SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
COIMBATORE-35
Accredited by NBA-AICTE and Accredited by NAAC – UGC with A+ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

ARTIFICIAL INTELLIGENCE FOR ELECTRICAL ENGINEERING

TOPIC : CROSS OVER AND MUTATION, GENERATIONAL CYCLE

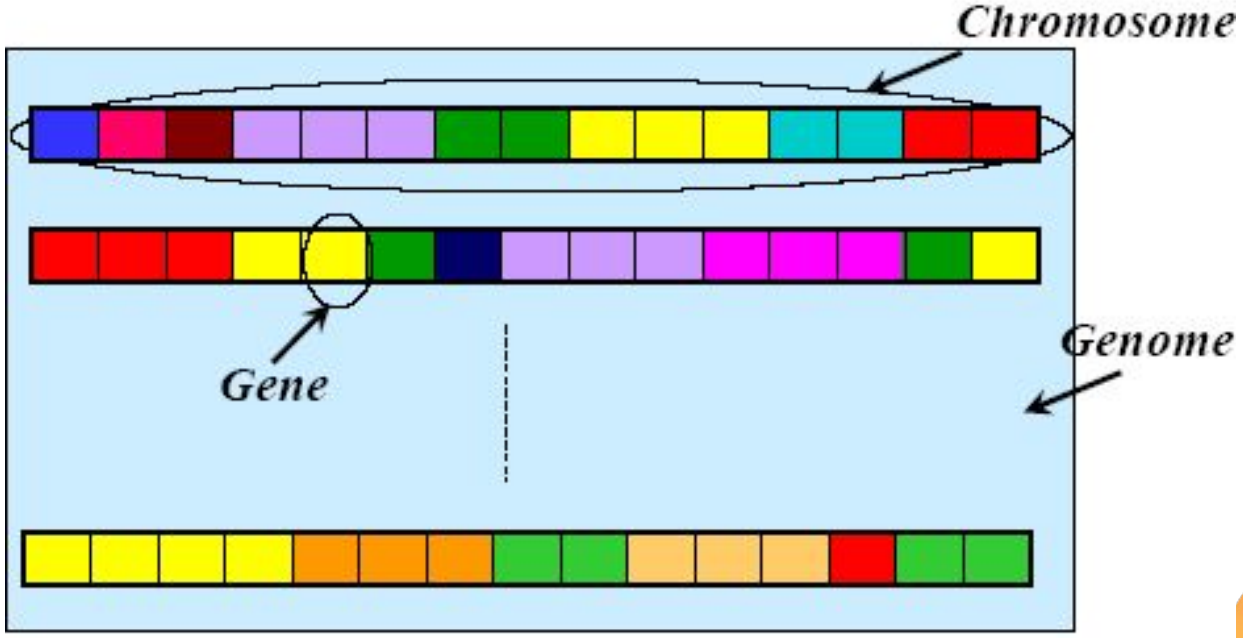
INTRODUCTION

- After scientists became disillusioned with classical and neo-classical attempts at modeling intelligence, they looked in other directions.
- Two prominent fields arose, connectionism (neural networking, parallel processing) and evolutionary computing.
- It is the latter that this essay deals with - genetic algorithms and genetic programming.

