Task 1: Pattern Finding via Genetic Algorithm

1. Problem Overview

The task involves finding specific patterns in a 3x3 grid using a genetic algorithm (GA). The grid is:

A B C
D E F
G H I

Given starting (`A`), middle (`I`), and ending (`C`) characters, the goal is to generate valid patterns that connect these points in sequence.

2. Genetic Algorithm Approach

2.1 Initialization

- **Grid Mapping**: The grid is mapped to positions for easy access.
- **Population**: Random patterns are generated, ensuring they start with `A`, include `I`, and end with `C`.

2.2 Fitness Evaluation

- Patterns are scored based on:
  - Starting, containing, and ending with the correct characters.
  - Validity within the grid (e.g., unique points, valid transitions).

2.3 Selection, Crossover, Mutation

- **Selection**: Randomly selects parents from the population.
- **Crossover**: Combines parents to produce offspring, validated to ensure correctness.
- **Mutation**: Random swaps in the pattern to maintain diversity.

3. Results

The algorithm efficiently finds valid patterns within a few hundred generations. Performance depends on parameters like mutation rate and population size.

4. Conclusion

The GA successfully generates valid grid patterns based on the input criteria. The implementation is flexible and can be adapted to more complex grids or constraints.

Task 2: Genetic Algorithm for Multi-Vehicle Routing Problem

1. Initialization

- **Population**: The algorithm begins by generating an initial population of random route sequences. Each route includes customer points and vehicle assignments, starting and ending at a depot.
- **Distance Matrix**: A matrix is precomputed to store distances between all points (customers, depot, vehicles) to accelerate fitness calculations.

1.2 Fitness Function

- **Fitness Calculation**: The fitness of each route is determined by the total transportation cost, which is influenced by the distance traveled and the cost per distance unit for each vehicle type. The algorithm penalizes routes that exceed vehicle capacity or budget constraints to ensure valid solutions.

1.3 Genetic Operations

- **Selection**: The best-performing individuals (routes) are selected based on their fitness scores, with an elite group guaranteed to pass to the next generation.
- **Crossover**: Two parent routes are combined by swapping segments to create offspring, ensuring diversity in the population while maintaining route validity.
- **Mutation**: Random swaps of customer points within a route introduce variability, helping to explore new potential solutions.

1.4 Convergence and Early Stopping

- **Early Stopping**: The GA monitors the fitness of top solutions across generations. If improvements become negligible, the algorithm stops early to save computational resources.

2. Results Indication

The GA successfully optimized vehicle routes by significantly reducing both the total cost and distance traveled. Key results include:

- **Cost Efficiency**: The algorithm minimized total costs by selecting optimal routes and vehicle assignments, ensuring no budget constraints were violated.
- **Distance Optimization**: It effectively minimized the distance each vehicle traveled, leading to lower fuel consumption and operational costs.
- **Convergence**: The solution typically converged within a few hundred generations, indicating that the GA efficiently explored the solution space.

3. Conclusion

The implementation of the GA proved effective for the Multi-Vehicle Routing Problem. It consistently generated optimized routes that balanced cost, distance, and capacity constraints.