

# CSG2341 Intelligent Systems

## Additional notes on Evolutionary Algorithms

Evolutionary algorithms are based on biological or “natural” evolution. Negnevitsky gives some very brief information on biological evolution at the start of Ch 7.2. These notes provide some additional background, and a summary of strengths and weaknesses of evolutionary algorithms.

### What is evolution?

Nature has a process that optimises the design of living organisms. This process is called evolution. Evolution is based on a set of key principles, including the use of a genetic code and Darwinian natural selection. Today’s scientists and engineers can use the same principles to solve difficult optimisation problems using computers to simulate evolution.

We begin with a simplified and abstracted description of evolution. Evolution is the process whereby successive generations of an organism are modified so as to operate better in their environment. We can think of this as a process of refining a design (the “recipe” for the organism).

The key point of Darwin’s theory of natural selection is that a creature’s traits (eye colour, length of toes, intelligence etc) tend to be inherited by its descendants, and that these traits affect the creature’s chances of survival and reproduction. Once these principles are accepted, the process of evolution by “survival of the fittest” is assured. Those creatures whose traits help them to survive better, and thus produce more offspring, inevitably become more numerous generation after generation, and good design features accumulate.

We know now how traits are inherited (though Darwin did not). Watson and Crick’s discovery of the structure of DNA in 1953 gave us the knowledge we needed to understand the mechanisms of chromosomes and genes.

Putting these together, we can describe the process of evolution:

*Every organism has within its cells a “recipe” according to which its body is created - its **genotype**. In humans, this consists of a sequence of 23 pairs of chromosomes – our DNA. When a creature reproduces, part of its DNA is passed on to its offspring. In sexual reproduction, this part of the creature’s DNA is combined with part of the other parent’s DNA. This is called **recombination**. Recombination allows traits of the parents to be combined in different ways in their offspring. Very occasionally, an error happens when the DNA is copied from parent to offspring. This is called a **mutation**. Mutations are usually deleterious to the offspring, but, rarely, may be beneficial.*

*While the offspring is developing into a new creature, parts of the DNA determine just how the body of new creature (its phenotype) will develop. The part of the DNA that determines a particular trait is called the **gene** for that trait, and different configurations of that gene are called **alleles**.*

*When the new creature ventures into the world, its chances of surviving long enough to reproduce (its **fitness**) depend in part on these inherited traits. Without mutation, this process simply juggles different combinations of existing alleles. Through mutation, new alleles can be introduced, producing truly different new creatures. Over many generations, alleles that tend to produce fitter creatures will predominate, ensuring “survival of the fittest”.*

## **How does DNA work?**

First we describe the main steps in the development of a cell from DNA (epigenesis):

*Genomes consist of DNA arranged as of a set of chromosomes (humans have 23 pairs). Each chromosome is a sequence of bases*

- *A – Adenine*
- *G – Guanine*
- *C – Cytosine*
- *T – Thymine*

*Each sequence is arranged spatially as a double-helix. The helix has two strands, each a linear sequence of bases, and the two strands contain complementary pairs of bases. Each strand is converted into messenger RNA by a process called splicing or transcription.*

