Computational Intelligence
Lecture 20:Intorcution to Genetic Algorithm

Farzaneh Abdollahi

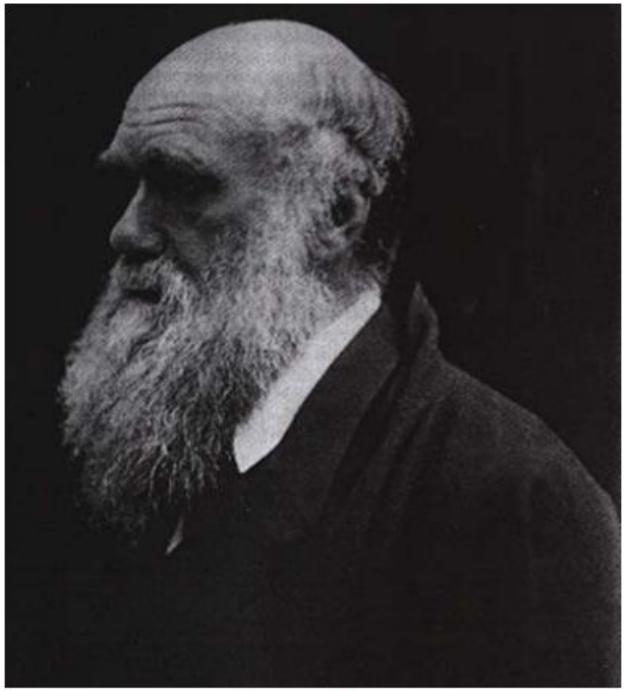
Department of Electrical Engineering

Amirkabir University of Technology

Fall 2012

Genetic algorithms (GA)

- Genetic algorithms (GA) are inspired by Darwin's theory about evolution.
- " Genetic Algorithms are good at taking large, potentially huge search spaces and navigating them, looking for optimal combinations of things, solutions you might not otherwise find in a lifetime."
  David E. Goldberg Computer Design, May 1995



Charles Darwin
1809–1882

Genetic algorithms (GA)

- Evolutionary computing was firstly introduced in the 1960s by I. Rechenberg in "Evolution strategies"
- GA is developed by John Holland, University of Michigan (1970s)
  - Holland's book "Adaption in Natural and Artificial Systems" was published in 1975.
- GA is search algorithm based on of biological evolution
- It provides effective techniques for optimization and machine learning applications

Evolutionary Computation

- It is inspired by biological evolution.
- It is started with a set of solutions (represented by chromosomes) called population.
- The population evolves over time, with the fittest at each generation contributing the most offspring to the next generation
- This is motivated by a hope, that the new population will be better than the old one.

Genetic Algorithm

1. Basic genetic population
   - It is named gene or chromosome
   - Chromosomes could be:
     - Bit strings like (0101...1100)
     - Real numbers like (43.2 – 33.1...0.089.2)
     - Any element like ($E11E3E7...E1E15$)
     - Lists of rules like ($R1R2R3...R22R23$)
     - Program elements like (*geneticprogramming*)
2. Initialization procedure (creation)
   - Create an initial population, it is usually generated randomly and can be any desired size, from just a few individuals to thousands.
3. Selection of parents (reproduction)

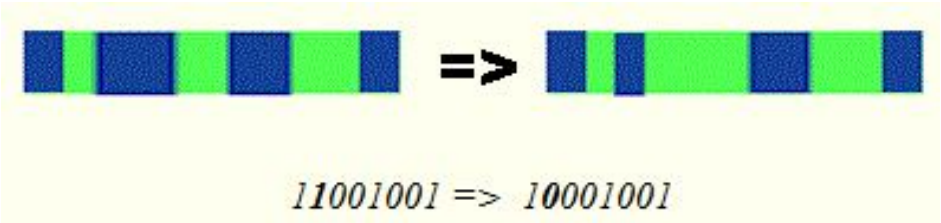
Genetic Algorithm Cont'd

4. Evaluation function (environment)
   - Each member of the population is then evaluated and a 'fitness' is calculated for that individual.
   - The fitness is calculated by how well it fits our the requirements such as 'faster algorithms are better', or more complex, 'stronger materials are better but they shouldn't be too heavy'.
   - The evaluator decodes a chromosome and assigns it a fitness measure
   - The evaluator is defined based on the problem to be solved

5. Genetic operators (mutation, recombination)
   - The operators are
     - Mutation (alter, local changes)
     - Crossover (recombination)
   - Performance of GA very depends on the operators.
   - Type and implementation of operators depends on encoding and also on a problem

