Faculty of Computing and Information Technology

University of the Punjab,

Lahore

Artificial Intelligence Lab 7

Instructor: Qamar U Zaman

1. Genetic Algorithm (GA)

Introduction to Genetic Algorithm

The Genetic Algorithm (GA) is an optimization and search heuristic inspired by the process of natural selection. GA is commonly used for solving optimization and search problems where solutions evolve over time, getting closer to the best solution with each generation.

Problem: Knapsack Problem

The objective is to solve the classic Knapsack problem using a Genetic Algorithm. The problem involves selecting items to maximize the total value without exceeding a given weight limit.

- **Knapsack Weight Limit**: 50 units
- **Available Items**: Each item has a specific weight and value.

Item	Weight	Value
1	10	60
2	20	100
3	30	120
4	15	75
5	25	90

Code Template

```
class Chromosome:
    def __init__(self, genes):
        # Initialize with genes (binary string) and fitness
        pass

    def calculate_fitness(self):
        # Calculate total value if weight is within limit, else
set fitness to 0
        pass

class GeneticAlgorithm:
    def __init__(self, population_size, mutation_rate,
crossover_rate, generations):
        # Initialize GA parameters
        pass

    def initialize_population(self):
        # Create an initial random population of chromosomes
        pass
```

```
def selection(self):
        # Select parents based on fitness (e.g., roulette wheel
selection)
        pass

    def crossover(self, parent1, parent2):
        # Perform crossover to produce offspring
        pass

    def mutate(self, chromosome):
        # Apply mutation by flipping random bits
        pass

    def evolve(self):
        # Run GA for the defined number of generations
        pass

    def get_best_solution(self):
        # Identify and return the best chromosome in the
population
        pass
```

Lab Tasks

1. **Implement the Genetic Algorithm for the Knapsack Problem**
    - Define a population with 5 items represented by binary genes (1 for inclusion, 0 for exclusion).
    - Use the knapsack weight limit of 50 and calculate fitness as the total value of included items within the weight limit.
2. **Perform Selection, Crossover, and Mutation**
    - Implement selection using Roulette Wheel or Tournament Selection.
    - Set the crossover rate to 0.7 and mutation rate to 0.01.
3. **Run the Algorithm and Evaluate**
    - Run the GA for 20 generations and track the maximum value achieved without exceeding the weight limit.