Genetic Programming
Agenda

- What is Genetic Programming?
- Background/History.
- Why Genetic Programming?
- How Genetic Principles are Applied.
- Examples of Genetic Programs.
- Future of Genetic Programming.
What is Genetic Programming (GP)?

ROBOTICS

Machine learning

evolutionary

Artificial Intelligence

GP  ES  EP  GA
Genetic Algorithms

- Most widely used
- Robust
- uses 2 separate spaces
  - search space - coded solution (genotype)
  - solution space - actual solutions (phenotypes)

Genotypes must be mapped to phenotypes before the quality or fitness of each solution can be evaluated
Evolutionary Strategies

- Like GP no distinction between search and solution space
- Individuals are represented as real-valued vectors.
- Simple ES
  - one parent and one child
  - Child solution generated by randomly mutating the problem parameters of the parent.
- Susceptible to stagnation at local optima
Evolutionary Strategies (cont’d)

- Slow to converge to optimal solution
- More advanced ES
  - have pools of parents and children
- Unlike GA and GP, ES
  - Separates parent individuals from child individuals
  - Selects its parent solutions deterministically
Evolutionary Programming

- Resembles ES, developed independently
- Early versions of EP applied to the evolution of transition table of finite state machines
- One population of solutions, reproduction is by mutation only
- Like ES operates on the decision variable of the problem directly (ie **Genotype = Phenotype**)
- **Tournament selection of parents**
  - better fitness more likely a parent
  - children generated until population doubled in size
  - everyone evaluated and the half of population with lowest fitness deleted.
General Architecture of Evolutionary Algorithms

- **Initialise**: random (coded) values to genotypes
- **Map**: genotypes to phenotypes
- **Evaluate**: phenotypes to fitness values
- **Select**: fitness values to genotypes
- **Fertility**: genotypes, fitness values to fertility values
- **Reproduce**: parent genotypes, fertility values to child genotypes via genetic operators
- **Replace**: genotypes to genotypes
- **Kill**: genotypes are deleted
- **Move**: population migrate/inject genotypes
- **Terminate?**: fitnesses / generations to halt evolution

**EMRYOLOGY**

**FITNESS**

**SELECTION**

**HERITABILITY & VARIATION**

**SPECIATION**
Genetic Programming

- Specialized form of GA
- Manipulates a very specific type of solution using modified genetic operators
- Original application was to design computer program
- Now applied in alternative areas eg. Analog Circuits
- Does not make distinction between search and solution space.
- Solution represented in very specific hierarchical manner.
By John R. Koza, Stanford University.


Combining the idea of machine learning and evolved tree structures.
Why Genetic Programming?

- It saves time by freeing the human from having to design complex algorithms.
- Not only designing the algorithms but creating ones that give optimal solutions.
- Again, Artificial Intelligence.
What Constitutes a Genetic Program?

- Starts with "What needs to be done"
- Agent figures out "How to do it"
- Produces a computer program - “Breeding Programs”
- Fitness Test
- Code reuse
- Architecture Design - Hierarchies
- Produce results that are competitive with human produced results
How are Genetic Principles Applied?

- “Breeding” computer programs.
- Crossovers.
- Mutations.
- Fitness testing.
Computer Programs as Trees

- Infix/Postfix
- \((2 + a) \times (4 - \text{num})\)
‘Breeding’ Computer Programs

- Start off with a large “pool” of random computer programs.
- Need a way of coming up with the best solution to the problem using the programs in the “pool”
- Based on the definition of the problem and criteria specified in the fitness test, mutations and crossovers are used to come up with new programs which will solve the problem.
The Fitness Test

- Identifying the way of evaluating how good a given computer program is at solving the problem at hand.
- How good can a program cope with its environment.
- Can be measured in many ways, i.e. error, distance, time, etc…
Fitness Test Criteria

- Time complexity a good criteria.
  - i.e. $n^2$ vs. $n \log n$.
- Accuracy - Values of variables.
- Combinations of criteria may also be tested.
Mutations in Nature

**Properties of mutations**

- Ultimate source of genetic variation.
- **Radiation, chemicals** change genetic information.
- Causes new genes to be created.
- One chromosome.
- Asexual.
- Very rare.

Before:
acgtactggctaa

After:
acatactggctaa
Mutations in Programs

- Single parental program is *probabilistically selected* from the population based on fitness.

  - Mutation point randomly chosen.
    - the subtree rooted at that point is deleted, and
    - a **new subtree is grown** there using the same random growth process that was used to generate the initial population.

- Asexual operations (mutation) are typically performed sparingly:
  - with a *low probability* of,
  - probabilistically selected from the population based on fitness.
Two parental chromosomes exchange part of their genetic information to create new hybrid combinations (recombinant).

- No loss of genes, but an exchange of genes between two previous chromosomes.
- No new genes created, preexisting old ones mixed together.

Genetic recombination as a result of crossing over.
Crossovers in Programs

- Two *parental programs* are selected from the population based on fitness.
- A crossover point is randomly chosen in the first and second parent.
  - The first parent is called *receiving*
  - The second parent is called *contributing*
- The *subtree* rooted at the crossover point of the first parent is deleted
- It is replaced by the subtree from the second parent.
- **Crossover** is the *predominant operation* in genetic programming (and genetic algorithm) research
- It is performed with a high probability (say, 85% to 90%).
Examples of Genetic Programs

1. Symbolic Regression -
   - the process of discovering:
     - the functional form of a target function
     - and all of its necessary coefficients,
     - or at least an approximation to these.

2. Analog circuit design
   - Embryo circuit is an initial circuit which is modified to create a new circuit according to functionality criteria.
Genetic Programming in the Future

- Speculative.
- Only been around for 8 years.
- Is very successful.
- Discovery of new algorithms in existing projects.
Summary

- Field of study in Machine Learning.
- Save time while creating better programs.
- Based on the principles of genetics.
- Symbolic Regression/Circuit Design.
- Future uncertain.
End of Show

Oh yeah. Hm hm yeah yeah hm.
It sucked.

Shut up Buttmunch.
That sucked.
Sources

CPSC 533, Artificial Intelligence

- Dan Kiely
- Ran Shoham
- Brent Heigold