GP

Genetic Programing

Jennifer Uhlmansiek

GP Overview

- What is Genetic Programing (GP)?
- Methodology
  - Creating Individuals and Populations
  - Testing Fitness
  - Selection and Genetic Operations
  - Control Parameters
- Issues with GP
- Summary

What is GP?

- Automatic Programing
  - Give a computer a problem and solve it on its own.
- Evolutionary algorithm
  - Uses Darwin's Theory of Evolution
    - Those with the greatest ability (or fitness) to survive pass their genetic material to the next generation
  - Survival of the fittest

GP Methodology

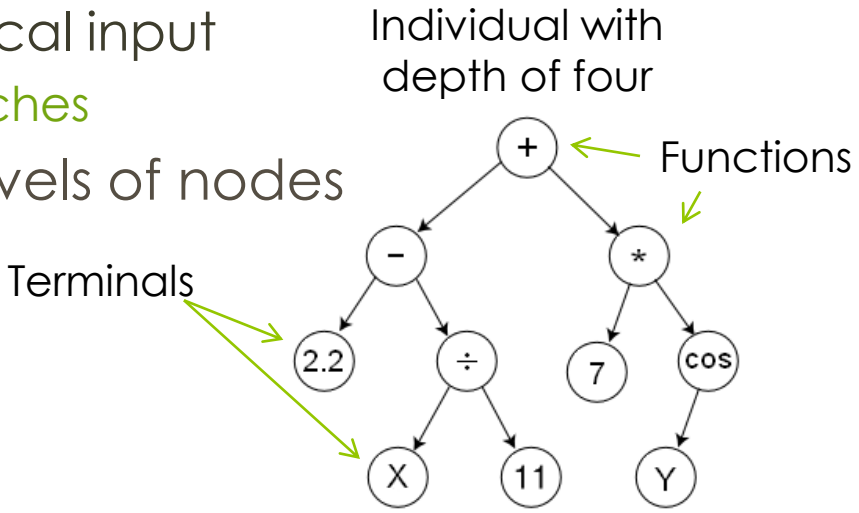
1. Start with a Problem
2. Create Individuals for a Population
   - Composed of genes (represent code)
3. Test Each Individual's Ability
   - Fitness Test (dependent on problem)
4. Select Individuals to Reproduce New Generation
5. Genetic Operations (for variation)
6. Multiple Generations for evolution
   - Final generation has ideally evolved into a exceptional solution

Problem: Robot Soccer

- Evolve a robot soccer team into World Cup Champions
- Need to be able to follow the ball
- Use GP to evolve programming code for good ball following.

Creating Individuals

- Made up of genes (i.e. code)
  - Genes:
    1. Functions—provide argument for a function
       - Nodes with branches
    2. Terminals—numerical input
       - Nodes without branches
  - Depth—number of levels of nodes



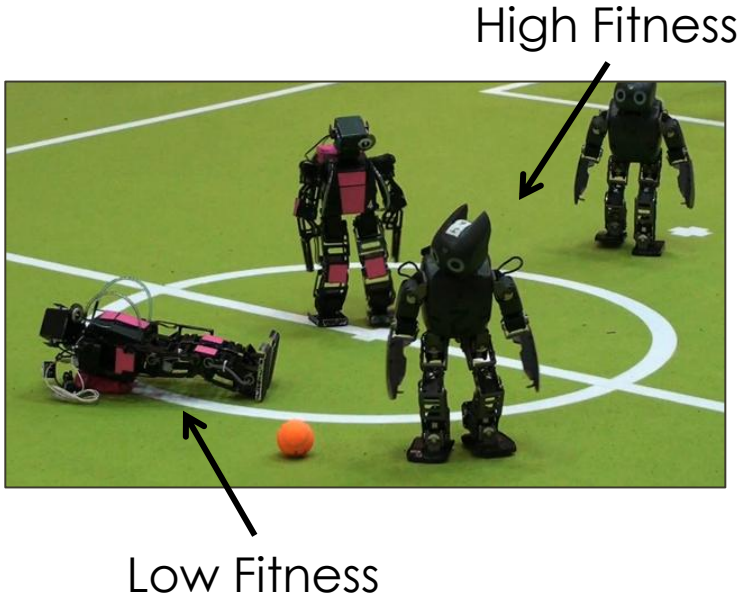
"Genetic" Code → $\left(2.2-\left(\frac{X}{11}\right)\right)+\left(7*\cos(Y)\right)$

