Outline

- Evolutionary computation
- Prototypical GA
- An example: GABIL
- Genetic Programming
- Individual learning and population evolution

Evoluationary Computation

1. Computational procedures patterned after biological evolution
2. Search procedure that probabilistically applies search operators to set of points in the search space

Biological Evolution

Lamarck and others:

- Species “transmute” over time

Darwin and Wallace:

- Consistent, heritable variation among individuals in population
- Natural selection of the fittest

Mendel and genetics:

- A mechanism for inheriting traits
- genotype $\rightarrow$ phenotype mapping

GA($Fitness, Fitness\_threshold, p, r, m$)

- *Initialize:* $P \leftarrow p$ random hypotheses
- *Evaluate:* for each h in P, compute $Fitness(h)$
- While $[\max_h Fitness(h)] < Fitness\_threshold$
  1. *Select:* Probabilistically select $(1-r)p$ members of P to add to P_S.
  $$\Pr(h_i) = \frac{Fitness(h_i)}{\Sigma_{j=1}^{p} Fitness(h_j)}$$
  2. *Crossover:* Probabilistically select $\frac{r \cdot p}{2}$ pairs of hypotheses from P. For each pair, $\langle h_1, h_2 \rangle$, produce two offspring by applying the Crossover operator. Add all offspring to P_s.
  3. *Mutate:* Invert a randomly selected bit in $m \cdot p$ random members of P_s
  4. *Update:* $P \leftarrow P_s$
  5. *Evaluate:* for each h in P, compute $Fitness(h)$
- Return the hypothesis from P that has the highest fitness.

Representing Hypotheses

Represent

$$(Outlook = Overcast \vee Rain) \wedge (Wind = Strong)$$

by

$$\begin{array}{cc} Outlook & Wind \\ 011 & 10 \end{array}$$

Represent

$$\text{IF} \quad Wind = Strong \quad \text{THEN} \quad PlayTennis = yes$$

by

$$\begin{array}{ccc} Outlook & Wind & PlayTennis \\ 111 & 10 & 10 \end{array}$$

Operators for Genetic Algorithms

	Initial strings	*Crossover Mask*	*Offspring*
Single-point crossover:	11101001000	11111000000	11101010101
	00001010101		00001001000
Two-point crossover:	11101001000	00111110000	11001011000
	00001010101		00101000101
Uniform crossover:	11101001000	10011010011	10001000100
	00001010101		01101011001
Point mutation:	11101001000		11101011000

Selecting Most Fit Hypotheses

Fitness proportionate selection:

$$\Pr(h_i) = \frac{Fitness(h_i)}{\Sigma_{j=1}^{p} Fitness(h_j)}$$

... can lead to *crowding*

Tournament selection:

- Pick h_1, h_2 at random with uniform prob.
- With probability p, select the more fit.

Rank selection:

- Sort all hypotheses by fitness
- Prob of selection is proportional to rank

GABIL [DeJong et al. 1993]

Learn disjunctive set of propositional rules, competitive with C4.5

Fitness:

$$Fitness(h) = (correct(h))^2$$

Representation:

IF $a_1 = T \wedge a_2 = F$ THEN $c = T$; IF $a_2 = T$ THEN $c = F$

represented by

$$\begin{array}{ccccccc} a_1 & a_2 & c & & a_1 & a_2 & c \\ 10 & 01 & 1 & & 11 & 10 & 0 \end{array}$$

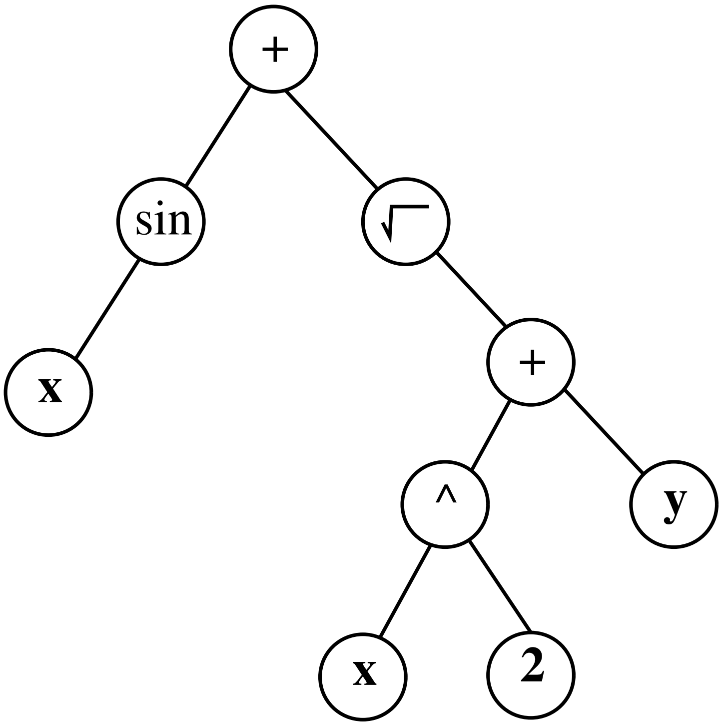
Genetic operators: ???

