A big Q in A.I.

"How can computers learn to solve problems without being explicitly programmed?"

—Arthur Samuel (1959)
Outline

- A Little History, and a Little Biology
- Genetic Programming
  - What is it?
  - How does it work?
- Demo
- Real-world GP
The GP Guide to Evolution

Genetic Programming!

Charles Darwin
“The Code is Alive”
Outline

- A Little History, and a Little Biology

- Genetic Programming
  - What is it?
  - How does it work?

- Demo

- Real-world GP
What is GP?

- A domain-independent method that genetically breeds a population of computer programs to **solve a problem**.
- A stochastic searching mechanism.
- An invention creating machine.
- Fun.
Basic Steps of GP

1. Populate Generation 0 with random computer programs.
2. Evaluate how “fit” each program is.
3. Is the “best-so-far” program “good enough”? 
4. Use the “fitness” values to determine which programs will produce offspring, to create the next generation.
5. Output the “good enough” program.
Preparation

- Program Ingredients
  - Functions and Terminals

- Fitness Measure
  - How good does our program work?
An example

- Symbolic Regression
  - Let’s find a curve that fits the data we give it as closely as possible.
Symbolic Regression

Given: Some points

Find: \( f(x) = x^2 + x + 1 \)

A “program” is a math function.

Terminal set
\[ T = \{ x, \text{random constants between -5.0 and 5.0} \} \]

Function set
\[ F = \{ +, -, *, \% \text{ (protected div.)} \} \]
(not modulo!)
Into the DNA Forest…

A program is just a tree.

Lisp

(a) 
\[
\begin{align*}
\text{\textbf{(x + 1) - 0}} \\
\text{\textbf{(- (+ x 1) 0)}}
\end{align*}
\]

(b) 
\[
\begin{align*}
\text{\textbf{1 + (x * x)}} \\
\text{\textbf{(+ 1 (* x x))}}
\end{align*}
\]

(c) 
\[
\begin{align*}
\text{\textbf{2 + 0}} \\
\text{\textbf{(+ 2 0)}}
\end{align*}
\]

(d) 
\[
\begin{align*}
\text{\textbf{x * (-1 - (-2))}} \\
\text{\textbf{(* x (- -1 -2))}}
\end{align*}
\]

Generation 0  Pop. Size = 4
Fitness Measure

Looking for error < .10

(a) \[ x + 1 \]

(b) \[ x^2 + 1 \]

(c) \[ 2 \]

(d) \[ x \]

.67 1.00 1.70 2.67
The birds and the bytes...

- 3 Fundamental Genetic Operations
  - (Asexual) Reproduction
  - Mutation
  - Recombination (Crossover)
(Asexual) Reproduction

Fitness = .67

Generation 0

(a)
- + 0
x 1
x + 1

COPY

Generation 1

(a)
- + 0
x 1
x + 1

Fitness = .67
Mutation

**Generation 0**

\[
\begin{array}{c}
(c) \\
+ \\
2 \\
0 \\
2
\end{array}
\]

Fitness = 1.70

**Generation 1**

\[
\begin{array}{c}
+ \\
\%
\end{array}
\]

Mutate

\[
\begin{array}{c}
+ \\
0 \\
x \\
x \\
1
\end{array}
\]

Fitness = ?
Recombination (Crossover)

Generation 0

(a) 
- 
+ 
0 
1 
x 
x + 1 

(b) 
+ 
* 
1 
* 
x 
x 

x^2 + 1

Generation 1

(c) 
- 
x 
0 
x 

(d) 
+ 
* 
1 
* 
x 
x 
x^2 + x + 1
Gen 0 $\rightarrow$ Gen 1

Crossover

Copy

Copy + Mutate

~10%

~90%

< 1%
A “Toy Problem”

- Contrived.
- Small population.
  - Real populations often 500 or more
- Exact answer in Generation 1.
  - Real runs take 50 or more generations
Interesting questions.

- **Q:** What kind of sex is happening?
  - **A:** It’s random. (1%, 10%, 90%)

- **Q:** Who gets to be involved?
  - **A:** It’s random, but more “fit” programs have a better chance.

