Lecture 2: Canonical Genetic Algorithms

What Are Genetic Algorithms?

- Genetic algorithms are optimization algorithm inspired from natural selection and genetics
- A candidate solution is referred to as an individual
- Process
  - Parent individuals generate offspring individuals
  - The resultant offspring are evaluated for their fitness
  - The fittest offspring individuals survive and become parents
  - The process is repeated
History of Genetic Algorithms

In 1960’s
- Rechenberg: “evolution strategies”
  - Optimization method for real-valued parameters
- Fogel, Owens, and Walsh: “evolutionary programming”
  - Real-valued parameters evolve using random mutation

In 1970’s
- John Holland and his colleagues at University of Michigan developed “genetic algorithms (GA)”
- Holland’s 1975 book “Adaptation in Natural and Artificial Systems” is the beginning of the GA
- Holland introduced “schemas,” the framework of most theoretical analysis of GAs.
In 1990’s

- John Koza: “genetic programming” used genetic algorithms to evolve programs for solving certain tasks

- It is generally accepted to call these techniques as evolutionary computation

- Strong interaction among the different evolutionary computation methods makes it hard to make strict distinction among GAs, evolution strategies, evolutionary programming and other evolutionary techniques
Differences Between GAs and Traditional Methods

- GAs operate on encodings of the parameters values, not necessarily the actual parameter values.
- GAs operate on a population of solutions, not a single solution.
- GAs only use the fitness values based on the objective functions.
- GAs use probabilistic computations, not deterministic computations.
- GAs are efficient in handling problems with a discrete or mixed search spaces.
The Canonical GA
Canonical GA

The canonical genetic algorithm refers to the GA proposed by John Holland in 1965.
Gene Representation

- Parameter values are encoded into **binary** strings of **fixed** and **finite** length
  - Gene: each bit of the binary string
  - Chromosome: a binary string
  - Individual: a set of one or multiple chromosomes, a prospective solution to the given problem
  - Population: a group of individuals

- Longer string lengths
  - Improve resolution
  - Requires more computation time
Binary Representation

- Suppose we wish to maximize $f(x)$

- Where $x \in \Omega; \quad \Omega = [x_{\text{min}}, x_{\text{max}}]$  

- Binary representation $x_{\text{binary}} \in [b_l b_{l-1} \cdots b_2 b_1]$  

- We map $[x_{\text{min}}, x_{\text{max}}]$ to $[0, 2^l - 1]$  

- Thus $x = x_{\text{min}} + \frac{x_{\text{max}} - x_{\text{min}}}{2^l - 1} \sum_{i=1}^{l} b_i 2^{i-1}$
Example

- Let \( l = 5, \Omega = [-5, 20] \)

Then

\[
x_{\text{binary}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow x = -5
\]

\[
x_{\text{binary}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix} \Rightarrow x = 20
\]

\[
x_{\text{binary}} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \end{bmatrix} \Rightarrow x = -5 + (2^4 + 2^1 + 2^0) \frac{20 - (-5)}{2^5 - 1} = 10.3226
\]
Fitness Evaluation

- Each individual $x$ is assigned with a fitness value $f(x)$ as the measure of performance.
- It is assumed that the fitness value is positive and the better the individual as a solution, the fitness value is more positive.
- The objective function can be the fitness function itself if it is properly defined.
Example 1

- Consider the problem

\[ \max g(x) = -x^2 + 4x, \quad x \in [1, 5] \]

- A fitness function

\[ f(x) = -g(x) + 100 = -x^2 + 4x + 100 \]
Example 2

- Consider the problem

\[ \min g(x) = x^2, \quad x \in [-10, 10] \]

- A fitness function

\[ f(x) = 1/(g(x) + 0.1) = 1/(x^2 + 0.1) \]
Selection

- Chooses individuals from the current population to constitute a mating pool for reproduction
- Fitness proportional selection methods
  - Each individual $x$ is selected and copied in the mating pool with the probability proportional to fitness $\left( \frac{f(x)}{\sum f(x)} \right)$
Roulette Wheel Selection

