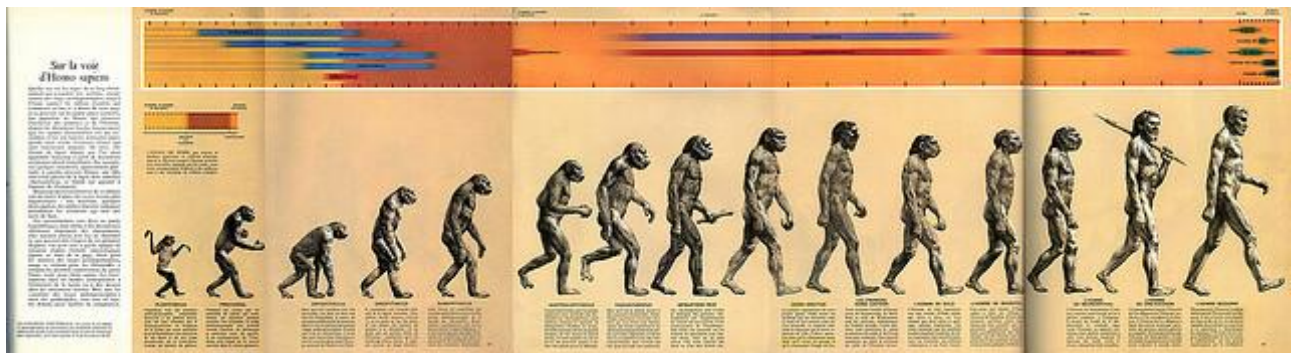


## EVOLUTIONARY COMPUTATION AND GENETIC ALGORITHMS



*Rudolph Zallinger's "The March of Progress" from Time-Life's 1965 book Early Man<sup>1</sup>*

The picture above was created by natural history painter Rudolph Zallinger and was first published in 1965 in the volume *Early Man*<sup>1</sup>, authored by anthropologist F. Clark Howell. Its depiction of evolution as the monkey who gradually transforms into a modern human being is fascinating but scientifically wrong<sup>2</sup>. The theory of evolution has its origin in the publishing of Charles Darwin's *On the Origin of Species*<sup>3</sup>. The central concept is natural selection; all living creatures are descendants of older species and the variations that can be seen in the natural world have been selected by nature in the process known as "survival of the fittest", which consists of some individuals having greater chance of reproduction due to some heritable differences. These individuals are said to have higher "fitness", which measures the "success" of an organism. The benefiting traits are essentials to survive in the competitive state of nature but also to increase the chance of reproduction (and are therefore subject to sexual selection)<sup>4,5</sup>.

There are four requirements for evolution to take place: one of the central concepts is the above mentioned fitness; without variation in it there would be no evolution. Essentially, fitness measures the ability of an individual to survive and reproduce. The consequence of the difference between the values of fitness from one organism to another is the higher number of offspring of the one with the higher fitness value<sup>6</sup>. Variation, the series of changes occurring in a species, is the second condition; it can be inherited or due to the environment. Inherited variation is a characteristic that has been genetically inherited: children receive half of their genetic information from each parent, forming a new genetic set for the new individual. Examples of inherited variation are eyes color and hair color. Organisms can also be affected by changes in their environment; climate and culture are two factors of environmental variation<sup>7</sup>. Third is reproduction, the process of “creating” new offsprings from parents; it can be asexual, where reproduction involves only one organism, and sexual, the most common form, that needs two different members of the species to occur<sup>8</sup>. Heredity is the fourth essential element: it is the process by which a parent passes some traits to offsprings. Heredity helps variation in the population and therefore may cause some species to evolve<sup>9</sup>.

The genetic mechanisms behind evolution were first uncovered by the work of Gregor Mendel in the 20<sup>th</sup> century. Each cell of an individual has chromosomes, or strands of DNA, and each chromosome consist of genes, that have their own position (locus) and are inherited from the parents during reproduction in a process called crossover; every gene encodes a different trait, such as height and eye color, while alleles are different varieties of genes: for example, there’s an allele for brown eyes, one for black eyes, and such. The set of all chromosomes is called genome. New traits can be introduced in a population by a gene flow, the transfer of alleles or genes, from

other populations or by mutation, a change in the structure of a gene that is usually the consequence of an error in the transcription of DNA<sup>10,11</sup>.