- **Q:** And why?
  - **A:** That’s life. (in the real biological world).
  - **A:** More importantly, it seems to work.
Practical Considerations

- Functions and Terminals
  - Are they sufficient to solve the problem?
  - Are they useful?
- Closure
  - What comes out must go in.
  - Protected operators (e.g. %, sqrt, log)
  - Types: booleans, integers, strings, etc...
How are trees created?

- **Full**
  - Terminals only on lowest level.

- **Grown**
  - Random depths

- **Ramped Half & Half**
  - ½ pop. full, ½ pop. grown
  - Builds trees with multiple depths
Fitness Functions

- Raw fitness
  - How close is our curve to the data?
  - How did we do on the training set?
  - How many times did our Go AI win?

- For better performance, you may need to scale/normalize the fitness function...
  - (or use “tournament selection”.)
Outline

- A Little History, and a Little Biology
- Genetic Programming
  - What is it?
  - How does it work?
- Demo
- Real-world GP
Mazewalker Demo

- **Terminal Set**
  - \( T = \{ \text{turn-right, turn-left, do-nothing, wall-right?, wall-left?, wall-ahead?} \} \)

- **Function Set**
  - \( F = \{ \text{IFELSE}\} \)
  - \( \text{IFELSE (condition?)} \)
    - [ do-something ]
    - [ do-other]
Mazewalker Demo
Code Bloat vs. Parsimony

- Parsimony
  - Is it always good?

- Introns

- Editing

Bloat:

\[ f(x) = 12 + 0 \times \left(5 + \frac{200}{x}\right) \]
Outline

- A Little History, and a Little Biology
- Genetic Programming
  - What is it?
  - How does it work?
- Demo
- Real-world GP
  - “Human Competitive”?
Sorting Network for 7 items

Only 16 steps necessary -- Floyd and Knuth, several years later

Rediscovered – GP, 1999
Balance/Unbalance Circuit

Sang Gug Lee, patented 2001
Rediscovered -- GP, 2002
Tuning Rules for a PID Controller

Ziegler-Nichols tuning rules, 1942 (industry standard)


GP evolved BETTER rules. 2003 (patent pending)
GP takes over the world?

- Will genetic programming replace human programmers?
  - Should we expect “Windows DNA”?

- Software: engineered vs. evolved?

  Intelligence required to pose the right problem...

  Discuss pros & cons.
When is GP the right tool?

- **Desire approximate solutions**
  - Circuit design, process control, bioinformatics, classification, data mining, forecasting

- **Areas human programmers are weak**
  - Parallel computers, cellular automata, multi-agent strategies, field-programmable gate arrays, digital signal processors, swarm intelligence.

- **You know what you want, but not how.**
  - IMPORTANT: Need a good fitness function
Non-tree GP

Linear Structures

(PDP 8 Assembly Language)

Graph Structures

← Finite State Automata
Source Credits

  - The artificial ant example.
- [www.genetic-programming.org](http://www.genetic-programming.org)
  - The symbolic regression example.
More Source Credits


Image Credit Sources

http://www.elitestaffingsvc.com/images/DNA.purple.jpg
http://www.sp.uconn.edu/~gage/Media/1-DNA%20structure.JPG
http://lcvmsun9.epfl.ch/DV/DVgallery/computer.gif
http://genetics.nbii.gov/images/evolution5.gif
http://www.alumni.ca/~laued3e/chromosome.jpg
http://www.mahoroba.ne.jp/~gonbe007/gif/dinosaur.gif
http://www.law.umkc.edu/faculty/projects/ftrials/conlaw/darwin.gif
http://www.raindrop.org/rugrat/fun/computer.gif
http://classes.kumc.edu/sah/clls705/images/genetics/flower.gif
http://www.cityhillmiddleschool.com/jellyfish.gif
http://www.nald.ca/fulltext/Clo/images/couple.gif
http://sfr.ee.teiath.gr/htmSELIDES/Art/Photos/Kaut/images/008mutation.jpg
http://www.itg.uiuc.edu/publications/techreports/98-005/images/image01-right.gif
http://web.syr.edu/~nrsmit01/webquest/ant.jpg
http://asab.icapb.ed.ac.uk/exercises/gcse/images/ant.gif
http://filebox.vt.edu/org/veccs/cookie.jpg
http://www.lotek.com/images/turtle.jpg

(in order of appearance)