Assignment 2

CS 776: Evolutionary Computing

Fall 2023

Abhishek Kundu

Email: abhishekk@nevada.unr.edu

1. Simple Genetic Algorithm

Input:

- Population size (N)
- Number of generations (G)
- Crossover probability (Pc)
- Mutation probability (Pm)
- Fitness evaluation function

Output:

- Best solution found

Procedure SimpleGeneticAlgorithm(N, G, Pc, Pm, FitnessEvaluationFunction):

1. Initialize a random population of N individuals.
2. For generation = 1 to G:
    a. Evaluate the fitness of each individual in the population.
    b. Select individuals to be parents for the next generation.
        - Individuals with higher fitness are more likely to be selected.
        - Techniques like roulette wheel selection or tournament selection or random choice can be used.
    c. Create offspring by applying crossover (recombination).
        - With probability Pc, select pairs of parents and combine their genetic material to create children.
        - This mimics the process of genetic recombination.
    d. Apply mutation to introduce genetic diversity.
        - For each gene in each child, with probability Pm, flip its value.
    e. Evaluate the fitness of the offspring DeJong fitness function.
    f. Select individuals to form the next generation.
        - This can involve strategies like generational replacement or steady-state replacement.
    g. Keep track of the best solution found.
3. Return the best solution.

2. Algorithm: CHC Adaptive Search Algorithm

Input:

- Chromosome length (N)
- Population size (P)
- Divergence rate (D)
- DeJong fitness function (dejong_func)
- Crossover probability (Pc)
- Mutation probability (Pm)

Output:

- Best individual found
- Fitness of the best individual

Procedure CHCAdaptiveSearch(N, P, D, dejong_func, Pc, Pm):

1. Initialize a population of P individuals with random binary strings of length N.
2. Set the best individual to the highest fitness individual in the initial population.
3. Set d to N divided by 4.
4. Initialize t to 0.
5. While t is less than D:
    a. Apply crossover to pairs of parents in the population with probability Pc.
    b. Apply mutation to the resulting children with probability Pm.
    c. Evaluate the fitness of the children using the DeJong fitness function.
    d. Combine the parents and children to form a combined population.
    e. Sort the combined population by fitness in descending order and select the top P individuals.
    f. If the new population is the same as the previous population:
        - Decrement d.
        - If d is less than 0, reinitialize the population and reset d.
    g. Set the best individual from the previous generation as the first individual in the new population.
    h. Update the current population with the new population.
    i. Increment t.
6. Find the best individual from the final population based on fitness.
7. Calculate the fitness of the best individual.
8. Return the best individual and its fitness.