optimum**. A problem in which there is only one local optimum is known as **unimodal**.

The link with an optimisation process is as straightforward as it is misleading, because evolution is not a unidirectional uphill process [103]. Because the

[1] This term is actually rather misleading. It is often, and incorrectly, taken to mean that the best fit individual *always* survives. Since nature, and EC by design, contains a lot of randomness, this does not always happen.

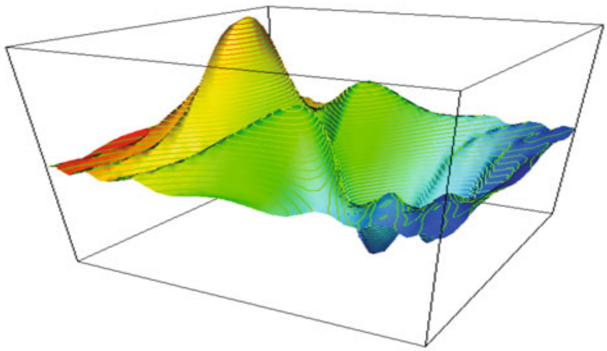

Fig. 2.2. Illustration of Wright's adaptive landscape with two traits

population has a finite size, and random choices are made in the selection and variation operators, it is common to observe the phenomenon of **genetic drift**, whereby highly fit individuals may be lost from the population, or the population may suffer from a loss of variety concerning some traits. This can have the effect that populations 'melt down' the hill, and enter low-fitness valleys. The combined global effects of drift and selection enable populations to move uphill as well as downhill, and of course there is no guarantee that the population will climb back up the same hill. Escaping from locally optimal regions is hereby possible, and according to Wright's shifting balance theory the maximum of a fixed landscape can be reached.

2.3.2 Genetics

The microscopic view of natural evolution is offered by molecular genetics. It sheds light on the processes below the level of visible phenotypic features, in particular relating to heredity. The fundamental observation from genetics is that each individual is a dual entity: its phenotypic properties (outside) are represented at a genotypic level (inside). In other words, an individual's **genotype** encodes its **phenotype**. **Genes** are the functional units of inheritance encoding phenotypic characteristics. For instance, visible properties like the fur colour or tail length could be determined by genes. Here it is important to distinguish genes and **alleles**. An allele is one of the possible values that a gene can have – so its relationship to a gene is just like that of a specific value to a variable in mathematics. To illustrate this by an oversimplified example, bears could have a gene that determines fur colour, and for a polar bear we would expect to see the allele that specifies the colour white. In natural systems the genetic encoding is not one-to-one: one gene might affect more phenotypic traits (**pleitropy**), and in turn one phenotypic trait can be determined by more than one gene (**polygeny**). Phenotypic variations are always caused by genotypic variations, which in turn are the consequences of mutations of genes, or recombination of genes by sexual reproduction.

Another way to think of this is that the genotype contains all the information necessary to build the particular phenotype. The term **genome** stands for the complete genetic information of a living being containing its total building plan. This genetic material, that is, all genes of an organism, is arranged in several chromosomes; there are 46 in humans. Higher life forms (many plants and animals) contain a double complement of chromosomes in most of their cells, and such cells – and the host organisms – are called **diploid**. Thus the chromosomes in human diploid cells are arranged into 23 pairs. **Gametes** (i.e., sperm and egg cells) contain only one single complement of chromosomes and are called **haploid**. The combination of paternal and maternal features in the offspring of diploid organisms is a consequence of fertilisation by a fusion of such gametes: the haploid sperm cell merges with the haploid egg cell and forms a diploid cell, the zygote. In the zygote, each chromosome pair is formed by a paternal and a maternal half. The new organism develops from this zygote by the process named **ontogenesis**, which does not change the genetic information of the cells. Consequently, all body cells of a diploid organism contain the same genetic information as the zygote it originates from.

In evolutionary computing, the combination of features from two individuals in offspring is often called crossover. It is important to note that this is not analogous to the working of diploid organisms, where **crossing-over** is not a process during mating and fertilisation, but rather happens during the formation of gametes, a process called meiosis. **Meiosis** is a special type of cell division that ensures that gametes contain only one copy of each chromosome. As said above, a diploid body cell contains chromosome pairs, where one half of the pair is identical to the paternal chromosome from the sperm cell, and the other half is identical to the maternal chromosome from the egg cell. During meiosis a chromosome pair first aligns physically, that is, the copies of the paternal and maternal chromosomes, which form the pair, move together and stick to each other at a special position (the centromere, not indicated, see Fig. 2.3, left). In the second step the chromosomes double so that four strands (called chromatids) are aligned (Fig. 2.3, middle). The actual crossing-over takes place between the two inner strands that break at a random point and exchange parts (Fig. 2.3, right). The result is four differ-