From the 1950s several computer scientist began to think of a way to apply evolutionary concepts to solve computational and optimization problems. Genetic algorithms (GAs), subfield of evolutionary computation, were invented and developed during the 60s at the University of Michigan by John Holland and its students<sup>12</sup>. The field of evolutionary computation is considered a strong tool to deal with complex problem solving. It is often required of a computer program to easily adapt to an evolving environment, to be innovative and original and to perform tasks that are simply too difficult to be programmed manually, especially in the field of artificial intelligence. The theory of evolution has been an inspiration to develop these concepts because it is a way of find optimal solutions; in biological terms the solutions are found in the set of all possible genetic sequences and are the individuals with the highest “fitness”. Also, as the human organism develops complex mechanisms to solve problems that would threaten his survival, computer programs need to develop innovative ways to get an optimal solution. The constant changes in the computational world are well represented by the ever-changing natural world. Another parallelism is the ability to evaluate different possibilities at the same time, as the different species that are simultaneously tested on their fitness<sup>13,14</sup>.

The language used to describe the elements of a genetic algorithm is directly derived from biology, from which they draw inspiration. The set of all possible solutions to the problem is called the search space, or *population*. Each feasible solution is an *individual*, characterised by two factors: a *chromosome*, which is typically encoded as a bit string, and a value of *fitness*. Short blocks of different bits of a chromosome that encode a different parameter of the solution are the

*genes*, which can have different values, or *alleles*: in a bit string an allele is 0 or 1. The position of a gene in a chromosome is the *locus* and the collection of all the genes is the *genotype*<sup>15,16</sup>.

In a genetic algorithm, some individuals are chosen as the parents in base of their fitness; individuals with higher fitness are more likely to be selected. The parents are then allowed to produce offspring by applying reproductive operators, such as crossover and mutation. Crossover

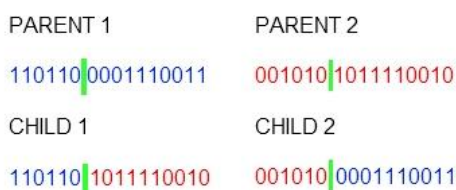


Figure 1. Crossover

operators (Figure 1) recombine the chromosomes of the parents. The most common type is called single point crossover. A random point in which to cut the two genes is selected. The first part of the first parent is then combined with the second part of the second parent to

create a child. Similarly, the second part of the first parent is combined with the first part of the second parent to produce the second child. Crossover does

not always occur; its probability is just between 60% and 70%. Mutation is rarer, it occurs with a probability between 0.01% and 0.02%. When it happens, one bit changes from 0 to 1 or



Figure 2. Mutation

vice versa (Figure 2). Mutation assures that the population doesn't become too static and therefore incapable of reproduction. The offsprings form a new generation and the process starts from the beginning. Again, fitness is evaluated for each individual of this new population and the selected ones reproduce until a certain number of new generations have been produced and analysed and the problem is satisfied<sup>17,18,19</sup>. Figure 3 shows the pseudocode for a typical GA.

n is the number of individuals in the population  
 x is the crossover rate  
 y is the mutation rate

```

GA(n,x,y)
  // Initialization: create an initial population
  k:=0;
  Pk:= a population of n individuals
  // Evaluation: calculate the fitness of every individual
  Evaluate fitness(i) for every i ∈ Pk
  Repeat until fitness of fittest individual in Pk is high enough
  {
    // Create a new population k+1:
    // Selection: select the best individuals in base of their
    fitness
    Select a fraction of n individuals and insert them in k+1
    // Crossover:
    Select x*n members of Pk, pair them up to produce offspring
    and add the offspring to Pk+1
    // Mutation:
    Select y*n members of Pk and invert a randomly selected bit
    in each of them
    // Evaluation: evaluate the fitness of every individual of k+1
    Evaluate fitness(i) for every i ∈ Pk+1
    // Increment the variable
    k:= k+1
  }
  return the fittest individual of Pk