Creating a Population

- For evolution to occur need a large population
- Need a variation of individuals
  - Several techniques in GP
    1. Grow—create individuals with max., *m*, depth
    2. Full—create individuals with final, *d*, depth
    3. Ramped Half-and-Half—both Grow and Full, *md*

Test Individuals' Ability

- Fitness test
  - Dependent on the problem
- Example robot soccer:
  - Robot's ability to follow the ball



Select Individuals to Reproduce

- Need to transfer individual's code to the next generation
  - Want a new and better generation of robots
- Different methods fitness of selection
  - Overall picked based on fitness

Genetic Operations

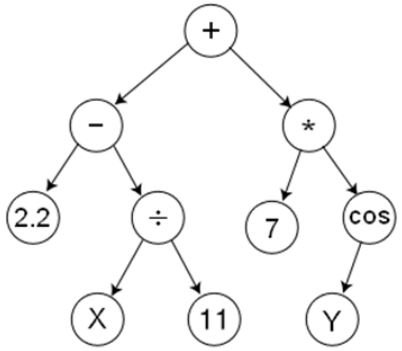
- Ways to change up and transfer individual's code to the next generation
  1. Reproduction
  2. Crossover
  3. Mutation
  4. Others: editing, permutation, encapsulation, and decimation.

- Important to have variation to allow for evolution!

Genetic Operations

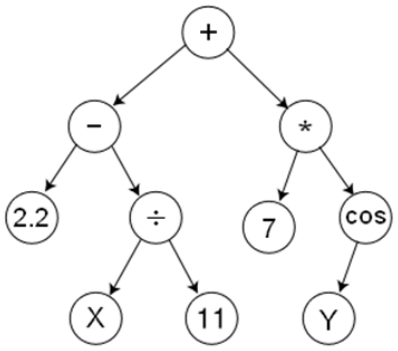
- Reproduction—Individuals selected are sent to next generation

First Generation



$$\left(2.2 - \left(\frac{X}{11}\right)\right) + \left(7 * \cos(Y)\right)$$

Second Generation



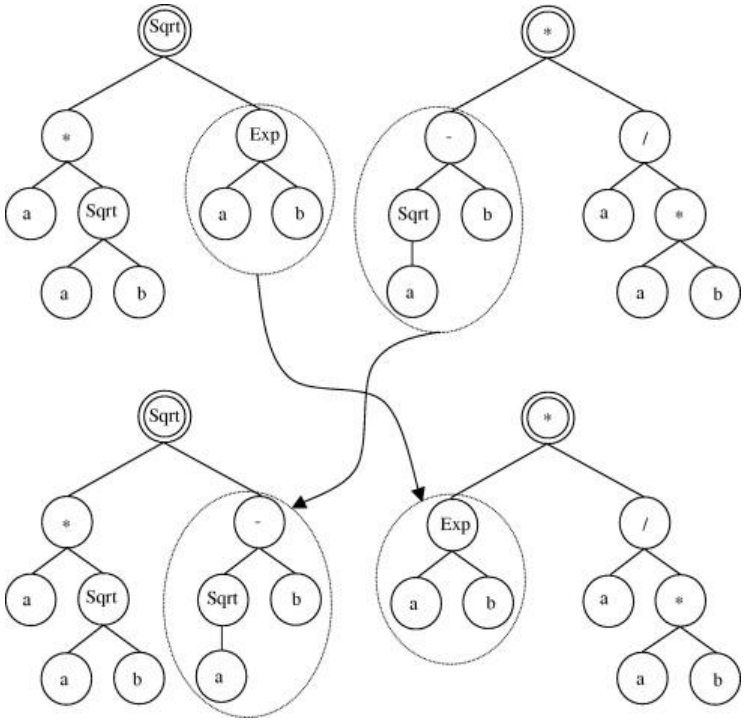
$$\left(2.2 - \left(\frac{X}{11}\right)\right) + \left(7 * \cos(Y)\right)$$

Genetic Operations

- Crossover—two individuals swap pieces of code and make two new individuals.

