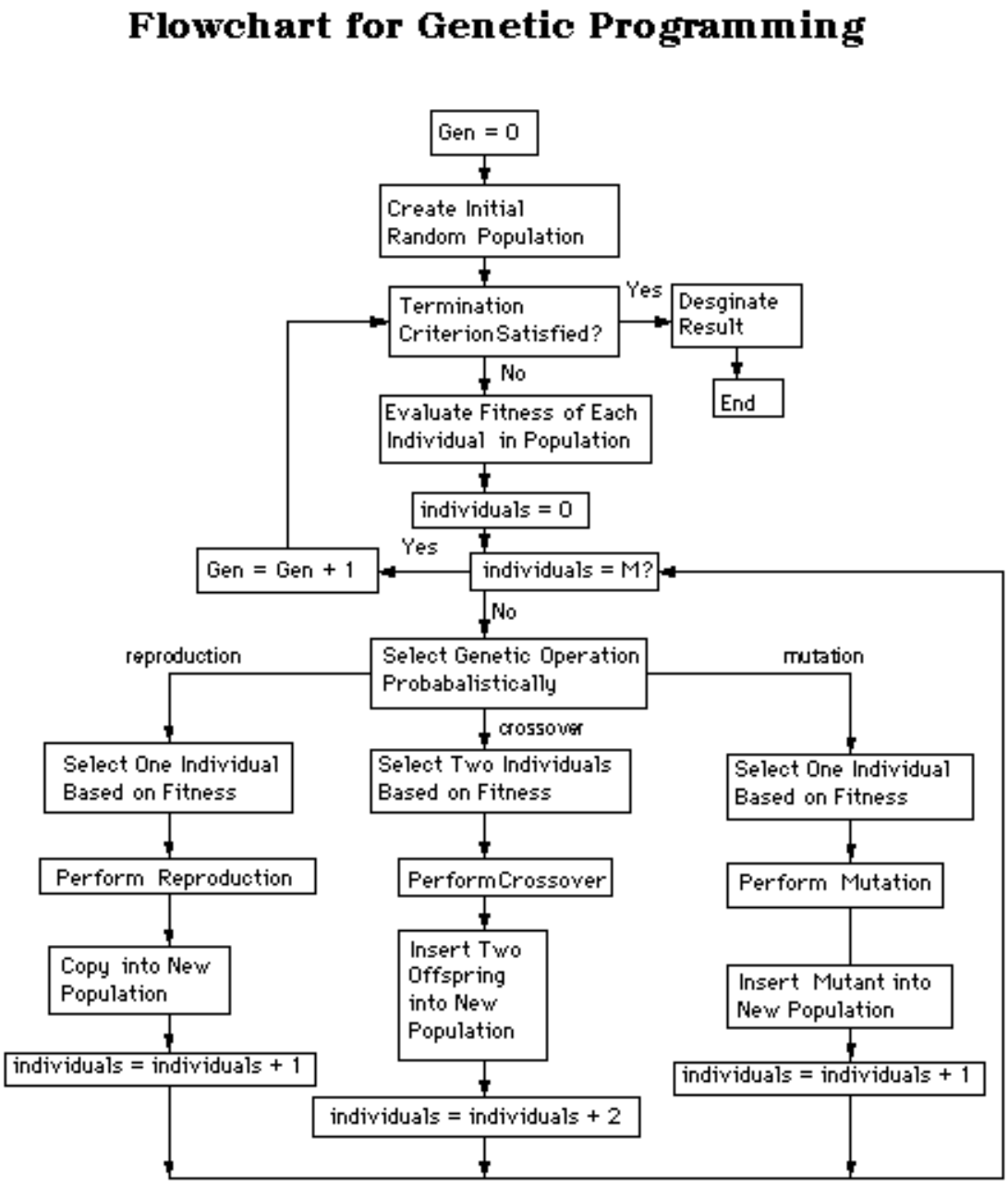
Introduction to
Evolutionary Programming
And Genetic Algorithms

General Algorithm for Genetic Algorithms



General Algorithm for Genetic Algorithms

Create a Random Initial Population

- An initial population is created from a **random** selection of solutions
- These solutions have been seen as represented by **chromosomes** as in living organisms
- The **genetic information** defines the behaviour of the individual
- A **chromosome** is a packet of genetic information organised in a standard way that defines completely and individual (solution)
- The genetic principles (way in which that information encodes the individual) enable the individuals to evolve in a given environment
- The genetic structure (way in which that information is packed and defined) enables the solutions to be manipulated
- The **genetic operands** (way in which that information can be manipulated) enables the solutions to reproduces and evolve

General Algorithm for Genetic Algorithms

Evaluate Fitness

- A value for **fitness** is assigned to each solution (chromosome) depending on how close it actually is to solving the problem

- Therefore we need to define the problem, model it, simulate it or have a data set as sample answers

- Each possible solution has to be tested in the problem and the answer evaluated (or marked) on how good it is

- The overall *mark* of each solution relative to all the *marks* of all solutions produces a fitness ranking

General Algorithm for Genetic Algorithms

Produce Next Generation

- Those chromosomes with a **higher fitness** value are more **likely** to reproduce offspring

- The population for the **next Generation** will be produced using the **genetic operators**

- Reproduction by **Copy** or **Crossing Over** and **Mutation** will be applied to the chromosomes according to the selection rule

- This rule states that the fitter and individual is, the higher the **probability** it has to reproduce

- Note that this works with **probabilities**!

- Why give a probability rather than choosing explicitly the best individuals?

General Algorithm for Genetic Algorithms

Next Generation or Termination

- If the population in the last generation contains a **solution** that produces an output that is close enough or equal to the desired answer then the problem has been solved.

- This is the ideal **termination criterion** of the evolution

- If this is not the case, then the new generation will go through the same process as their parents did, and the **evolution** will continue

- This will iterate until a solution is reached or another of the **termination criteria** is satisfied

- A termination criterion that **always must be included is Time-Out** (either as computing time or as number of generations evaluated)

- Since one drawback of Evolutionary Programming is that is very difficult (impossible most of the time) to know if the ideal termination criterion is going to be satisfied, or when

Evolutionary Programming

Difference between various names:

- What is a **Genetic Algorithm**?
  - Named by John Holland in the 70's.
  - A string of 1's and 0's to encode different solutions in the form of vectors of values or parameters.

- What is **Genetic Programming**?
  - Named by John Koza in the early 90's.
  - Evolving LISP programs using the GA principle.

- What is the **Genetic Paradigm**?
  - John Koza realised that not only programs could be "evolved" but other elements like equations, sets of rules, etc…
  - Then the application field exploded.

- What is **Evolutionary Programming**?
  - So… at the end, nearly any type of computing tool could be "evolved" in some way using the GA principles
  - The most generic term came out.