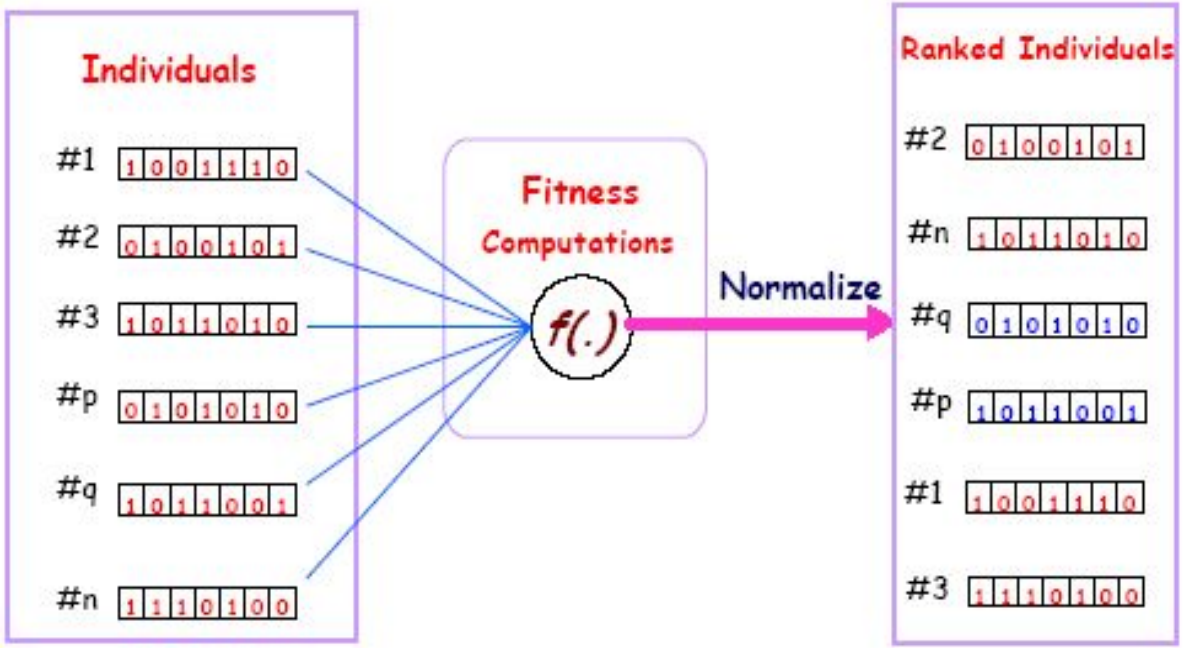
WHAT IS GA

- A genetic algorithm (or GA) is a search technique used in computing to find true or approximate solutions to optimization and search problems.
- Genetic algorithms are categorized as global search heuristics.
- Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).

CHROMOSOME, GENES AND GENOMES



A FITNESS FUNCTION



CROSSOVER

- The most common type is single point crossover. In single point crossover, you choose a locus at which you swap the remaining alleles from on parent to the other. This is complex and is best understood visually.
- As you can see, the children take one section of the chromosome from each parent.
- The point at which the chromosome is broken depends on the randomly selected crossover point.
- This particular method is called single point crossover because only one crossover point exists. Sometimes only child 1 or child 2 is created, but oftentimes both offspring are created and put into the new population.
- Crossover does not always occur, however. Sometimes, based on a set probability, no crossover occurs and the parents are copied directly to the new population. The probability of crossover occurring is usually 60% to 70%.

CROSSOVER


Parent 1
1011010 01010011 0
Parent 2
0011010 11011010 1
Child 1
1011010 11011010 1
Child 2
0011010 01010011 0

MUTATION

- After selection and crossover, you now have a new population full of individuals.
- Some are directly copied, and others are produced by crossover.
- In order to ensure that the individuals are not all exactly the same, you allow for a small chance of mutation.
- You loop through all the alleles of all the individuals, and if that allele is selected for mutation, you can either change it by a small amount or replace it with a new value. The probability of mutation is usually between 1 and 2 tenths of a percent.
- Mutation is fairly simple. You just change the selected alleles based on what you feel is necessary and move on. Mutation is, however, vital to ensuring genetic diversity within the population.

