**Genetic Algorithm for Vehicle Routing Problem**

***Chromosome Representation:***

Chromosomes are represented as sets of routes for multiple vehicles, where each route serves a subset of customers and starts and ends at the depot.

Example:

Chromosome 1:

{('A', [(4.4184, 114.0932), (4.3555, 113.9777, 5), (4.3163, 114.0764, 3), (4.3184, 113.9932, 6), (4.4024, 113.9896, 5), (4.4142, 114.0127, 8), (4.4804, 114.0734, 3), (4.4184, 114.0932)]),

('A', [(4.4184, 114.0932), (4.3976, 114.0049, 8), (4.3818, 114.2034, 6), (4.4935, 114.1828, 5), (4.4932, 114.1322, 8), (4.4184, 114.0932)])}

***Fitness Function:***

The fitness function evaluates the total cost of all routes, considering both distance traveled and vehicle type costs.

***Population Initialization:***

Initialization randomly assigns customers to random type of vehicles, ensuring that each route does not exceed vehicle capacity.

***Parent Selection:***

Parents are selected from the current population based on their fitness. In this implementation, the top half of the population based on fitness scores is chosen as parents for the next generation.

***Crossover & Mutation***

One-point crossover is employed, combining genetic information from two parent solutions to create offspring. Swap mutation is used to introduce random changes to offspring solutions, maintaining diversity within the population.

***Survival Selection:***

Offspring population is combined with the parent population. The top individuals based on their fitness scores are selected to survive and form the next generation.

***Repair Mechanism:***

A repair mechanism ensures that offspring solutions adhere to vehicle capacity by removing duplicates and adding unassigned customers to routes.

***Optimization:***

Various population sizes and numbers of generations were tested to find the optimal combination. The best parameters were determined by running the Genetic Algorithm with different configurations and selecting the one yielding the lowest total cost.