References

Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143:29–36

Argonaute-mediated Cleavage

► Target Cleavage

Arrays

► DNA Microarrays

Arrow Ontology

► Edge Ontology

Artificial Evolution

Christoph Adami
Department of Microbiology and Molecular Genetics, Michigan State University, East Lansing, MI, USA

Synonyms

Simulated evolution

Definition

Artificial evolution refers to any procedure that uses the mechanism of Darwinian evolution to generate a product. While this definition encompasses the evolution of organic life forms by means of artificial selection (breeding of animals, plants, or microorganisms), artificial evolution commonly refers to the instantiation of evolution within a nonbiological medium. Artificial evolution is sometimes used synonymously with ► artificial life, but in principle the concepts are overlapping but different.

Characteristics

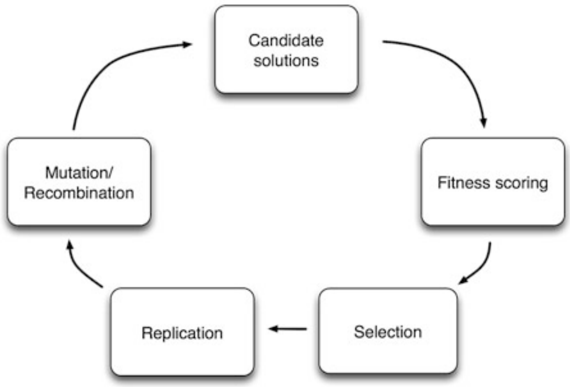
In Computer Science

Within computer science, artificial evolution is commonly implemented in terms of an evolutionary or genetic algorithm (► Genetic Algorithms, ► Evolution Programming) (GA). A GA is a search procedure that implements the three essential elements of Darwinian evolution - replication, mutation, and selection. Typically, the procedure acts on a population of candidate solutions, which are scored according to a fitness criterion and selected to be replicated according to a formula that takes the fitness score into account (see Fig. 1).

This formula may be as simple as selecting a percentage of the highest fitness individuals (elite selection) or be a probability that is proportional to fitness (roulette wheel selection) or a mixture of those. The fitness function is usually determined by the user so that the maximum of that function is achieved for the optimal solution. In cases where it is not known which function is maximized by the optimal solution to a problem, the construction of an appropriate fitness function can be a difficult research problem. After being selected for replication, individuals are mutated and recombined with a given rate to create new candidate solutions that have inherited the features of their parents but potentially carry new features not previously present in the population. Typically, GAs use a variety of mutation mechanisms to create variation. Evolutionary algorithms are alternatives to more traditional random or heuristic search algorithms and work best in large search spaces where the best solution consists of partial solutions that are also fit.

In Engineering

Artificial evolution can be used to solve engineering problems by applying evolutionary computation techniques to hardware rather than software. For example, it is possible to design electronic circuits in reconfigurable hardware (Thompson 1996) that have novel properties that exploit the material properties of the substrate. In another example of unconventional computing (► Unconventional Computation), a team



Artificial Evolution, Fig.1 Typical flow diagram of events in a genetic algorithm

Generation 01: WDLTMNLT DTJBKWIRZREZLMQCO P
Generation 02: WDLTMNLT DTJBSWIRZREZLMQCO P
Generation 10: MDLDMNLS ITJISWHRZREZ MECS P
Generation 20: MELDINLS IT ISWPRKE Z WECSEL
Generation 30: METHINGS IT ISWLIKE B WECSEL
Generation 40: METHINKS IT IS LIKE I WEASEL
Generation 43: METHINKS IT IS LIKE A WEASEL

Artificial Evolution, Fig. 2 Selected generations on the line of descent obtained by running Dawkins' "weasel" program

evolved a circuit with the goal of producing an oscillatory signal. One of the resulting circuits evolved a radio receiver that captured the clock signal from a computer on a nearby desk (Bird and Layzell 2002).