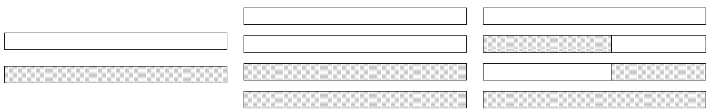

Fig. 2.3. Three steps in the (simplified) meiosis procedure regarding one chromosome

ent copies of the chromosome in question, of which two are identical to the original parental chromosomes, and two are new recombinations of paternal and maternal material. This provides enough genetic material to form four haploid gametes, which is done via a random arrangement of one copy of each chromosome. Thus in the newly created gametes the genome is composed of chromosomes that are either identical to one of the parent chromosomes, or recombinants. The resulting four haploid gametes are usually different from both original parent genomes, facilitating genotypic variation in offspring.

In the 19th century Mendel first investigated and understood heredity in diploid organisms. Modern genetics has added many details to his early picture, but we are still very far from understanding the whole genetic process. What we do know is that all life on Earth is based on DNA – the famous double helix of nucleotides encoding the whole organism be it a plant, animal, or *Homo sapiens*. Triplets of nucleotides form so-called codons, each of which codes for a specific amino acid. The genetic code (the translation table from the $4^3 = 64$ possible codons to the 20 amino acids from which proteins are created) is universal, that is, it is the same for all life on Earth. This fact is generally acknowledged as strong evidence that the whole biosphere has the same origin. Genes are larger structures on the DNA, containing many codons, carrying the code of proteins. The path from DNA to protein consists of two main steps: transcription, where information from the DNA is written to RNA, and translation, the step from RNA to protein. It is one of the principal dogmas of molecular genetics that this information flow is only one-way. Speaking in terms of genotypes and phenotypes, this means that phenotypic features cannot influence genotypic information. This refutes earlier theories (for instance, that of Lamarck), which asserted that features acquired during an individual's lifetime could be passed on to its offspring via inheritance. A consequence of this view is that changes in the genetic material of a population can only arise from random variations and natural selection and definitely not from individual learning. It is important to understand that all variations (mutation and recombination) happen at the genotypic level, while selection is based on actual performance in a given environment, that is, at the phenotypic level.

2.3.3 Putting It Together

The Darwinian theory of evolution and the insights from genetics can be put together to clarify the dynamics behind the emergence of life on Earth. For the purposes of this book a simplified picture is sufficient. The main points are then the following. Any living being is a dual entity with an invisible code (its genotype) and observable traits (its phenotype). Its success in surviving and reproducing is determined by its phenotypical properties, e.g., good ears, strong muscles, white fur, friendly social attitude, attractive scent, etc. In other words, the forces known as natural selection and sexual selection act on the phenotype level. Obviously, selection also affects the genotype level, albeit implicitly. The key here is reproduction. New individuals may have one single parent (**asexual reproduction**) or two parents (**sexual reproduction**). In either case, the genome of the new individual is not identical to that of the parent(s), because of small reproductive variations and because the combination of two parents will differ from both. In this way genotype variations are created, which in turn translate to phenotype variations[2] and thus are subject to selection. Hence, at a second level, genes are also subject to the game of survival and reproduction, and some evolutionary biologists would argue that viewing evolution from the perspective of genes is more productive – so that rather than thinking about populations of individuals, we should think about a 'gene pool' containing genes which compete and replicate over time, being evaluated as they reoccur in different individuals [100].

Elevating this process to an abstract level, we can perceive each newborn individual as a new sample in the space of all possible living things. This new sample is produced by forces of variation, i.e., asexual or sexual reproduction, and it is evaluated by the forces of selection. It needs to pass two hurdles: first proving viable to live on its own, then proving capable of reproducing. In species using sexual reproduction, this implies an extra test of being able to find a mate (sexual selection). This cycle of production and evaluation may sound familiar to readers with an algorithmic background, such procedures are known as generate-and-test methods.

2.4 Evolutionary Computing: Why?

Developing automated problem solvers (that is, algorithms) is one of the central themes of mathematics and computer science. Just as engineers have always looked at Nature's solutions for inspiration, copying 'natural problem solvers' is a stream within these disciplines. When looking for the most powerful natural problem solver, there are two rather obvious candidates:

- the human brain (that created "the wheel, New York, wars and so on" [4, Chap. 23]);
- the evolutionary process (that created the human brain).

Trying to design problem solvers based on the first candidate leads to the field of neurocomputing. The second option forms a basis for evolutionary computing.