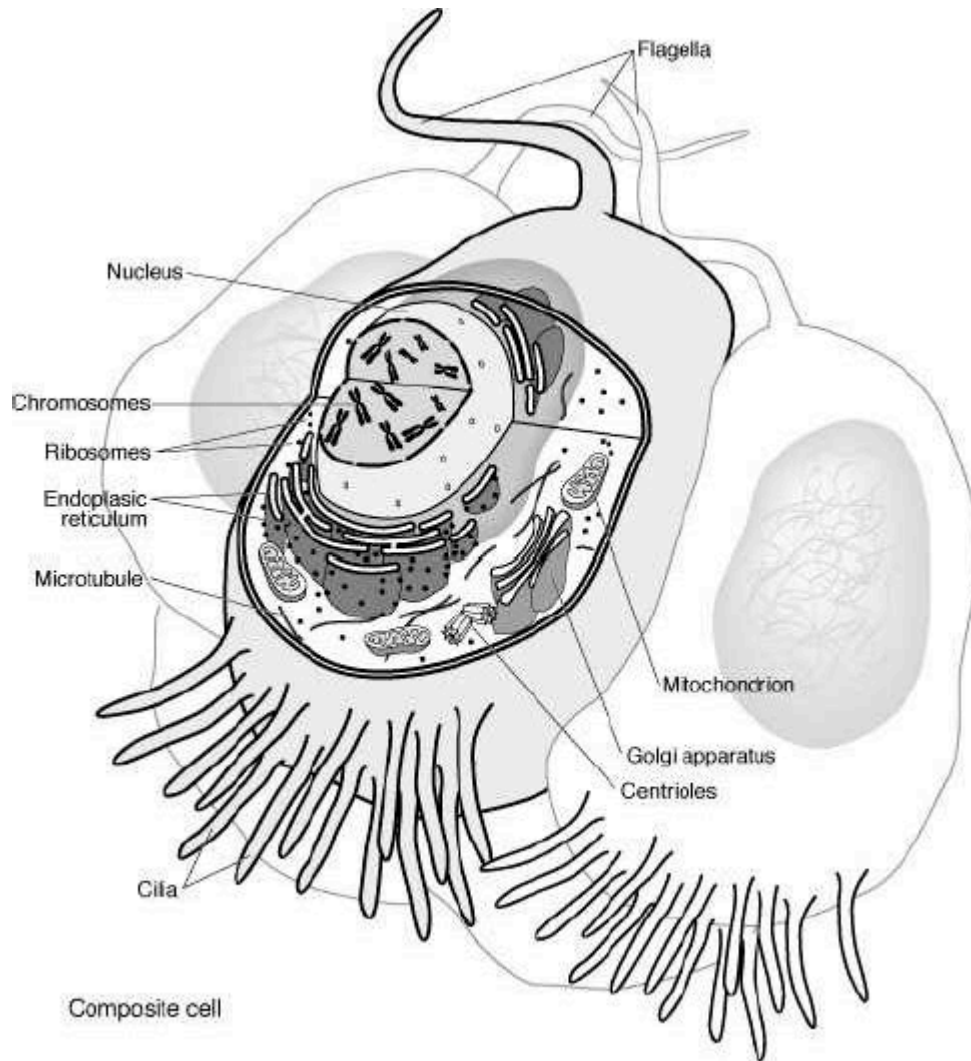
*In splicing, parts of the DNA not used for protein synthesis are discarded, and bases are exchanged for their complementary pairs (T for A, G for C, and vice-versa, except that in RNA U – Uracil - is used instead of T). Other changes can also occur.*

*The resulting string of bases is considered as segmented into sets of three bases, each called a codon. There are thus 64 possible codons. In the next step, each codon is translated into one of 20 possible amino-acids (three of the 64 operations translate to a “no-op”). The resulting amino acids make up polypeptides (chains of amino-acids). Which in turn combine to form proteins. The proteins form cells.*

*The sequence of bases in a chromosome that results in the formation of a particular polypeptide is considered a gene. At the beginning and end of its sequence, a gene has special base sequences called controlling elements, which take part in the mechanism of transcription.*

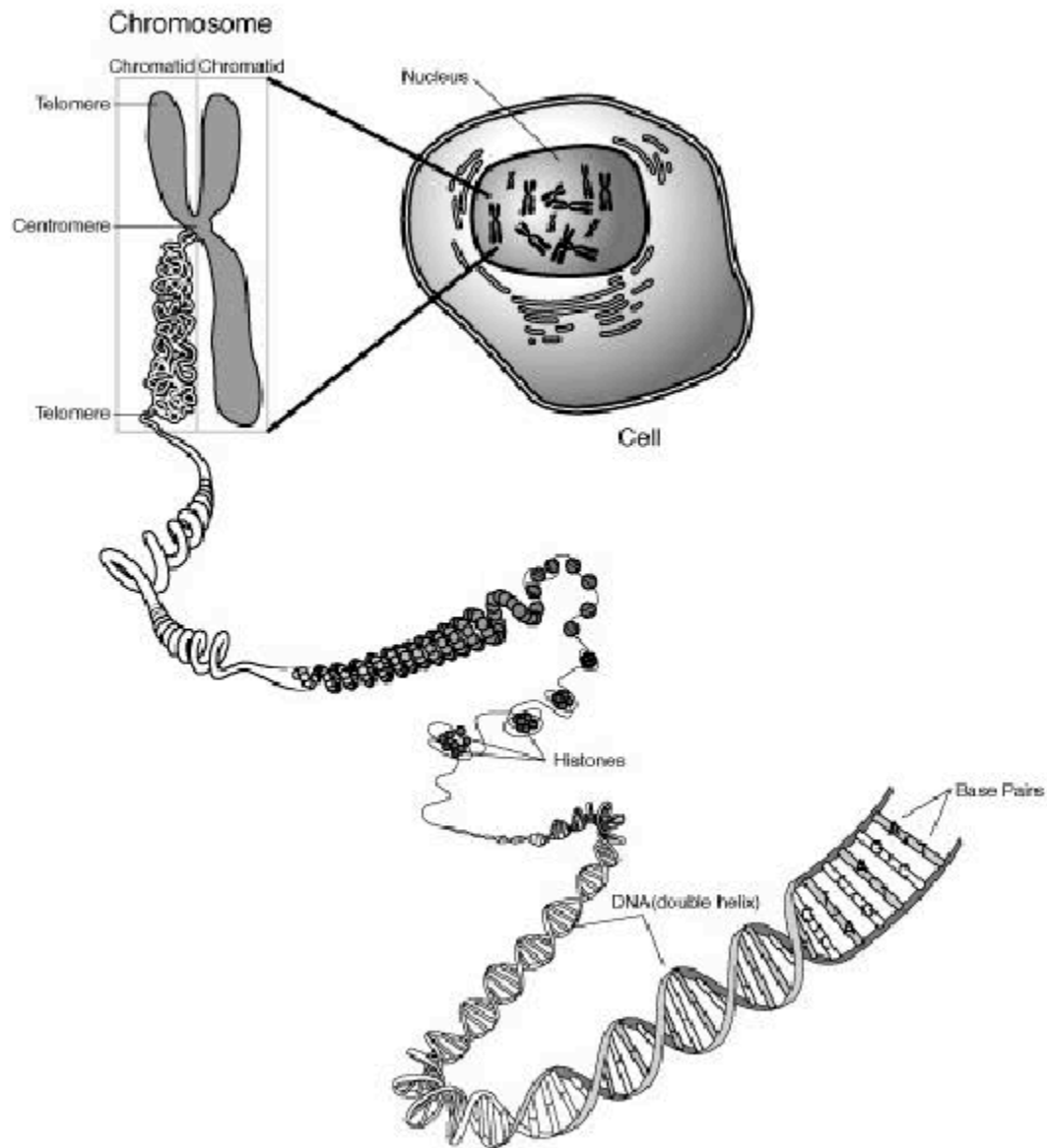
The above is generally true for all biological life on earth. The detailed mechanisms are different for single-celled (e.g. bacteria) and multi-celled (e.g. us) organisms. A small proportion of genetic material uses a different coding