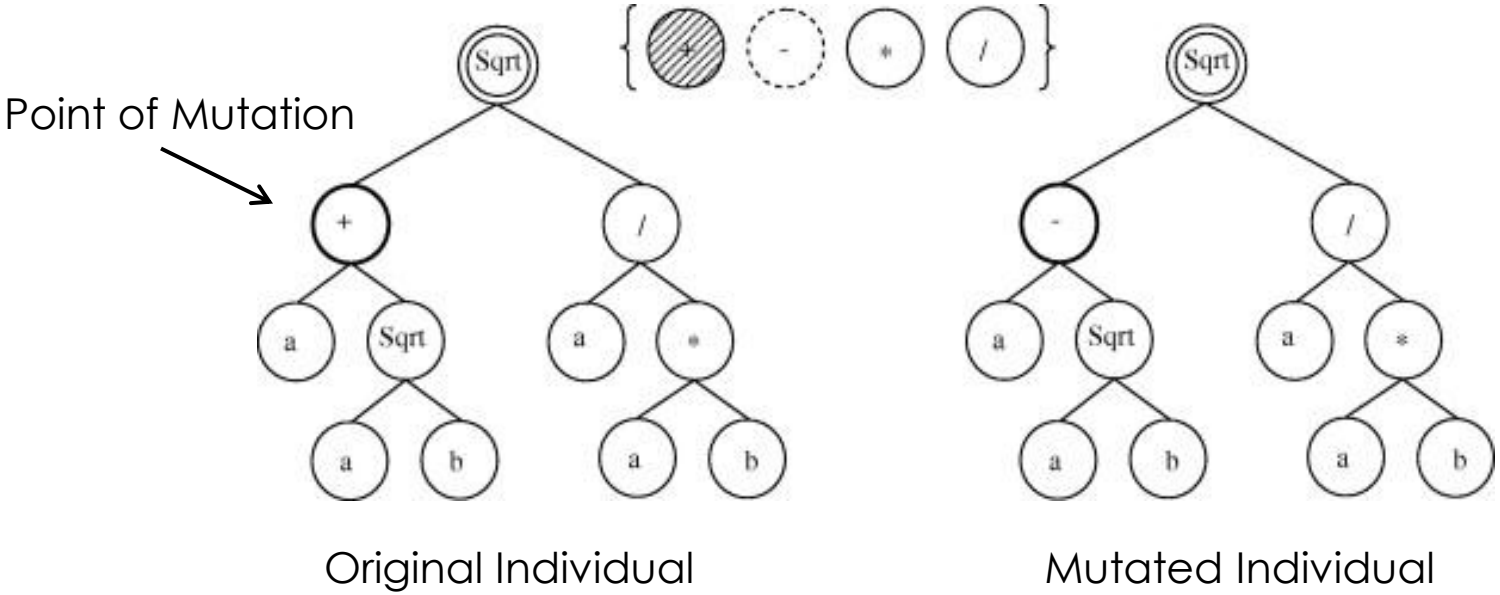
Parent Generation

New Generation (Children)



Genetic Operations

- Mutation—random point in code is deleted and replaced making new code



Original Individual

Mutated Individual

Generations

- Repeat genetic operations of fit individuals over time.
  - Overall fitness of population increases over time.
- Ideally, last generation will give us World Cup Champions!

Control Parameters

- Population size
- Maximum number of generations
- Probability of genetic operations
- Probability of selection and reproduction

- Not too constrictive—can't evolve!
- Not too laissez-faire—can evolve too fast or not enough for quality evolution to occur!

Issues with GP

- Massive computer power needed.
  - Mostly in computing fitness
- “Genetic” code is vital for solving the problem.
- Can be persnickety with control parameters.
- Long run times.