```

Figure 3. Typical GA pseudocode<sup>20,21,22</sup>

Genetic algorithms can be applied in a variety of different situations in science, engineering, but also in economics and social sciences. Areas of application are:

- Combinatorial optimization problems: an example of this is the Travelling Salesman Problem. The aim, given a list of cities and their distances, is to find the shortest route to visit each city once and then return to the starting point. The

benefits of the use of genetic algorithms are linked to the mutation and crossover operators. The former allows a parallel search, while the latter allows individuals to share information<sup>23</sup>.

- Economics: genetic algorithms are applied in game theory, a field of economics that studies strategic decision making, the cooperation and conflicts between rational agents making decisions. They are also applied in the game of prisoner's dilemma, a game that illustrates the possibility that two rational beings might decide not to cooperate, even if it seems their best interest to do so. Genetic algorithms can be used to explore the possibility of developing cooperative behaviour. Economics uses the following features of genetic algorithms: the ability to represent individuals as part of a population whose agents differ

in strategy, the processing of information in parallel and the selection of better-performing strategies<sup>24</sup>.

- Scheduling: GAs are often applied in order to solve production scheduling problems because they operate in a set of various solutions in parallel. The population of solutions consist in different and often conflicting answers. The goal of the genetic algorithm is to find the best way to assign the time to the activities to maximize the overall performance<sup>25</sup>.
- Robotics: genetic algorithms are applied in this field to obtain good navigating systems. The navigation system must be able to direct the robot in a constricted environment and get it to its destination, without it crashing into objects and getting lost. The genetic algorithm scans the intermediary points of every candidate path to find the most direct move to the destination; when an obstacle is found, it stops and goes back to the last free path section analysed<sup>26</sup>.

It is proven that genetic algorithms are able to perform very well in real life and complex situations, because they are adaptive to the environment to which they are applied, just like real life organisms adapt to the natural environment in which they find themselves. The study of GAs is also used to model ecosystems, immune, cognitive, and social systems and in the future years it will become even more prominent in the modelling of artificial life systems<sup>27</sup>.

## References

1. Howell FC. Early Man. New York: Time Life Books; 1965.
2. Khan Academy. Introduction to Evolution and Natural Selection [Internet]. Available from: <http://www.khanacademy.org/science/biology/evolution-and-natural-selection/v/introduction-to-evolution-and-natural-selection>
3. Darwin C. On the origin of species. Oxford: Oxford University Press; 1996.
4. New Scientist, Pickrell J. Introduction: evolution [Internet]. 4 September 2006. Available from: <http://www.newscientist.com/article/dn9953-instant-expert-evolution.html>
5. Open Yale Courses. Principles of Evolution, Ecology and Behavior: The nature of evolution: selection, inheritance and history [Internet]. 27 August 2009. Available from: <http://www.youtube.com/watch?v=VjgHd6HKtvE>
6. Allen Orr H. Fitness and its role in evolutionary genetics. Nature Reviews Genetics. [Internet]. 2009;10(8): 531-539. Available from: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2753274/pdf/nihms121275.pdf>
7. BBC Bitesize: Science. Variation and Classification – Introduction [Internet]. Available from: [http://www.bbc.co.uk/schools/ks3bitesize/science/organisms\\_behaviour\\_health/variation\\_classification/revise1.shtml](http://www.bbc.co.uk/schools/ks3bitesize/science/organisms_behaviour_health/variation_classification/revise1.shtml)
8. University of California Berkeley. Understanding Evolution, Mechanisms: the processes of evolution [Internet]. Available from: [http://evolution.berkeley.edu/evolibrary/article/0\\_0\\_0/evo\\_14](http://evolution.berkeley.edu/evolibrary/article/0_0_0/evo_14)
9. Downes SM. "Heritability", *The Stanford Encyclopedia of Philosophy* [Internet] Available from: <http://plato.stanford.edu/archives/fall2010/entries/heredity/>
10. Mitchell M. An introduction to genetic algorithms. Cambridge, Mass: MIT University Press; 1996.