scheme entirely (e.g. mitochondrial DNA). Viruses and phages hijack the mechanisms of their hosts and have no cells of their own.



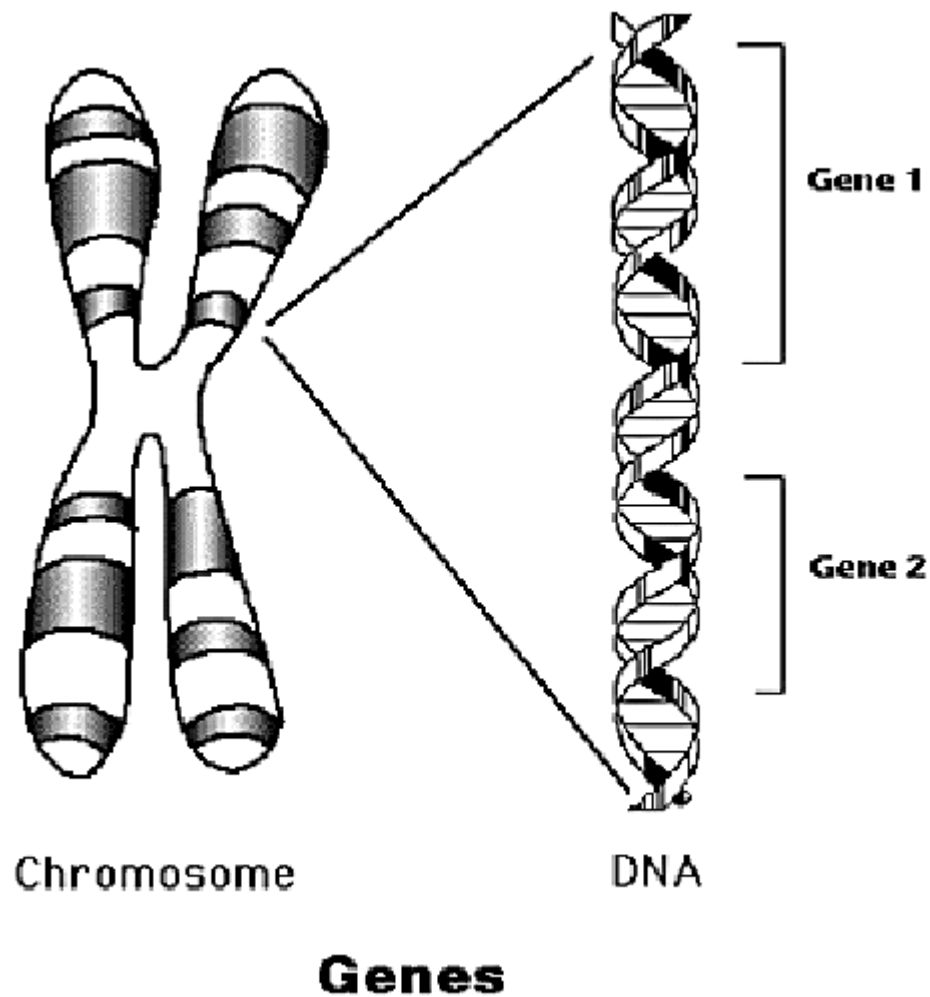
**Figure 1 - Diagram of a eukaryote cell**