Individuals with fitness values

Assign a piece proportional to the fitness value

Mating pool
Crossover

- **Single-point crossover** is assumed
- Two parent individuals are selected from mating pool
- Crossover operation is executed with the probability $p_c$
- Crossover point is randomly chosen and the strings are swapped with respect to the crossover point between the two parents
Single Point Crossover

Parent 1 1 0 1 0 0 1 1 0
Parent 2 1 1 0 0 1 0 1 1

Crossover point

Child 1 1 0 1 0 1 0 1 1
Child 2 1 1 0 0 0 1 1 0
Mutation

- Mutation operator is applied gene-wise, that is, each gene undergoes mutation with the probability $p_m$.
- When the mutation operation occurs to a gene, its gene value is flipped.

![Mutation diagram]

Mutation points

1 1 0 0 1 0 1 1

$\rightarrow$

1 0 0 0 1 1 1 1
Overview of Canonical GA

Initial population → Fitness Evaluation $f(x)$ → Selection

Mutation → Crossover → Mating pool
Summary of Canonical GA

- Binary representation
- Fixed string length
- Fitness proportional selection operator
- Single-point crossover operator
- Gene-wise mutation operator
A Manual Example
Using Canonical GAs
Problem Description

- Consider the following maximization problem

\[
\max f(x) = x^2
\]

where \( x \) is an integer between 0 and 31

\( f(x) = x^2 \) has its maximum value 961 at \( x = 31 \)
Gene Representation

- Before applying GA, a representation method must be defined
- Use unsigned binary integer of length 5

10110 $\rightarrow$ $1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 22$

- A five-bit unsigned binary integer can have values between 0(00000) and 31(11111)
Canonical GA Execution Flow

START

Initialization

Fitness evaluation

Selection

Crossover

Mutation

STOP?

END
Initialization

- Initial populations are randomly generated
- Suppose that the population size is 4
- An example initial population

<table>
<thead>
<tr>
<th>Individual No.</th>
<th>Initial population</th>
<th>$x$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01101</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>11000</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>01000</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>10011</td>
<td>19</td>
</tr>
</tbody>
</table>
Fitness Evaluation

- Evaluate the fitness of initial population
- The objective function $f(x)$ is used as the fitness function
- Each individual is decoded to integer and the fitness function value is calculated
Fitness Evaluation

- Decoding

01000 \[ \xrightarrow{\text{Decode}} \] \( x = 8 \) \[ \xrightarrow{\text{Evaluation}} \] \( f(8) = 8^2 = 64 \)

- Results

<table>
<thead>
<tr>
<th>Individual No.</th>
<th>Initial population</th>
<th>( x ) value</th>
<th>( f(x) )</th>
<th>( f_i / \Sigma f )</th>
<th>Expected number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01101</td>
<td>13</td>
<td>169</td>
<td>0.14</td>
<td>0.56</td>
</tr>
<tr>
<td>2</td>
<td>11000</td>
<td>24</td>
<td>576</td>
<td>0.49</td>
<td>1.96</td>
</tr>
<tr>
<td>3</td>
<td>01000</td>
<td>8</td>
<td>64</td>
<td>0.06</td>
<td>0.24</td>
</tr>
<tr>
<td>4</td>
<td>10011</td>
<td>19</td>
<td>361</td>
<td>0.31</td>
<td>1.24</td>
</tr>
</tbody>
</table>
Selection

- Select individuals to the mating pool
- Selection probability is proportional to the fitness value of the individual
- Roulette wheel selection method is used