MUTATION

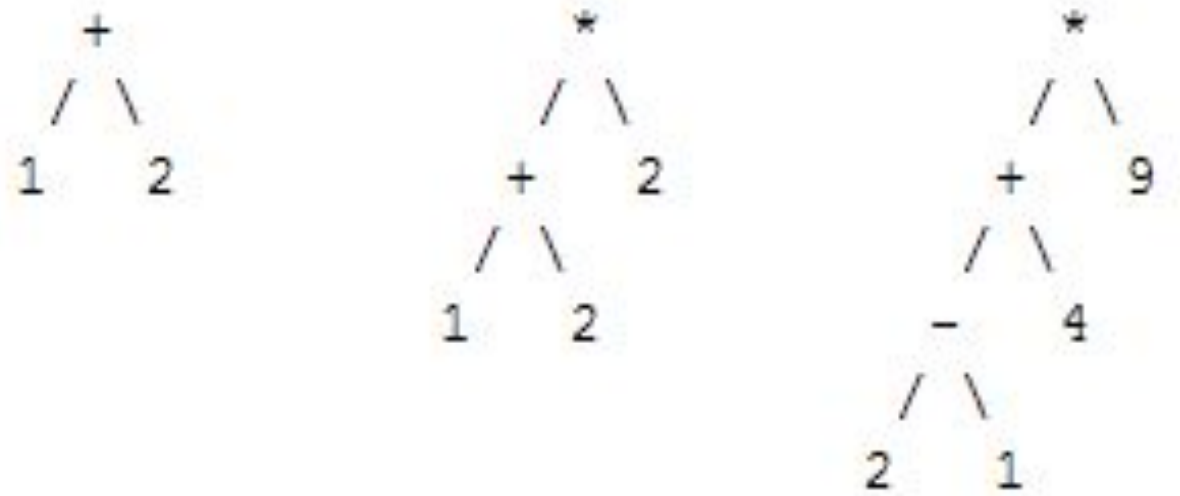
SYMBOLIC AI VS. GENETIC ALGORITHMS

- Most symbolic AI systems are very static.
- Most of them can usually only solve one given specific problem, since their architecture was designed for whatever that specific problem was in the first place.
- Thus, if the given problem were somehow to be changed, these systems could have a hard time adapting to them, since the algorithm that would originally arrive to the solution may be either incorrect or less efficient.
- Genetic algorithms (or GA) were created to combat these problems; they are basically algorithms based on natural biological evolution.

GENETIC PROGRAMMING

- In programming languages such as LISP, the mathematical notation is not written in standard notation, but in prefix notation. Some examples of this:
- + 2 1 : 2 + 1
- * + 2 1 2 : 2 * (2+1)
- * + - 2 1 4 9 : 9 * ((2 - 1) + 4)
- Notice the difference between the left-hand side to the right? Apart from the order being different, no parenthesis! The prefix method makes it a lot easier for programmers and compilers alike, because order precedence is not an issue.
- You can build expression trees out of these strings that then can be easily evaluated, for example, here are the trees for the above three expressions.

GENETIC PROGRAMMING

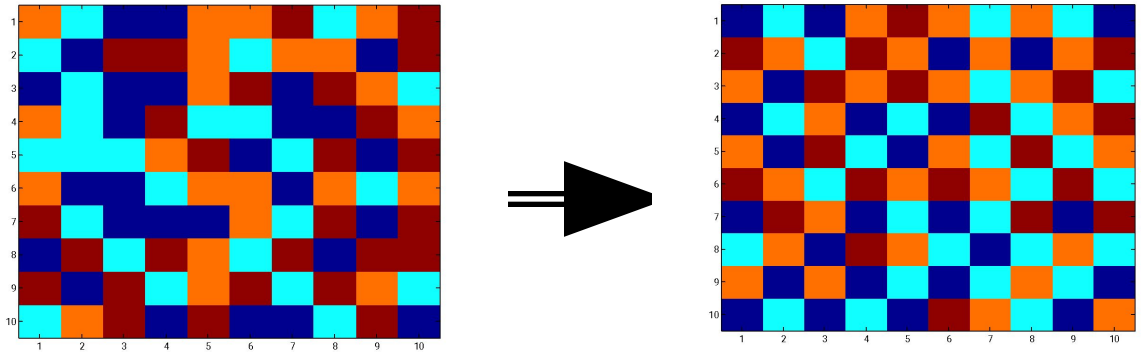


EVOLVING NEURAL NETWORKS

- Many would think that a learning function could be evolved via genetic programming. Unfortunately, genetic programming combined with neural networks could be *incredibly* slow, thus impractical.
- As with many problems, you have to constrain what you are attempting to create.
- For example, in 1990, David Chalmers attempted to evolve a function as good as the delta rule.
- He did this by creating a general equation based upon the delta rule with 8 unknowns, which the genetic algorithm then evolved.

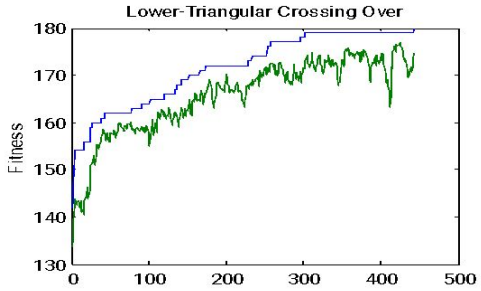
CHECKBOARD EXAMPLE

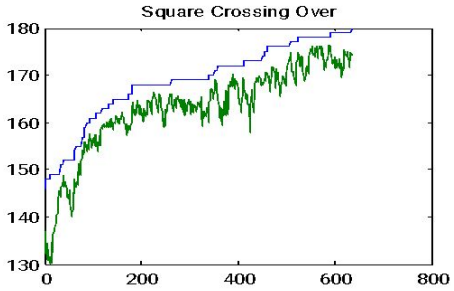
- We are given an ***n*** by ***n*** checkboard in which every field can have a different colour from a set of four colors.
- Goal is to achieve a checkboard in a way that there are no neighbours with the same color (not diagonal)