(Diagram from Richard Simon's Bioinformatics course,  
[http://chortle.ccsu.edu/cs580\\_70/LectureNotes\\_2001/lecturepage\\_2001.html](http://chortle.ccsu.edu/cs580_70/LectureNotes_2001/lecturepage_2001.html))



**Figure 2 - Expanding view of DNA in a cell**

(Diagram from Richard Simon's Bioinformatics course,  
[http://chortle.ccsu.edu/cs580\\_70/LectureNotes\\_2001/lecturepage\\_2001.html](http://chortle.ccsu.edu/cs580_70/LectureNotes_2001/lecturepage_2001.html))



**Figure 3 - Genes within a chromosome**

(Diagram from Richard Simon's Bioinformatics course,  
[http://chortle.ccsu.edu/cs580\\_70/LectureNotes\\_2001/lecturepage\\_2001.html](http://chortle.ccsu.edu/cs580_70/LectureNotes_2001/lecturepage_2001.html))

Reproduction is achieved through a different mechanism from the one described above. Single-celled organisms simply need to replicate, which they can do by unzipping the double helix, and using each half as a template for a new copy. Multi-celled organisms often reproduce sexually. In sexual reproduction, two kinds of crossover occur:

- For each of the (in humans, 23) pairs of chromosomes, the offspring gets 1 chromosome from each parent. Remember each chromosome contains a number of genes, so each individual gets two variations (alleles) of each gene. These genes can interact in various ways (e.g. dominance and recessiveness).
- Within each chromosome, one- or two-point crossover can occur.

In addition, some traits are determined by a combination of genes (pleiotropy) and some genes affect several traits (polygeny). All these things combine to provide a great deal of potential for variation and new combinations within a population.

In addition, both sexual and asexual reproduction can be affected by mutations:

- Point mutations – change a base in the DNA, which may change the corresponding amino-acid in the polypeptide;
- Inversions ( a section of DNA is reversed);
- Other mutations.

### **What is an Evolutionary Algorithm?**

An evolutionary algorithm is a simulation of the processes of evolution, used to search for good solutions to a problem, often a design problem of some sort. The steps in a typical evolutionary algorithm are:

- Start with a population of solutions, each with its own genome (like a population of organisms)
- Test them out and give each one a FITNESS (for survival of the fittest)
- A new generation of solutions is created by
  - Choosing old solutions, favouring FITTER ones (these are the ones that survive to be parents)
  - Possibly crossing over the genomes of the two parents
  - Possibly mutating the resulting genomes
  - Developing new solutions (children) from these genomes
- Keep creating new generations until it's time to stop

### **What are evolutionary algorithms good for?**

So-called evolutionary algorithms are now routinely used to solve real-world problems that are impractical to solve by traditional analytical methods. This may be because the problems are computationally intractable, or because they involve a messy, highly non-linear, complex system. A key advantage of evolutionary algorithms is that they operate on general principles, requiring only a way to calculate which potential solutions are better (“fitter”). This is enough to guide the algorithm towards high quality solutions.

Some of the advantages of evolutionary algorithms are:

- Can handle complex, non-linear problems where it is not feasible to develop analytical models – the “fitness” calculation used to evaluate designs can be arbitrarily complex.
- Do not become “stuck” as simpler hill-climbing methods can.
- Can produce superior solutions that are non-obvious for a human designer.
- Can perform multi-objective optimisation, balancing different or even conflicting criteria.
- Can produce more than one high quality solution, so that the user can carry out more detailed validation on a small set of candidate solutions.
- Can easily be hybridised/integrated into other systems.

Some disadvantages are:

- Can be slow – depending on the size of the problem, the complexity of the calculations and the convergence rate, computations may take anywhere from a few seconds to a few days to run.
- Can be difficult to design a good representation.

Many classical methods begin by posing the problem as finding the best values for a fixed number of unknown variables. But in many cases, the numbers of variables required is not known a-priori. So these methods can only be used to solve part of the whole problem, or an approximation, or a restricted version of the problem.

Numerical methods also usually rely on the problem being not too complicated. For example, some methods require the problem to be stated in terms of linear equations and inequalities. Some allow at most quadratic equations. Most require that the problem can be described in terms of functions that are differentiable.

Many existing methods are “local” optimisers, meaning that they can be fooled into settling for solutions that cannot be improved by making small changes, but are far from the true best solution (the global optimum). By contrast, evolutionary methods are *global optimisers* (along with other methods like simulated annealing, tabu search, ant colony optimisation etc).