11. Goldberg GE. Genetic algorithms in search, optimization, and machine learning. 23<sup>rd</sup> edition. Boston, Mass: Addison Wesley Longman; 2002.
12. Reeves CR, Rowe JE. Genetic algorithms principles and perspectives - A guide to GA theory. London: Kluwer Academic Publishers; 2003.
13. Mitchell M. An introduction to genetic algorithms. Cambridge, Mass: MIT University Press; 1996.
14. Indiana University Bloomington School of Informatics and Computing, Yaeger L. Intro to Genetic Algorithms [Internet]. Available from:  
<http://informatics.indiana.edu/larryy/al4ai/lectures/03.IntroToGAs.pdf>
15. Mitchell M. An introduction to genetic algorithms. Cambridge, Mass: MIT University Press; 1996.
16. Genetic algorithms warehouse, Skinner M. Genetic algorithms overview [Internet]. Available from: <http://geneticalgorithms.ai-depot.com/Tutorial/Overview.html>
17. Jamshidi M. Tools for Intelligent Control: Fuzzy Controllers, Neural Networks and Genetic Algorithms. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*. [Internet]. 2003;361(1809): 1781-1808. Available from:  
<http://www.jstor.org/stable/3559221>
18. Indian Institute of Technology Bombay Department of civil engineering, Mathew TV. Genetic Algorithm [Internet]. Available from:  
[http://www.civil.iitb.ac.in/tvm/2701\\_dga/2701-ga-notes/gadoc/gadoc.html#foot10](http://www.civil.iitb.ac.in/tvm/2701_dga/2701-ga-notes/gadoc/gadoc.html#foot10)
19. Griffith University Faculty of engineering and applied science, Randall M. The future and applications of genetic algorithms [Internet]. Available from:  
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.3716>



20. University College Cork: Computer Science. Genetic algorithms [Internet]. Available from: <http://www.cs.ucc.ie/~dgb/courses/tai/notes/handout12.pdf>
21. Obitko M. Genetic algorithms [Internet]. Available from: <http://www.obitko.com/tutorials/genetic-algorithms/index.php>
22. Jacobson L. Creating a genetic algorithm for beginners. The project spot. Weblog. Available from: <http://www.theprojectspot.com/tutorial-post/creating-a-genetic-algorithm-for-beginners/3>
23. Imperial College London Computing. Application of genetic algorithm [Internet]. Available from: [http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol1/tcw2/article1.html](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/tcw2/article1.html)
24. Dawid H, Kopel M. On Economic Applications of the Genetic Algorithm: A Model of the Cobweb Type. Journal of Evolutionary Economics. [Internet]. 1998;8(3): 297-315. Available from: [http://download.springer.com/static/pdf/565/art%253A10.1007%252Fs001910050066.pdf?auth66=1354991276\\_76a28a74b7598bdfb4856c2da3203398&ext=.pdf](http://download.springer.com/static/pdf/565/art%253A10.1007%252Fs001910050066.pdf?auth66=1354991276_76a28a74b7598bdfb4856c2da3203398&ext=.pdf)
25. Wall MB. A Genetic Algorithm for Resource-Constrained Scheduling. Thesis (Ph.D.). Massachusetts Institute of Technology. 1996. Available from: <http://lancet.mit.edu/~mwall/phd/thesis/thesis.pdf>
26. Man KF, Tang KS, Kwong S. Genetic Algorithms: Concepts and Applications. IEEE Transactions on industrial electronics. [Internet] 1996;43(5): 519-534. Available from: [ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia707\\_1s06/textos/00538609.pdf](ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia707_1s06/textos/00538609.pdf)
27. Imperial College London Computing. Genetic algorithm [Internet]. Available from: [http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/tcw2/report.html#Vis\\_Concl](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html#Vis_Concl)