

The background of the slide features a dense pattern of vibrant green leaves, likely from a tree, with visible veins. At the bottom of the image, there are soft, circular ripples in a light blue-green color, suggesting water. The overall aesthetic is natural and fresh.

Genetic Algorithms: an Overview

Kennon Ballou

BarCamp Leipzig

May 2nd, 2008

Charles Darwin

- ✿ English naturalist
- ✿ b. 1809 d.1882
- ✿ Developed the theory of Natural Selection while traveling around the world on an expedition





Natural Selection

Natural selection is the process by which **favorable traits** that are **heritable** become more common in successive generations of a **population** of **reproducing** organisms, and unfavorable traits that are heritable become less common.

(wikipedia)



Important Terms

Important Terms

✿ Phenotype vs. Genotype

Important Terms

- ✿ Phenotype vs. Genotype
- ✿ Expression

Important Terms

- ✿ Phenotype vs. Genotype
- ✿ Expression
- ✿ Population

Important Terms

- ✿ Phenotype vs. Genotype
- ✿ Expression
- ✿ Population
- ✿ Fitness

Important Terms

- ✿ Phenotype vs. Genotype
- ✿ Expression
- ✿ Population
- ✿ Fitness
- ✿ Allele

Important Terms

- ✿ Phenotype vs. Genotype
- ✿ Expression
- ✿ Population
- ✿ Fitness
- ✿ Allele
- ✿ Selection / selected

Optimization Problems

Optimization Problems

✿ Find the **best** solution to a problem

Optimization Problems

- ✿ Find the **best** solution to a problem
- ✿ Solutions are 'individuals'

Optimization Problems

- ✿ Find the **best** solution to a problem
- ✿ Solutions are 'individuals'
- ✿ Black box algorithms

Optimization Problems

- ✿ Find the **best** solution to a problem
- ✿ Solutions are 'individuals'
- ✿ Black box algorithms
- ✿ Fitness function

Optimization Problems

- ✿ Find the **best** solution to a problem
- ✿ Solutions are 'individuals'
- ✿ Black box algorithms
- ✿ Fitness function
- ✿ Simplest: local search

Optimization Problems

- ✿ Find the **best** solution to a problem
- ✿ Solutions are 'individuals'
- ✿ Black box algorithms
- ✿ Fitness function
- ✿ Simplest: local search
- ✿ Local minima vs. global



Genetic Algorithms

Genetic Algorithms


- ✿ Popularized by John Holland in the 1960s

Genetic Algorithms

- ✿ Popularized by John Holland in the 1960s
- ✿ Form of local search




Basic Algorithm




Basic Algorithm

1. Start with initial population




Basic Algorithm

1. Start with initial population
2. Evaluate the fitness of each individual



Basic Algorithm

1. Start with initial population
2. Evaluate the fitness of each individual
3. Repeat...



Basic Algorithm

1. Start with initial population
2. Evaluate the fitness of each individual
3. Repeat...
 1. Select the best individuals to reproduce

Basic Algorithm

1. Start with initial population
2. Evaluate the fitness of each individual
3. Repeat...
 1. Select the best individuals to reproduce
 2. Breed or modify individuals to create next generation

Basic Algorithm

1. Start with initial population
2. Evaluate the fitness of each individual
3. Repeat...
 1. Select the best individuals to reproduce
 2. Breed or modify individuals to create next generation
 3. Evaluate the fitness of each individual

Basic Algorithm

1. Start with initial population
2. Evaluate the fitness of each individual
3. Repeat...
 1. Select the best individuals to reproduce
 2. Breed or modify individuals to create next generation
 3. Evaluate the fitness of each individual
 4. Replace worst ranked individuals

Basic Algorithm

1. Start with initial population
2. Evaluate the fitness of each individual
3. Repeat...
 1. Select the best individuals to reproduce
 2. Breed or modify individuals to create next generation
 3. Evaluate the fitness of each individual
 4. Replace worst ranked individuals
4. ...until termination

Representation



Representation

- ✿ Genotypic representation can determine your solution space!

Representation

- ✿ Genotypic representation can determine your solution space!
- ✿ Common: good ol' bit string



Illegal Solutions

Illegal Solutions

- ✿ Valid genotypes might express invalid phenotypes

Illegal Solutions

- ✿ Valid genotypes might express invalid phenotypes
- ✿ Different strategies:

Illegal Solutions

- ✿ Valid genotypes might express invalid phenotypes
- ✿ Different strategies:
 - ✿ Forbid invalid genotypes (expensive)

Illegal Solutions

- ✿ Valid genotypes might express invalid phenotypes
- ✿ Different strategies:
 - ✿ Forbid invalid genotypes (expensive)
 - ✿ Repairing (can lead to non-uniform distribution)

Illegal Solutions

- ✿ Valid genotypes might express invalid phenotypes
- ✿ Different strategies:
 - ✿ Forbid invalid genotypes (expensive)
 - ✿ Repairing (can lead to non-uniform distribution)
 - ✿ Penalty function (difficult to calibrate)



Selection Strategies

Selection Strategies

- ✿ How do we choose which individuals to reproduce?