Summary

- GP uses methods from evolution to solve problems
- Evolve populations of individuals comprised of code
- Goal is over generations a fit population or excellent solution will evolve

References

(n.d.). Retrieved October 2012, from Technology and Science: http://www.slatev.com/channel_slider/technology-and-science/

(n.d.). Retrieved October 2012, from Robot: http://wilmarleonardo.blogspot.com/

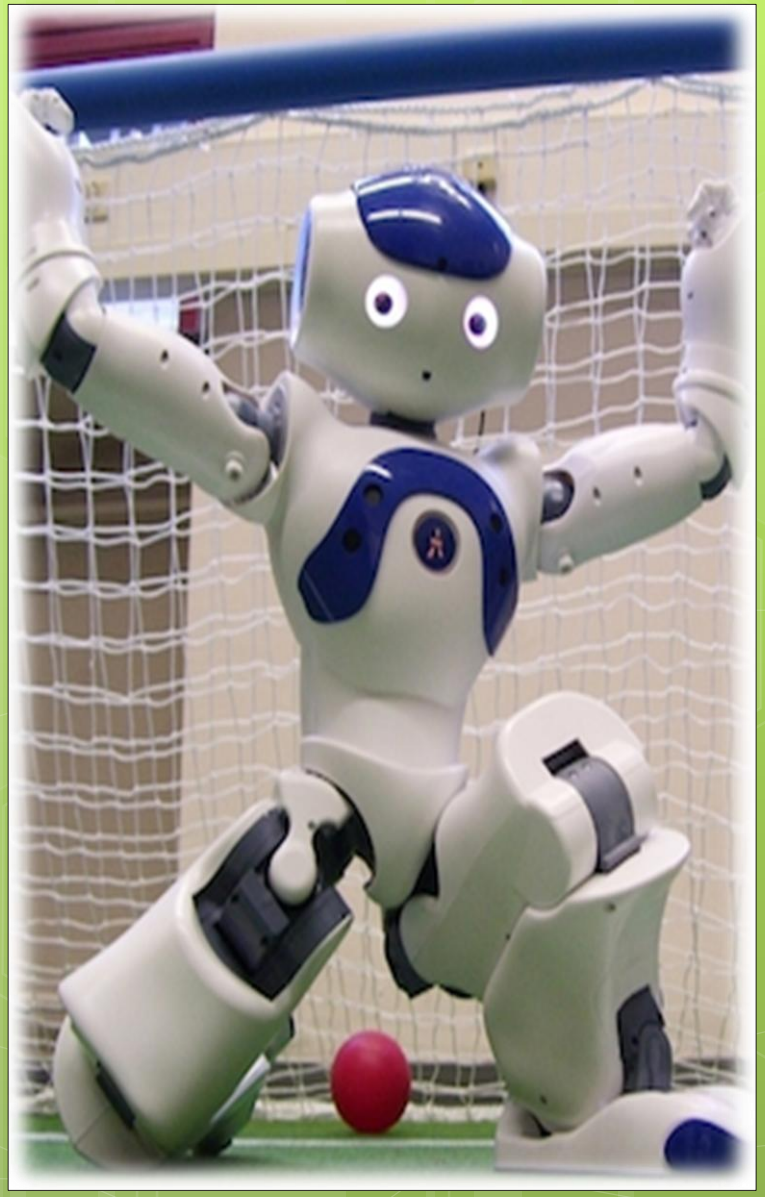
Students and robots rule in RoMeLa. (2008, June 30). Retrieved October 2012, from Virginia Tech: http://www.vt.edu/spotlight/achievement/2008-06-30_romela/2008-06-30_romela.html

File:Genetic Program Tree.png. (2012, October 3). Retrieved October 2012, from Wikipedia: http://en.wikipedia.org/wiki/File:Genetic_Program_Tree.png

Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). *Genetic Programming: An Introduction*. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Sarıdemir, M. (2010, October). Genetic programming approach for prediction of compressive strength of concretes containing rice husk ash. *Construction and Building Materials, 24*(10), 1911–1919.

Walker, M. (2007, October 7). *Introduction to Genetic Programming*. Retrieved October 2012, from http://www.cs.montana.edu/~bwall/cs580/introduction_to_gp.pdf



Thank you!

Questions?