Home work 3: Genetic Algorithm for the Knapsack Problem

Problem Statement

The Knapsack Problem involves a set of items, each with a specific weight and value, and a knapsack with a limited weight capacity. The goal is to select a subset of items to maximize the total value without exceeding the knapsack's weight capacity.

Your Task

Write a Python program to solve the Knapsack Problem using a Genetic Algorithm. Your program should include the following components:

Representation of an Individual Solution

Represent an individual solution (a potential set of items to include in the knapsack) as a binary string or a list of binary values, where 1 indicates the inclusion of an item and 0 indicates exclusion.

Initialization

Generate an initial population of these individual solutions randomly.

Fitness Function

Define a fitness function that calculates the total value of items included in the knapsack. Penalize the solution if the total weight exceeds the capacity.

Selection Mechanism

Implement a selection mechanism, like tournament selection or roulette wheel selection, to choose individuals for reproduction based on their fitness.

Crossover and Mutation

- Apply crossover (e.g., single-point or uniform crossover) to generate offspring from selected parents.
- Implement mutation (e.g., flipping a bit in the binary representation) with a small probability.

Creating a New Generation

Create a new generation of solutions using the offspring and perhaps some individuals from the current generation.

Termination Condition

Continue iterating through generations until a certain condition is met (like a maximum number of generations or a satisfactory fitness level).

Output

The program should output the optimal set of items to include in the knapsack and the total value achieved.

Constraints

- Assume a list of items, where each item is represented as a tuple `(value, weight)`.
- The weight capacity of the knapsack is provided.

Sample Data

```
items = [(60, 10), (100, 20), (120, 30)]  # Each tuple is (value, weight)
knapsack_capacity = 50
```

This question tests your understanding of genetic algorithms, including representation of solutions, fitness evaluation, genetic operations (crossover and mutation), and the iterative process of evolving a population towards an optimal solution.