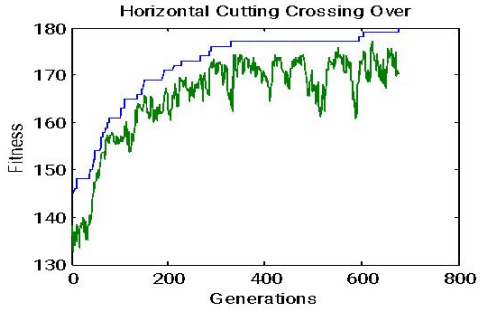


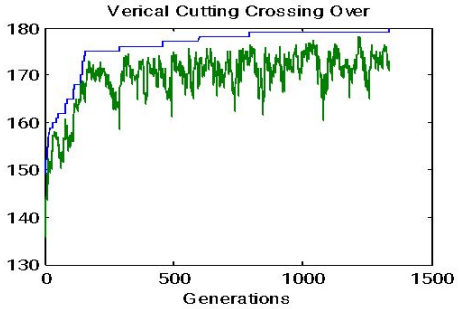
CHECKBOARD EXAMPLE CONT'D

- Fitnesscurves for different cross-over rules:

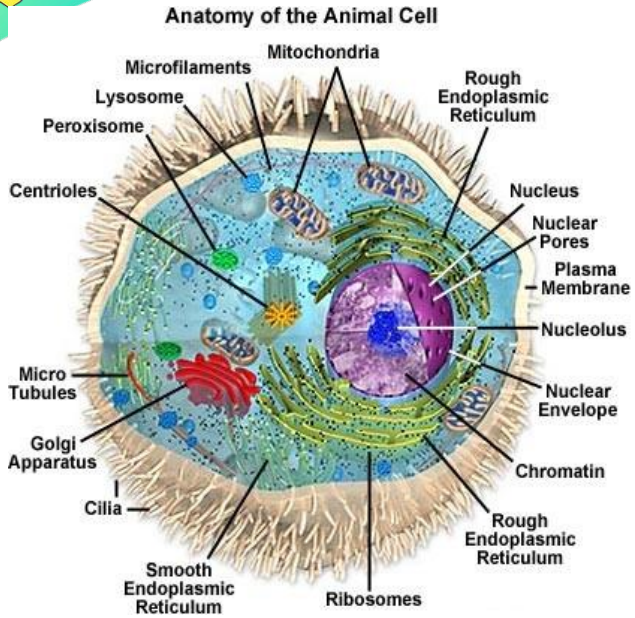


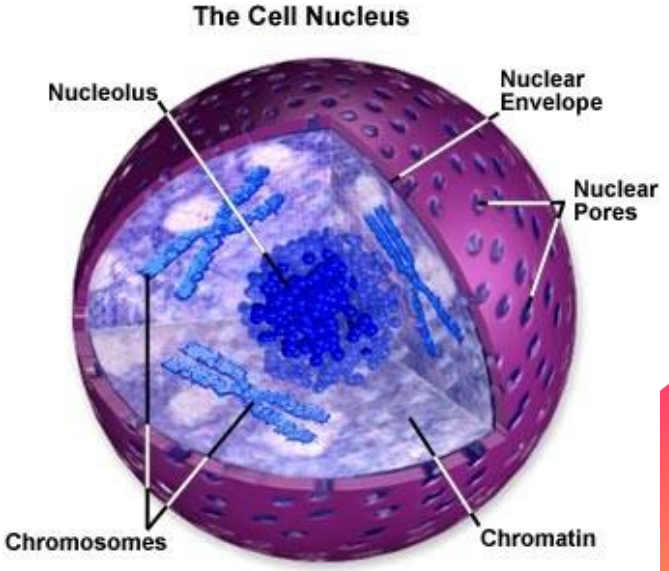






THANK YOU


Anatomy of the Animal Cell
Microfilaments
Mitochondria
Lysosome
Rough Endoplasmic Reticulum
Peroxisome
Centrioles
Nucleus
Nuclear Pores
Plasma Membrane
Nucleolus
Micro Tubules
Nuclear Envelope
Golgi Apparatus
Chromatin
Cilia
Rough Endoplasmic Reticulum
Smooth Endoplasmic Reticulum
Ribosomes


The Cell Nucleus
Nucleolus
Nuclear Envelope
Nuclear Pores
Chromosomes
Chromatin