- want variable length rule sets
- want only well-formed bitstring hypotheses

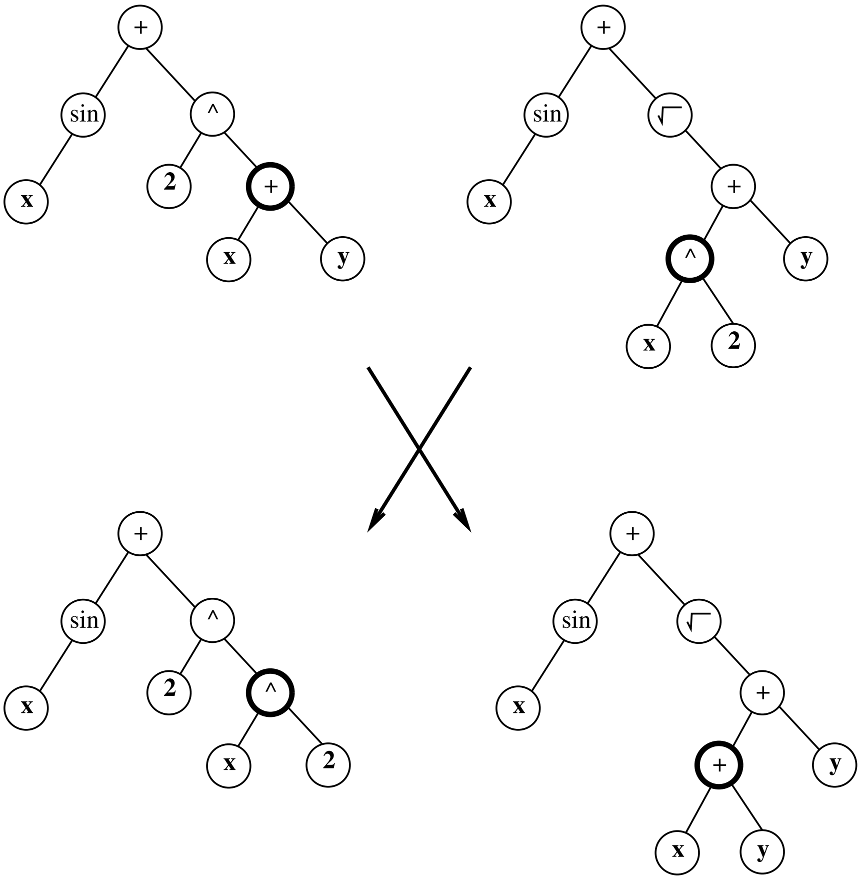
Genetic Programming

Population of programs represented by trees

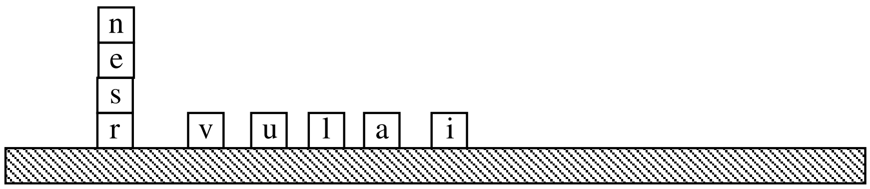
$$\sin(x) + \sqrt{x^2 + y}$$



Crossover



Block Problem



Goal: spell UNIVERSAL

Terminals:

- CS ("current stack") = name of the top block on stack, or F.
- TB ("top correct block") = name of topmost correct block on stack
- NN ("next necessary") = name of the next block needed above TB in the stack

Primitive functions:

- (MS x): ("move to stack"), if block x is on the table, moves x to the top of the stack and returns the value T. Otherwise, it does nothing and returns the value F.
- (MT x): ("move to table"), if block x is somewhere in the stack, moves the block at the top of the stack to the table and returns the value T. Otherwise, it returns the value F.
- (EQ x y): ("equal"), which returns T if x equals y, and returns F otherwise.
- (NOT x): which returns T if $x = F$, and returns F if $x = T$
- (DU x y): ("do until") which executes the expression x repeatedly until expression y returns the value T

Learned Program

Trained to fit 166 test problems

Using population of 300 programs, found this after 10 generations:

(EQ (DU (MT CS)(NOT CS)) (DU (MS NN)(NOT NN)))

More interesting example: design electronic filter circuits

- Individuals are programs that transform begining circuit to final circuit, by adding/subtracting components and connections
- Use population of 640,000, run on 64 node parallel processor
- Discovers circuits competitive with best human designs