Genetic operators

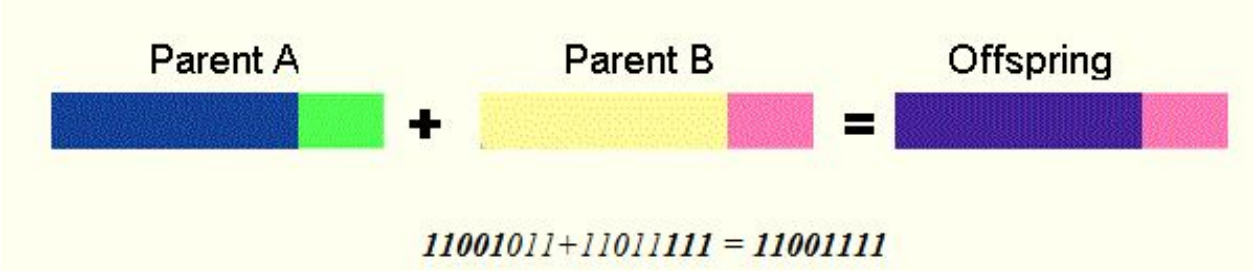
- Mutation (alter, local changes)
  - Bit inversion: selected bits are inverted



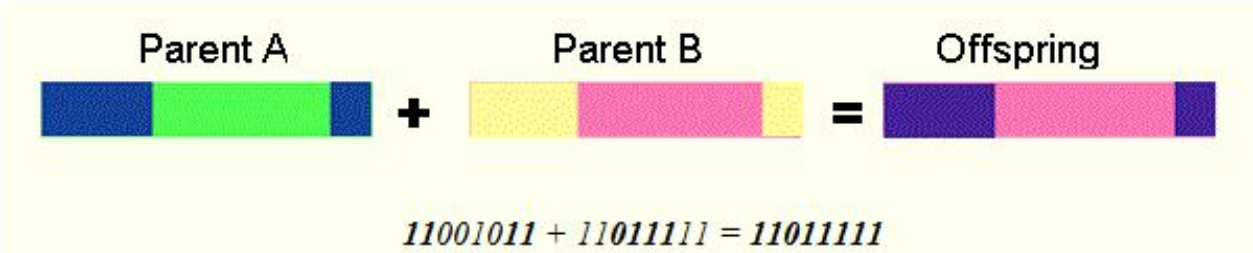
  - Order changing:two numbers are selected and exchanged (123456897) $\Rightarrow$ (183456297)
  - Adding a small number :(for real value encoding)to selected values is added (or subtracted) a small number (1.295.682.864.115.55) $\Rightarrow$ (1.295.682.734.225.55)

Genetic operators

- Crossover (recombination)
  - Single point crossover: one crossover point is selected, binary string from beginning of chromosome to the crossover point is copied from one parent, the rest is copied from the second parent

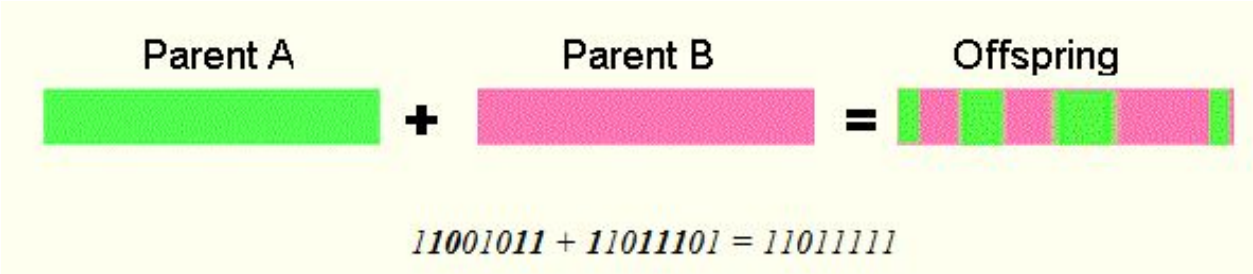


  - Two point crossover: two crossover point are selected, binary string from beginning of chromosome to the first crossover point is copied from one parent, the part from the first to the second crossover point is copied from the second parent and the rest is copied from the first parent

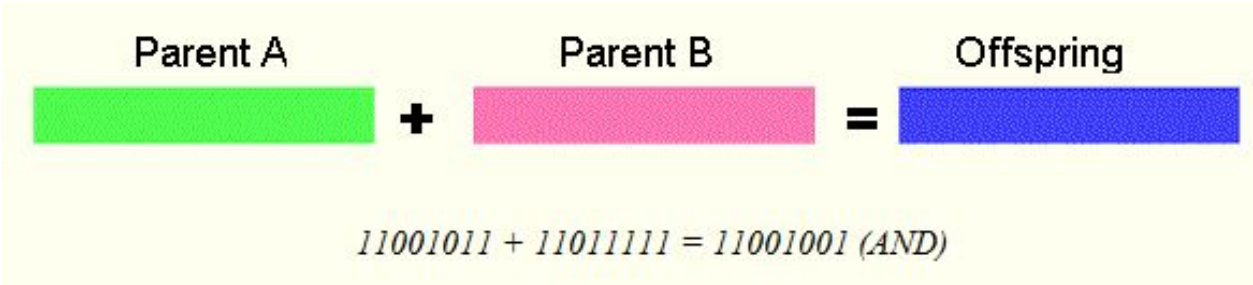


Genetic operators

- Crossover (recombination)
  - Uniform crossover: bits are randomly copied from the first or from the second parent