<table>
<thead>
<tr>
<th>Individual No.</th>
<th>Initial population</th>
<th>$x$ value</th>
<th>$f(x)$</th>
<th>$f_i / \Sigma f$</th>
<th>Expected number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01101</td>
<td>13</td>
<td>169</td>
<td>0.14</td>
<td>0.56</td>
</tr>
<tr>
<td>2</td>
<td>11000</td>
<td>24</td>
<td>576</td>
<td>0.49</td>
<td>1.96</td>
</tr>
<tr>
<td>3</td>
<td>01000</td>
<td>8</td>
<td>64</td>
<td>0.06</td>
<td>0.24</td>
</tr>
<tr>
<td>4</td>
<td>10011</td>
<td>19</td>
<td>361</td>
<td>0.31</td>
<td>1.24</td>
</tr>
</tbody>
</table>
Selection

- Roulette wheel
  - Outcome of Roulette wheel is 1, 2, 2, and 4
  - Resulting mating pool

<table>
<thead>
<tr>
<th>No.</th>
<th>Mating pool</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 1 1 0 1</td>
</tr>
<tr>
<td>2</td>
<td>1 1 0 0 0</td>
</tr>
<tr>
<td>3</td>
<td>1 1 0 0 0</td>
</tr>
<tr>
<td>4</td>
<td>1 0 0 1 1</td>
</tr>
</tbody>
</table>
Crossover

- Two individuals are randomly chosen from mating pool
- Crossover occurs with the probability of \( p_c = 1 \)
- Crossover point is chosen randomly

<table>
<thead>
<tr>
<th>Mating pool</th>
<th>Crossover point</th>
<th>New population</th>
<th>( x )</th>
<th>( f(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 0 1 1</td>
<td>4</td>
<td>0 1 1 0 0</td>
<td>12</td>
<td>144</td>
</tr>
<tr>
<td>1 1 0 0 0 0</td>
<td></td>
<td>1 1 0 0 1</td>
<td>25</td>
<td>625</td>
</tr>
<tr>
<td>1 1 0 0 0</td>
<td>2</td>
<td>1 1 0 1 1</td>
<td>27</td>
<td>729</td>
</tr>
<tr>
<td>1 0 0 1 1</td>
<td></td>
<td>1 0 0 0 0</td>
<td>16</td>
<td>256</td>
</tr>
</tbody>
</table>
Mutation

- Applied on a bit-by-bit basis
- Each gene mutated with probability of $p_m = 0.001$

<table>
<thead>
<tr>
<th>Before Mutation</th>
<th>After Mutation</th>
<th>$x$</th>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 0 0</td>
<td>0 1 1 0 0</td>
<td>12</td>
<td>144</td>
</tr>
<tr>
<td>1 1 0 0 1</td>
<td>1 1 0 0 1</td>
<td>25</td>
<td>625</td>
</tr>
<tr>
<td>1 1 0 1 1</td>
<td>1 1 0 1 1</td>
<td>27</td>
<td>729</td>
</tr>
<tr>
<td>1 0 0 0 0</td>
<td>1 0 0 1 0</td>
<td>18</td>
<td>324</td>
</tr>
</tbody>
</table>
Fitness Evaluation

- Fitness values of the new population are calculated

<table>
<thead>
<tr>
<th>Old population</th>
<th>$x$</th>
<th>$f(x)$</th>
<th>New population</th>
<th>$x$</th>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>01101</td>
<td>13</td>
<td>169</td>
<td>01100</td>
<td>12</td>
<td>144</td>
</tr>
<tr>
<td>11000</td>
<td>24</td>
<td>576</td>
<td>11001</td>
<td>25</td>
<td>625</td>
</tr>
<tr>
<td>01000</td>
<td>8</td>
<td>64</td>
<td>11011</td>
<td>27</td>
<td>729</td>
</tr>
<tr>
<td>10011</td>
<td>19</td>
<td>361</td>
<td>10010</td>
<td>18</td>
<td>324</td>
</tr>
<tr>
<td>sum</td>
<td>1170</td>
<td></td>
<td>sum</td>
<td>1756</td>
<td></td>
</tr>
<tr>
<td>avg</td>
<td>293</td>
<td></td>
<td>avg</td>
<td>439</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>576</td>
<td></td>
<td>max</td>
<td>729</td>
<td></td>
</tr>
</tbody>
</table>