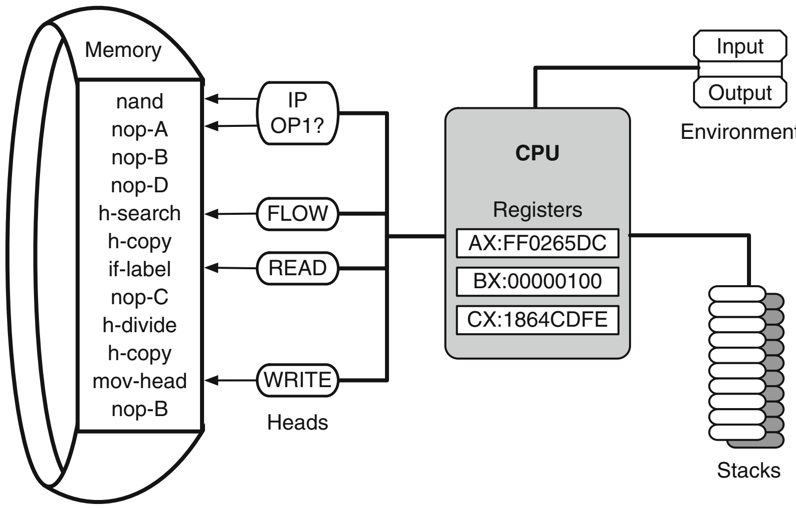
In Evolutionary Biology

Artificial evolution is used in evolutionary biology to illustrate the process of evolution itself by instantiating evolution algorithmically, usually within a computer (Adami 1998). One of the earliest uses of computers to study the process of evolution is Richard Dawkins' thought experiment that demonstrates the process of evolution in an artificial medium by evolving the target phrase "Methinks it is like a weasel" from a randomly generated sequence of letters drawn from an alphabet of 28 (Dawkins 1986). In this example, sequences are copied 100 times (instantiating heredity), while 1 in 20 characters is changed (instantiating the process of mutation). The resulting sequences are compared to the target phrase, and the sequence that is closest to the target is chosen to be copied again (instantiating selection), see Fig. 2.

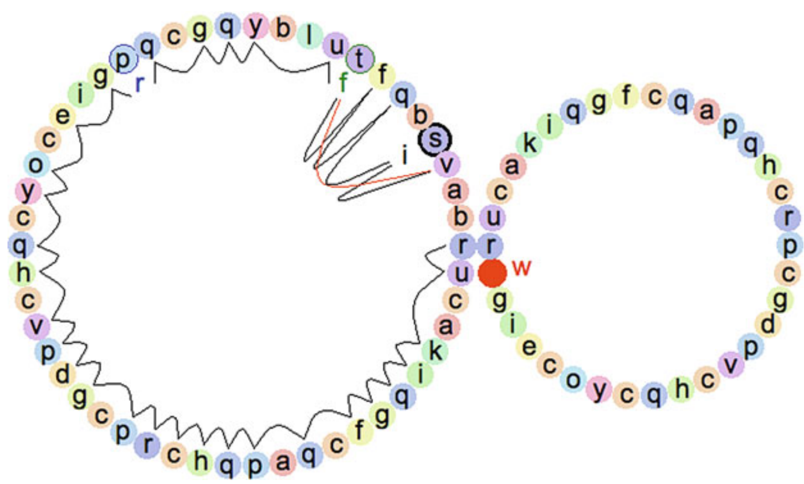
In the same book, Dawkins introduces the "biomorph" program that evolves tree-like structures whose appearance is determined by nine developmental "genes" that determine the tree's appearance. The genes that encode the tree are mutated and recombined at random and selected by the user that, thus, guides the process of evolution to generate desired shapes. Another well-known example of artificial evolution that probes the coevolution of morphology and behavior is due to Karl Sims (Sims 1994), who studied how creatures built from interconnected blocks and controlled by a neural architecture evolve in a three-dimensional world with simulated physics. This work highlighted the importance of complex environments in the evolution of complexity.

Digital Life

Computer viruses (or, more generally, malware) evolve by artificial means when the malware creators react to a changed fitness landscape (e.g., new security countermeasures) by adapting their virus to these changes. In this instantiation of artificial evolution, replication is usually a key feature of the malware program, while the fitness function is implicit to the environment within which the virus seeks to replicate, rather than specified externally. Variation is usually directed by the malicious users but can sometimes be autonomous. The concept of an evolving computer program as an instantiation of evolution (as opposed to a simulation of evolution) gave rise to the field of digital life, where self-replicating computer programs are executed on virtual (simulated) central processing units (CPUs). By encasing programs in a virtual world, it is possible to design the language they are written in (their genetic code), to control the program's rate of mutation, and the complexity of the world they live in. The first working digital life system called "tierra" was created by Tom Ray (Ray 1992). In tierra, self-replicating computer programs live in simulated core memory and compete for CPU time and memory space. Because the ► fitness of any particular ► digital organism in tierra is solely determined by its ability to contribute offspring to future generations, there is no a priori optimal program. The digital life system "Avida" (see Figs. 3 and 4) was developed at the California Institute of Technology in order to study fundamental aspects of evolutionary biology, such as the evolution of complexity (Adami and Brown 1994; Ofria et al. 1998). Digital life research can provide a system's view of evolutionary biology because



Artificial Evolution, Fig. 3 The CPU acting on a self-replicating program in the Avida environment. In this version, four threads (marked "Heads") are executing the avidian code simultaneously, akin to the transcription of genetic code by four DNA polymerases



Artificial Evolution, Fig. 4 An avidian genome in the act of self-replication. Letters *a–z* stand for one of the 26 possible instructions. The *solid black line* follows the forward execution, while *red* denotes a backward jump

digital organisms are best understood within the selective environment they evolved in, which includes the population of programs itself.