Another motivation can be identified from a technical perspective. Computerisation in the second half of the 20th century created a growing demand for problem-solving automation. The growth rate of the research and development capacity has not kept pace with these needs. Hence, the time available for thorough problem analysis and tailored algorithm design has been decreasing. A parallel trend has been the increase in the complexity of problems to

[2] Some genotype variations may not cause observable phenotype differences, but this is not relevant for the present argument.

be solved. These two trends imply an urgent need for robust algorithms with satisfactory performance. That is, there is a need for algorithms that are applicable to a wide range of problems, do not need much tailoring for specific problems, and deliver good (not necessarily optimal) solutions within acceptable time. Evolutionary algorithms do all this, and so provide an answer to the challenge of deploying automated solution methods for more and more problems, which are ever more complex, in less and less time.

A third motivation is one that can be found behind every science: human curiosity. Evolutionary processes are the subjects of scientific studies where the main objective is to understand how evolution works. From this perspective, evolutionary computing represents the possibility of performing experiments differently from traditional biology. Evolutionary processes can be simulated in a computer, where millions of generations can be executed in a matter of hours or days and repeated under various circumstances. These possibilities go far beyond studies based on excavations and fossils, or those possible *in vivo*. Naturally, the interpretation of such simulation experiments must be done very carefully. First, because we do not know whether the computer models represent the biological reality with sufficient fidelity. Second, it is unclear whether conclusions drawn in a digital medium, *in silico*, can be transferred to the carbon-based biological medium. Despite these caveats there is a strong tradition within evolutionary computing to 'play around' with evolution for the sake of understanding how it works. Application issues do not play a role here, at least not in the short term. But, of course, learning more about evolutionary processes in general can help in designing better algorithms later.

Having given three rather different reasons why people might want to use evolutionary computation, we next illustrate the power of evolutionary problem solving by a number of application examples from various areas.

A challenging optimisation task that has successfully been carried out by evolutionary algorithms is the timetabling of university classes [74, 329]. Typically, some 2000–5000 events take place during a university week, and these must each be given a day, time, and room. The first optimisation task is to reduce the number of clashes, for example, a student needing to be in two places at once, or a room being used for two lectures at the same time. Producing feasible timetables with no clashes is a hard task. In fact, it turns out that in most cases the vast majority of the space of all timetables is filled with infeasible solutions. In addition to producing feasible timetables, we also want to produce timetables that are optimised as far as the users are concerned. This optimisation task involves considering a large number of objectives that compete with each other. For example, students may wish to have no more than two classes in a row, while their lecturers may be more concerned with having whole days free for conducting research. Meanwhile, the main goal of the university management might be to make room utilisation more efficient, or to cut down the amount of movement around or between the buildings.

EC applications in industrial design optimisation can be illustrated with the case of a satellite dish holder boom. This ladder-like construction connects the satellite's body with the dish needed for communication. It is essential that this boom is stable, in particular vibration resistant, as there is no air in space that would damp vibrations that could break the whole construction. Keane et al. [245] optimised this construction using an evolutionary algorithm. The resulting structure is 20,000% (!) better than traditional shapes, but for humans it looks very strange: it exhibits no symmetry, and there is no intuitive design logic visible (Fig. 2.4). The final design looks pretty much

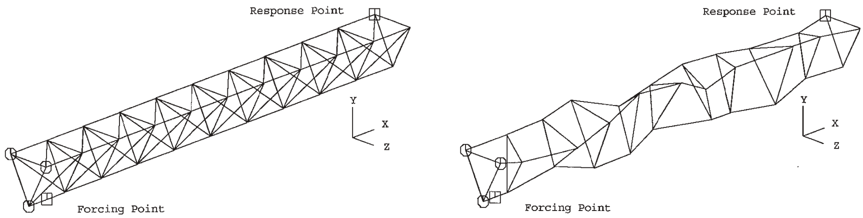

Fig. 2.4. The initial, regular design of the 3D boom (*left*) and the final design found by a genetic algorithm (*right*)

like a random drawing, and the crucial thing is this: it *is* a random drawing, drawn without intelligence, but evolving through a number of consecutive generations of improving solutions. This illustrates the power of evolution as a designer: it is not limited by conventions, aesthetic considerations, or ungrounded preferences for symmetry. On the contrary, it is purely driven by quality, and thereby it can come to solutions that lie outside of the scope of human thinking, with its implicit and unconscious limitations. It is worth mentioning that evolutionary design often goes hand-in-hand with reverse engineering. In particular, once a provably superior solution is evolved, it can be analysed and explained through the eyes of traditional engineering. This can lead to generalisable knowledge, i.e., the formulation of new laws, theories, or design principles applicable to a variety of other problems of similar type.[3]

Modelling tasks typically occur in data-rich environments. A frequently encountered situation is the presence of many examples of a certain event or phenomenon without a formal description. For instance, a bank may have one million records (profiles) of clients containing their sociogeographical data, financial overviews of their mortgages, loans, and insurances, details of their card usage, and so forth. Certainly, the bank also has information about client