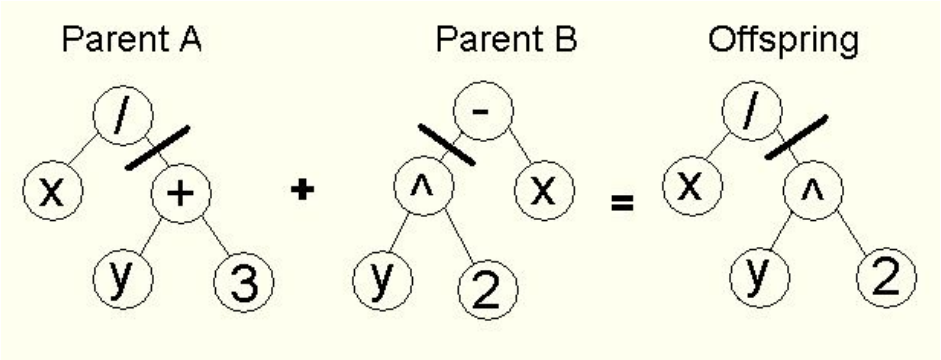


  - Arithmetic crossover : some arithmetic operation is performed to make a new offspring



Genetic operators

- Crossover (recombination)
  - Single point crossover: one crossover point is selected, till this point the permutation is copied from the first parent, then the second parent is scanned and if the number is not yet in the offspring it is added.
    Note: there are more ways how to produce the rest after crossover point
    $(123456789) + (453689721) = (123456897)$
  - Tree crossover : in both parent one crossover point is selected, parents are divided in that point and exchange part below crossover point to produce new offspring



Genetic Algorithm (GA) Cont'd

6. Selection:
   - To improve our populations overall fitness, selection helps us discard bad designs and only keep the best individuals.
   - There are a few different selection methods but the basic idea is the same, make it easier for fitter individuals to be selected for the next generation
   - Generational GA:
     the entire population is replaced at each iteration
   - Steady-state GA:
     just a few members are replaced at each generation

GA Cont'd

```
{
initialize node population;
evaluate node population;
while (TerminationCriteriaNotSatisfied)
{
select parent nodes for reproduction;
perform recombination and mutation;
evaluate population;
}
}
```

Some Issue to be Considered

- Choose basic implementation issues carefully:
  - Representation
  - Population size, mutation rate, ...
  - Selection, deletion policies
  - Crossover, mutation operators
  - Termination Criteria
  - Performance, scalability
  - How acceptable is the solution? It is tightly depends on choosing the evaluation function (often hardest part)

Advantages of Using GA

- The concept is easy to understand
- It is modular and separate from application
- It can supports multi-objective optimization
- It is good for noisy environments
- It always gives answer; answer gets better with time
- Since it is inherently parallel it is easily distributed
- There are multiple ways to speed up and improve a GA-based application as soon as knowledge about problem domain is gained
- Disadvantage of GAs They can be slower than some other methods. But with todays computers it is not so big problem.

When Using GA is Recommended?

- Alternate solutions are too slow or much complicated
- Need an exploratory tool to examine new approaches
- Problem is similar to one that has already been successfully solved.
- Want to hybridize with an existing solution

Some Applications of GA

- Optimization and design
  - Numerical optimization
  - circuit design
  - factory scheduling
  - drug design
  - travelling salesman
  - resource allocation
- Automatic programming
  - evolving computer programs (e.g., for image processing)
  - evolving cellular automata
- Adaptive control
  - robot navigation, trajectory planning
  - evolution of rules for solving expert problems
  - evolution of neural networks, improving classification algorithms, classifier systems

Some Useful References


http://www.obitko.com/tutorials/genetic-algorithms/.
Date of Acess: Jan, 2013.


http://www.theprojectspot.com/tutorial-post/creating-a-genetic-algorithm-for-beginners/3.
Date of Acess: Jan, 2013.


http://cgm.cs.mcgill.ca/~soss/cs644/projects/marko/.
Date of Acess: Jan, 2013.


W. Williams, "Gentetic algorithms: A tutorial,"