Digital life research has given rise to a number of discoveries in evolutionary biology that have been validated later in biological organisms, such as the "survival-of-the-flattest" effect or the importance of ▶ epistasis in the evolution of ▶ complexity (Adami 2006). This type of work illustrates the idea that the difference between evolution in the biochemical realm and evolution in the computational domain lies only in the substrate that carries the information that evolves, but that the processes that underlie the dynamics (the algorithmic rules) are the same. Because of this generality, digital life systems such as Avida have been used not only to study the process of evolution itself but can also be adapted to study the evolution of behavior and intelligence as well as software design (Beckmann et al. 2008).

Cross-References

▶ Artificial Life
▶ Complexity
▶ Digital Organism
▶ Epistasis
▶ Evolution Programming
▶ Fitness
▶ Genetic Algorithms
▶ Unconventional Computation

References

Adami C (1998) Introduction to artificial life. Springer, New York

Adami C (2006) Digital genetics: unravelling the genetic basis of evolution. Nat Rev Genet 7:109–18

Adami C, Brown CT (1994) Evolutionary learning in the 2D artificial life system "Avida". In: Brooks R, Maess P (eds) 4th international conference on the synthesis and simulation of living systems. MIT Press, Boston, pp 377–381

Beckmann BE, Grabowski LM, McKinley PK, Ofria C (2008) Applying digital evolution to the design of self-adaptive software. In: IEEE symposium on artificial life '09, IEEE Computational Intelligence Society, Nashville, TN, pp 100–107

Bird J, Layzell P (2002) The evolved radio and its implications for modelling the evolution of novel sensors. In: Evolutionary computation, 2002. CEC '02, IEEE Press, pp 1836–1841

Dawkins R (1986) The blind watchmaker. Oxford University Press, Oxford, UK

Ofria C, Brown CT, Adami C (1998) The avida user's manual. In: Adami C (ed) Introduction to artificial life. Springer, New York, pp 297–350

Ray TS (1992) An approach to the synthesis of life. In: Langton CG, Taylor C, Farmer JD, Rasmusse S (eds) Second interdisciplinary workshop on the synthesis and simulation of living systems. Addison Wesley, Santa Fe, p 371

Sims K (1994) Evolving virtual creatures. In: Glassners A (ed) Proceedings of the 21st annual conference on computer graphics and interactive techniques. ACM Press, pp 15–22

Thompson A (1996) Silicon evolution. In: Koza JR, Goldberg DE, Fogel DB, Riolo RL (eds) Genetic programming 1996: proceedings of the 1st annual conference (GP96). MIT Press, pp 444–445

Artificial Immune Systems

► Lymphocyte Dynamics and Repertoires, Modeling

Artificial Life

Jan T. Kim
School of Computing Sciences, University of East Anglia, Norwich, Norfolk, UK

Definition

Artificial Life (Langton 1992b; Bedau et al. 2000) is a highly interdisciplinary field of research which comprises areas of the biological sciences including genetics, evolutionary biology, ecology, neurobiology, and behavioral biology, as well as aspects of computer science, mathematics, engineering, and robotics. Predecessor disciplines that have informed Artificial Life also include theoretical and mathematical biology and cybernetics. In turn, Artificial Life can be one of the predecessors of Systems Biology.

The central aim of Artificial Life is to identify, understand, and use the fundamental principles that underpin and organize biological systems. Special emphasis is given to finding *simple* principles or rules that give rise to the complexity that is characteristic of biological systems, as complexity is regarded as an emergent phenomenon (► Emergence). A central methodological pattern of Artificial Life research is to study dynamics or phenomena of biological systems by synthesizing or simulating them in entirely different media. These media include software (e.g., using computational or mathematical models to study evolution), hardware (e.g., using robots to study behavior), and even wetware (i.e., building systems from scratch using techniques from chemistry and molecular biology).

The Artificial Life approach is analogous to that of Artificial Intelligence, which aims to engineer intelligent machines and comprises various techniques that are based on principles gleaned from existing natural systems (e.g., the brain). The name "Artificial Life" should be understood as a reflection of this analogy.

Interest in the synthesis of life-like systems may derive from two distinct motivations. On the one hand, bio-inspired mechanisms are adopted to engineer systems that exhibit desirable properties of living systems. Examples for such properties include robustness to unpredictable environments, learning, or other forms of adaptation. On the other hand, such systems may be built to be amenable to observation or experimentation that is infeasible with the original object of bioscientific inquiry. Computational simulations are paradigmatic of this approach, as they allow continuous, nondestructive observation of dynamical processes where such observation would be impossible for objects in molecular biology.

The biosciences typically focus on existing living systems, and understanding of biological system therefore tends to be descriptive rather than predictive. The scope of Artificial Life goes beyond the existing "life as we know it," and expressly includes "life as it could be." As an example, a traditional approach to