[3] In the case of the satellite dish boom, it is exactly the asymmetric character that works so well. Namely, vibrations are waves that traverse the boom along the rungs. If the rungs are of different lengths then these waves meet in a different phase and cancel each other. This small theory sounds trivial, but it took the asymmetric evolved solution to come to it.

behaviour in terms of paying back loans, for instance. In this situation it is a reasonable assumption that the profile (facts and known data from the past) is related to behaviour (future events). In order to understand the repayment phenomenon, what is needed is a model relating the profile inputs to the behavioural patterns (outputs). Such a model would have predictive power, and thus would be very useful when deciding about new loan applicants. This situation forms a typical application context for the areas of machine learning and data mining. Evolutionary computing is a possible technology that has been used to solve such problems [179].

Another example of this type of modelling approach can be seen in [370], where Schulenburg and Ross use a learning classifier system to evolve sets of rules modelling the behaviour of stock market traders. As their inputs they used ten years of trading history, in the form of daily statistics such as volume of trade, current price, change in price over the last few days, whether this price is a new high (or low), and so on for a given company's stock. The evolved traders consisted of sets of `condition`$\rightarrow$`action` rules. Each day the current stock market conditions were presented to the trader, triggering a rule that decided whether stock was bought or sold. Periodically a genetic algorithm is run on the set of (initially random) rules, so that well-performing ones are rewarded, and poorly performing ones are discarded. It was demonstrated that the system evolved trading agents that outperformed many well-known strategies, and varied according to the nature of the particular stock they were trading. Of particular interest, and benefit, compared to methods such as neural networks (which are also used for this kind of modelling problem in time-series forecasting), is the fact that the rule-bases of the evolved traders are easily examinable, that is to say that the models that are evolved are particularly transparent to the user.

Evolutionary computing can also be applied to simulation problems, that is, to answer what-if questions in a context where the investigated subject matter is evolving, i.e., driven by variation and selection. Evolutionary economics is an established research area, roughly based on the perception that the game and the players in the socioeconomic arena have much in common with the game of life. In common parlance, the survival of the fittest principle is also fundamental in the economic context. Evolving systems with a socioeconomic interpretation can differ from biological ones in that the behavioural rules governing the individuals play a very strong role in the system. The term agent-based computational economy is often used to emphasise this aspect [427]. Academic research in this direction is often based on a simple model called Sugarscape world [155]. This features agent-like inhabitants in a grid space, and a commodity (the sugar) that can be consumed, owned, traded, and so on by the inhabitants. There are many ways to set up system variants with an economics interpretation and conduct simulation experiments. For instance, Bäck et al. [31] investigate how artificially forced sugar redistribution (tax) and evolution interact under various circumstances. Clearly, interpretation of the outcomes of such experiments must be done very carefully, avoiding ungrounded claims on transferability of results into a real socioeconomic context.

Finally, we note that evolutionary computing experiments with a clear biological interpretation are also very interesting. Let us mention two approaches by way of illustration: trying existing biological features or trying nonexisting biological features. In the first approach, simulating a known natural phenomenon is a key issue. This may be motivated by an expectation that the natural trick will also work for algorithmic problem-solving, or by simply being willing to test whether the effects known in carbon would occur in silicon as well. Take incest as an example. A strong moral taboo against incest has existed for thousands of years, and for the last century or two there is also scientific insight supporting this: incest leads to degeneration of the population. The results in [158] show that computer-simulated evolution also benefits from incest prevention. This confirms that the negative effects of incest are inherent for evolutionary processes, independently from the medium in which they take place. The other approach to simulations with a biological flavour is the opposite of this: it implements a feature that does not exist in biology, but can be implemented in a computer. As an illustration, let us take multiparent reproduction, where more than two parents are required for mating, and offspring inherit genetic material from each of them. Eiben et al. [126, 128] have experimented a great deal with such mechanisms showing the beneficial effects under many different circumstances.

To summarise this necessarily brief introduction, evolutionary computing is a branch of computer science concerned with a class of algorithms that are broadly based on the Darwinian principles of natural selection, and that draw inspiration from molecular genetics. Over the history of the world, many species have arisen and evolved to suit different environments, all using the same biological machinery. In the same way, if we provide an evolutionary algorithm with a new environment we hope to see adaptation of the initial population in a way that better suits the environment. Typically (but not always) this environment will take the form of a problem to be solved, with feedback to the individuals representing how well the solutions they represent solve the problem, and we have provided some examples of this. However, as we have indicated, the search for optimal solutions to some problem is not the only use of evolutionary algorithms; their nature as flexible adaptive systems gives rise to applications varying from economic modelling and simulation to the study of diverse biological processes during adaptation.

For exercises and recommended reading for this chapter, please visit
www.evolutionarycomputation.org.