Genetic Algorithms

Russell & Norvig, Cha. 4.3

What is Evolutionary Computation?

An abstraction from the theory of biological evolution that is used to create optimization procedures or methodologies, usually implemented on computers, that are used to solve problems.

The Argument

Evolution has optimized biological processes;

therefore

Adoption of the evolutionary paradigm to computation and other problems can help us find optimal solutions.

Components of Evolutionary Computing

- Genetic Algorithms
  - invented by John Holland (University of Michigan) in the 1960's
- Evolution Strategies
  - invented by Ingo Rechenberg (Technical University Berlin) in the 1960's
- Started out as individual developments, but have begun to converge in the last few years

Natural Selection

- Limited number of resources
- Competition results in struggle for existence
- Success depends on fitness --
  - fitness of an individual: how well-adapted an individual is to their environment. This is determined by their genes (blueprints for their physical and other characteristics).
- Successful individuals are able to reproduce and pass on their genes

When changes occur ...

- Previously "fit" (well-adapted) individuals will no longer be best-suited for their environment
- Some members of the population will have genes that confer different characteristics than "the norm". Some of these characteristics can make them more "fit" in the changing environment.

Genetic Change in Individuals

- Mutation in genes
  - may be due to various sources (e.g. UV rays, chemicals, etc.)

Start:

1001001001001001001001

After Mutation:

1001000001001001001001

Location of Mutation

Genetic Change in Individuals

- Recombination (Crossover)
  - occurs during reproduction -- sections of genetic material exchanged between two chromosomes

Recombination (Crossover)

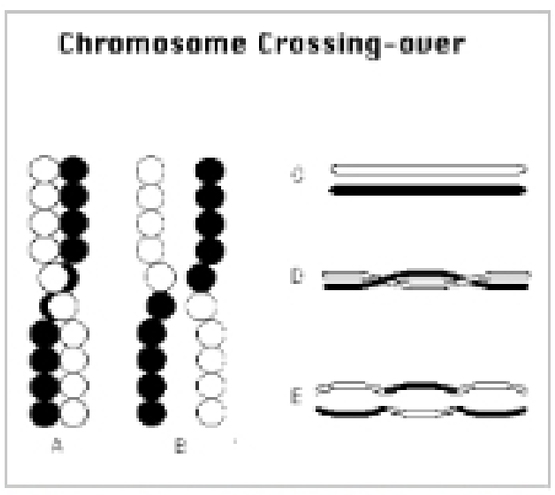


Image from http://esg-www.mit.edu:8001/bioing/meiosis.html

The Nature of Computational Problems

- Require search through many possibilities to find a solution
  - (e.g. search through sets of rules for one set that best predicts the ups and downs of the financial markets)
  - Search space too big -- search won't return within our lifetimes
- Require algorithm to be adaptive or to construct original solution
  - (e.g. interfaces that must adapt to idiosyncrasies of different users)

Why Evolution Proves to be a Good Model for Solving these Types of Problems

- Evolution is a method of searching for an (almost) optimal solution
  - Possibilities -- all individuals
  - Best solution -- the most "fit" or well-adapted individual
- Evolution is a parallel process
  - Testing and changing of numerous species and individuals occur at the same time (or, in parallel)
- Evolution can be seen as a method that designs new (original) solutions to a changing environment

The Metaphor

EVOLUTION		PROBLEM SOLVING
Individual	⟷	Candidate Solution
Fitness	⟷	Quality
Environment	⟷	Problem

Genetic Algorithms

- Closely follows a biological approach to problem solving
- A simulated population of randomly selected individuals is generated then allowed to evolve

Encoding the Problem

- Example: Looking for a new site which is closest to several nearby cities.
- Express the problem in terms of a bit string

z = (1001010101011100)

where the first 8 bits of the string represent the X-coordinate and the second 8 bits represent the Y-coordinate

Basic Genetic Algorithm

- Step 1. Generate a random population of *n* chromosomes
- Step 2. Assign a fitness to each individual
- Step 3. Repeat until *n* children have been produced
  - Choose 2 parents based on fitness proportional selection
  - Apply genetic operators to copies of the parents
  - Produce new chromosomes

Fitness Function

- For each individual in the population, evaluate its relative fitness
- For a problem with *m* parameters, the fitness can be plotted in an *m*+1 dimensional space

Sample Search Space

- A randomly generated population of individuals will be randomly distributed throughout the search space

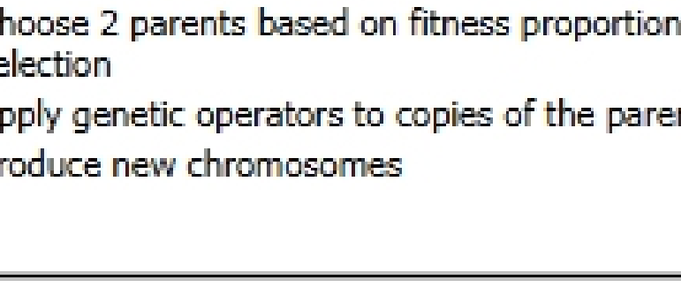
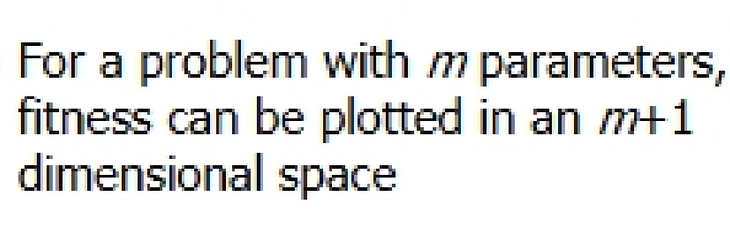


Image from http://www2.informatik.uni-erlangen.de/~jacob/Evolvica/Java/MultiModalSearch/rats.017/Surface.gif

Genetic Operators

- Cross-over



- Mutation