Genetic Algorithms

ROSLINA RASHID
Prepared by
Ganesan
Kean Beng
Introduction

• Definition
  – A class of stochastic search algorithms based on biological evolution

• GA applies few steps as by Davis, 1991 and Mitchell, 1996.
GA Steps

• Step 1: Set population size and probability
• Step 2: Define fitness function
• Step 3: Generate initial population
• Step 4: Calculate fitness for each chromosome
• Step 5: Mating of chromosomes
• Step 6: Create offspring-crossover & mutation
• Step 7: Offspring in new population
• Step 8: Repeat Step 5 until new=initial population
• Step 9: Replace initial population with new
• Step 10: To Step 4 & repeat until criteria achieved
About GA

• Common practise - To terminate GA after specified number of generations to examine the best chromosome
• Roulette wheel selection – chromosome selection technique
How does Crossover operator work?

1. Choose crossover point
2. Break/exchange/clone between parent
3. New offspring
Mutation

• Change in gene
• 2 consequences: GOOD and BAD
• Purpose: prevent search algorithm trapped on local optimum.
• Mutation=random search, aids in avoiding loss of genetic diversity
- Randomly **flips** a selected gene.

- Occur in any gene with specific probability.

- Probability: small (0.001 - 0.01)
- Genetic algorithms assure: the continuous improvement of the average fitness
- after a number of generations: the population evolves to a near-optimal solution.
- In the example, the final population would consist of only chromosomes 1000 and 0111, both with highest fitness.
E.g: A “peak” function with two variables:

\[ f(x, y) = (1 - x)^2 e^{-x^2-(y+1)^2} - (x - x^3 - y^3) e^{-x^2-y^2} \]

To find the maximum of the function:

**Step 1:** represent parameters \( x \) and \( y \) as a concatenated binary string.

**Step 2:** choose size:

\[
\begin{array}{c}
\text{1000101001111011} \\
\hline
x & y
\end{array}
\]

**Step 3:** calculate the fitness (decode to real numbers and substitute in “peak” function)
-16 bits to 8 bits partition:

\[
\begin{align*}
10001010 \quad \text{and} \quad 00111011
\end{align*}
\]

- Convert from base 2 to (decimal value) base 10:

\[
\begin{align*}
(10001010)_2 &= (138)_{10} \\
(00111011)_2 &= (59)_{10}
\end{align*}
\]

- The -3 to 3 range mapped to handle the 8-bits:

\[
\frac{6}{256 - 1} = 0.0235294
\]

- Actual value : multiply decimal value with mapped range.

\[
\begin{align*}
x &= (138)_{10} \times 0.0235294 - 3 = 0.2470588 \\
y &= (59)_{10} \times 0.0235294 - 3 = -1.6117647
\end{align*}
\]
- Using value of x and y, calculate the fitness.

- To find the maximum: crossover probability = 0.7 and mutation probability = 0.001

- Specify num. of generations: e.g 100.
- Used to represent the fitness function for the real world problems with stochastic (random) solution like GA.

- shows: **average** performance of entire population and performance of **best** individual chromosome.
- erratic behaviour of average curve: because of mutation.

-solution: increase the chromosome population.

Figure 7.8 Performance graphs for 2.0 generations of 60 chromosomes

$R_e = 0.7, \ p_m = 0.001$