Lesson 10: Genetic Algorithm

Reviewer

Key Terms

- ***Mutation***
- ***Gene***
- ***Chromosome***
- ***Crossover***
- ***Population***
- ***Fitness***

What is Genetic Algorithm?

Genetic algorithms (GAs) and ***genetic programming (GP)*** are branches of ***evolutionary computing***, a subset of artificial intelligence where solutions evolve over time to fit a given set of parameters or solve specific problems.

Process Flow

1. ***Start***
2. ***Initialization***
3. ***Evaluation***
4. ***Selection***
5. ***Crossover***
6. ***Mutation***
7. ***Generation***
8. Check: ***End?***
    - If No: Loop back
    - If Yes: ***Stop***

Understanding Genetic Algorithms

A ***genetic algorithm*** or ***GA*** is a search technique used in computing to find true or approximate solutions to ***optimization and search problems***.

The techniques are inspired by ***natural evolution*** such as ***inheritance***, ***mutation***, ***selection***, and ***crossover***.

GAs are implemented by having an array of bits or characters to represent the ***chromosomes***.

Detailed Process Flow

1. ***Start***
2. ***Initialization***: Create ***Initial Population***
3. Loop:
    (a) ***Selection***
    (b) ***Crossover***
    (c) ***Mutation***
    (d) Create ***New Population***
4. Check Termination (Quiet?):
    - If No: Return to ***Old Population*** and repeat loop.
    - If Yes: ***End***

Steps for Genetic Algorithm

1. ***Choose initial population***
2. ***Evaluate the fitness*** of each individual in the population
3. Repeat until termination:
4. ***Select best ranking individuals*** to reproduce
5. Breed new generation through ***crossover*** and/or ***mutation*** and give birth to offspring
6. ***Evaluate the individual fitnesses*** of the offspring
7. Replace worst ranked part of the population with offspring

Foundation of Genetic Algorithms

Genetic algorithms are based on an analogy with the genetic structure and behavior of ***chromosomes*** of the population. Following is the foundation of GAs based on this analogy:

1. Individuals in the population ***compete for resources and mate***.
2. Those individuals who are successful (***fittest***) then mate to create more offspring than others.
3. Genes from the ***"fittest" parent*** propagate throughout the generation; that is, sometimes parents create offspring which is ***better than either parent***.
4. Thus each successive generation is ***more suited for their environment***.

Operators of Genetic Algorithms

1. ***Selection Operator***: The idea is to give preference to the individuals with ***good fitness scores*** and allow them to pass their genes to successive generations.
2. ***Crossover Operator***: This represents ***mating between individuals***. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a ***completely new individual (offspring)***.

   Example Diagram Description:

   - Parent 1: A B | C D | E F G H
   - Parent 2: F G | H A | D B E A
   - Offspring: F G H B C D E A (Genes exchanged at crossover sites)
3. ***Mutation Operator***: The key idea is to ***insert random genes*** in offspring to maintain the ***diversity in the population*** to avoid premature convergence.

   Example Diagram Description:

   - Before Mutation: F G H B C D E A
   - After Mutation: F G ***M*** B C D E ***N*** (Random genes 'M' and 'N' inserted)

Algorithm Summary

The whole algorithm can be summarized as:

1. ***Randomly initialize populations***
2. ***Determine fitness*** of population
3. Until convergence repeat:
   (a) ***Select parents*** from population
   (b) ***Crossover*** and generate new population
   (c) Perform ***mutation*** on new population
   (d) ***Calculate fitness*** for new population