Evolutionary 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 → phenotype mapping
\[ \text{GA}(\text{Fitness}, \text{Fitness\_threshold}, p, r, m) \]

- \textbf{Initialize:} \( P \leftarrow p \) random hypotheses
- \textbf{Evaluate:} for each \( h \) in \( P \), compute \( \text{Fitness}(h) \)
- \textbf{While} \( \left[ \max_h \text{Fitness}(h) \right] < \text{Fitness\_threshold} \)
  1. \textbf{Select:} Probabilistically select \((1 - r)p\)
      members of \( P \) to add to \( P_s \).
      \[ \text{Pr}(h_i) = \frac{\text{Fitness}(h_i)}{\sum_{j=1}^{p} \text{Fitness}(h_j)} \]
  2. \textbf{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. \textbf{Mutate:} Invert a randomly selected bit in \( m \cdot p \) random members of \( P_s \)
  4. \textbf{Update:} \( P \leftarrow P_s \)
  5. \textbf{Evaluate:} for each \( h \) in \( P \), compute \( \text{Fitness}(h) \)
- \textbf{Return the hypothesis from} \( P \) \textbf{that has the highest fitness.}
Representing Hypotheses

Represent

\[(Outlook = Overcast \lor Rain) \land (Wind = Strong)\]

by

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

Represent

IF \( Wind = Strong \) THEN \( PlayTennis = yes \)

by

\[
\begin{array}{ccc}
Outlook & Wind & PlayTennis \\
111 & 10 & 10 \\
\end{array}
\]
Operators for Genetic Algorithms

<table>
<thead>
<tr>
<th>Initial strings</th>
<th>Crossover Mask</th>
<th>Offspring</th>
</tr>
</thead>
<tbody>
<tr>
<td>11101001000</td>
<td>11111000000</td>
<td>11101010101</td>
</tr>
<tr>
<td>00001010101</td>
<td></td>
<td>00001001000</td>
</tr>
</tbody>
</table>

**Single-point crossover:**

<table>
<thead>
<tr>
<th>Initial strings</th>
<th>Crossover Mask</th>
<th>Offspring</th>
</tr>
</thead>
<tbody>
<tr>
<td>11101001000</td>
<td>00111110000</td>
<td>11001011000</td>
</tr>
<tr>
<td>00001010101</td>
<td></td>
<td>00100010010</td>
</tr>
</tbody>
</table>

**Two-point crossover:**

<table>
<thead>
<tr>
<th>Initial strings</th>
<th>Crossover Mask</th>
<th>Offspring</th>
</tr>
</thead>
<tbody>
<tr>
<td>11101001000</td>
<td>10011010011</td>
<td>10001000100</td>
</tr>
<tr>
<td>00001010101</td>
<td></td>
<td>01101011001</td>
</tr>
</tbody>
</table>

**Uniform crossover:**

<table>
<thead>
<tr>
<th>Initial strings</th>
<th>Crossover Mask</th>
<th>Offspring</th>
</tr>
</thead>
<tbody>
<tr>
<td>11101001000</td>
<td></td>
<td>11101010100</td>
</tr>
</tbody>
</table>

**Point mutation:**

<table>
<thead>
<tr>
<th>Initial strings</th>
<th>Crossover Mask</th>
<th>Offspring</th>
</tr>
</thead>
<tbody>
<tr>
<td>11101001000</td>
<td></td>
<td>11101011000</td>
</tr>
</tbody>
</table>
Selecting Most Fit Hypotheses

Fitness proportionate selection:

\[
Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}
\]

... 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
Genetic Programming

Population of programs represented by trees

\[ \sin(x) + \sqrt{x^2 + y} \]
Crossover

\[
\begin{align*}
\sin x^2 + \sin x^2 \wedge \sin y^2 + \sin y^2 \\
\sin x^2 + \sin x^2 \\
\sin x^2 + \sin x^2 \\
\end{align*}
\]
GP for Classifying Images

[Teller and Veloso, 1997]

**Fitness:** based on coverage and accuracy

**Representation:**

- Primitives include Add, Sub, Mult, Div, Not, Max, Min, Read, Write, If-Then-Else, Either, Pixel, Least, Most, Ave, Variance, Difference, Mini, Library

- Mini refers to a local subroutine that is separately co-evolved

- Library refers to a global library subroutine (evolved by selecting the most useful minis)

**Genetic operators:**

- Crossover, mutation

- Create “mating pools” and use rank proportionate reproduction
Biological Evolution

Lamark (19th century)

- Believed individual genetic makeup was altered by lifetime experience
- But current evidence contradicts this view

What is the impact of individual learning on population evolution?
Summary: Evolutionary Programming

- Conduct randomized, parallel, hill-climbing search through $H$
- Approach learning as optimization problem (optimize fitness)
- Nice feature: evaluation of Fitness can be very indirect
  - consider learning rule set for multistep decision making
  - no issue of assigning credit/blame to indiv. steps