Selection Strategies

- ✿ How do we choose which individuals to reproduce?
- ✿ Tournament selection is a common and easy method:

Selection Strategies

- ✿ How do we choose which individuals to reproduce?
- ✿ Tournament selection is a common and easy method:
 - ✿ Choose k individuals at random

Selection Strategies

- ✿ How do we choose which individuals to reproduce?
- ✿ Tournament selection is a common and easy method:
 - ✿ Choose k individuals at random
 - ✿ Evaluate fitness for individuals

Selection Strategies

- ✿ How do we choose which individuals to reproduce?
- ✿ Tournament selection is a common and easy method:
 - ✿ Choose k individuals at random
 - ✿ Evaluate fitness for individuals
 - ✿ Select the best one with probability $P \dots$



Operators

Operators

✿ Mutation

Operators

✿ Mutation

✿ Crossover

Mutation



Mutation

- ✿ Randomly mutate the genotype to try to avoid local minima

Mutation

- ✿ Randomly mutate the genotype to try to avoid local minima
- ✿ Bit string: flip each bit with probability M

Mutation

- ✿ Randomly mutate the genotype to try to avoid local minima
- ✿ Bit string: flip each bit with probability M
- ✿ M too high: too much chaos



Crossover

Crossover

- ✿ Simulates reproduction by combining existing individuals to create new individuals

Crossover

- ✿ Simulates reproduction by combining existing individuals to create new individuals
- ✿ Basically: genes gettin' it on!

Crossover

- ✿ Simulates reproduction by combining existing individuals to create new individuals
- ✿ Basically: genes gettin' it on!
- ✿ Common: one-point and two-point crossover



Diversity

Diversity

✿ Crossover requires diversity

Diversity

- ✿ Crossover requires diversity
- ✿ Mutation can increase diversity

Diversity

- ✿ Crossover requires diversity
- ✿ Mutation can increase diversity
- ✿ Niches

Replacement



Replacement

✿ Strong vs Weak:

Replacement

✿ Strong vs Weak:

✿ Strong exploits good solutions, but

Replacement

✿ Strong vs Weak:

- ✿ Strong exploits good solutions, but
- ✿ Weak allows exploration



Replacement Strategies

Replacement Strategies

✿ Replace random element - weak

Replacement Strategies

- ✿ Replace random element - weak
- ✿ Replace worst element - strong

Replacement Strategies

- ✿ Replace random element - weak
- ✿ Replace worst element - strong
- ✿ Replace one of the parents - weak

Replacement Strategies

- ✿ Replace random element - weak
- ✿ Replace worst element - strong
- ✿ Replace one of the parents - weak
- ✿ Replace only if offspring is better - strong

Genetic Programming



Genetic Programming

✿ Main proponent: John R. Koza

Genetic Programming

- ✿ Main proponent: John R. Koza
- ✿ Genotype is a program

Genetic Programming

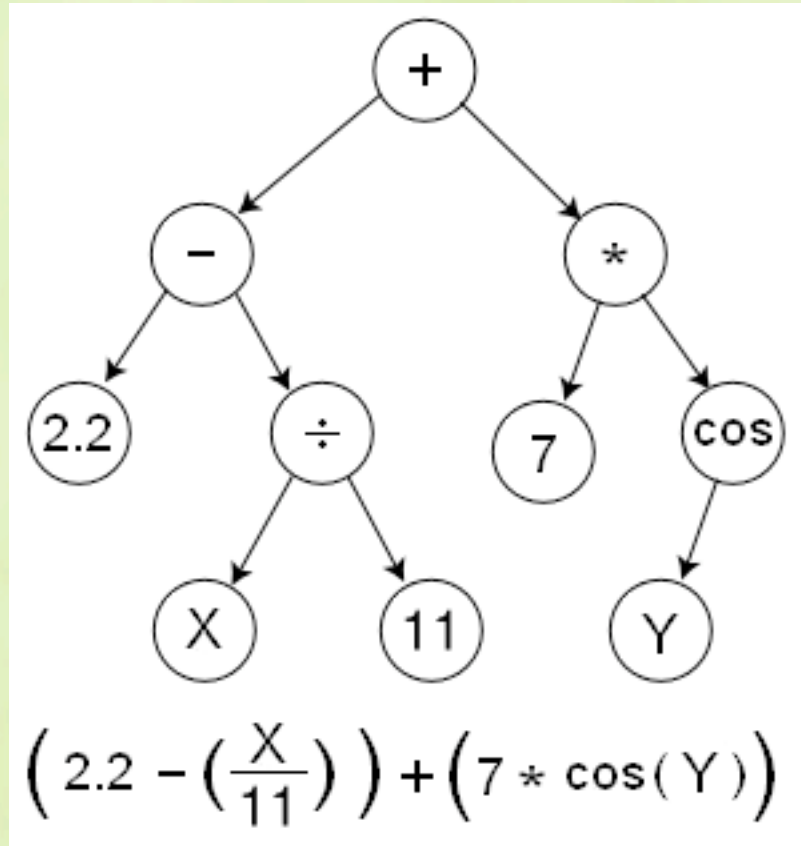
- ✿ Main proponent: John R. Koza
- ✿ Genotype is a program
- ✿ Main operations: mutation and crossover

Genetic Programming

- ✿ Main proponent: John R. Koza
- ✿ Genotype is a program
- ✿ Main operations: mutation and crossover
- ✿ Usually a tree-based representation

Tree Representation

- ✿ Usually in a language that has a natural tree structure such as Lisp





Tree Operations



Tree Operations

Mutation

-  Randomly generation information




Tree Operations

Mutation

-  Randomly generation information
-  Create new subtree




Tree Operations

Mutation

-  Randomly generation information
-  Create new subtree
-  Must take parity into account

Tree Operations




Mutation

-  Randomly generation information
-  Create new subtree
-  Must take parity into account

Crossover

Tree Operations

Mutation




-  Randomly generation information
-  Create new subtree
-  Must take parity into account

Crossover



-  Switch two subtrees

Tree Operations

Mutation

-  Randomly generation information
-  Create new subtree
-  Must take parity into account

Crossover

-  Switch two subtrees
-  Can drastically alter phenotype!



Moon Lander Example