COMP5511

SHI Yusen, 25041465G

# I. INTRODUCTION

The Vehicle Routing Problem (VRP) is a combinatorial optimization challenge that determines the most efficient routes for a fleet of vehicles to serve multiple customers while minimizing total travel distance, time, and operational costs. The Vehicle Routing Problem (VRP) is an optimization challenge that requires businesses to determine the most efficient routes for multiple vehicles to deliver goods or services to customers. It involves considering constraints, such as vehicle capacity, time windows, and service times, to create routes.

The Genetic Algorithm (GA) is a metaheuristic inspired by natural selection. It evolves a population of candidate solutions over generations using operations like, **Selection:** Choose better individuals based on fitness. **Crossover:** Combine two parent solutions to create offspring. **Mutation:** Randomly alter parts of a solution to maintain diversity.

GA is widely used for solving VRP because it can efficiently explore large, complex search spaces and find near-optimal solutions when exact algorithms are too slow.

# II. Methodology

Note: Include five subsections regarding the five tasks. In each subsection, you should give a detailed description of the designed algorithm, including the overall framework, variation operators (e.g., crossover and mutation), selection operator, and other components.

## A. CLASSICAL VRP

I Use the genetic algorithm to find the optimal route for a single vehicle (capacity 200) serving all 100 customers from depot NO = 0. The route starts and ends at the depot, respecting the capacity constraint. Customer locations and demands are in ‘VRP.csv’. VRP

The basic VRP is solved using a Genetic Algorithm (GA) implemented with the DEAP framework. Each individual represents a permutation of customers, indicating the visiting order for the vehicle. The algorithm aims to minimize the total travel distance while satisfying vehicle capacity constraints. A large penalty is applied whenever the vehicle exceeds its capacity.

First, the data is processed to separate the depot and customer information from the original CSV file, extract the node coordinates, and extract the numbers in the customer data into a list idx\_to\_cust for subsequent identification of customer node sequences. To speed up the reading of coordinates, the adjacency matrix is ​​established, which can calculate the distance faster in the GA algorithm. The specific implementation of the GA algorithm is registered using the deap package.

**Other Components：**

The fitness function eval\_vrp\_classical() calculates the total route distance using the depot and customer coordinates. If the accumulated demand exceeds the vehicle’s capacity, a large penalty (1e7) is added to the total cost. The algorithm continues until a fixed number of generations or convergence is reached.

In order to reduce the number of single training, after each training, the top ten optimal solutions are saved and added to the initial population in the next training.

## B. STOCHASTIC DEMAND PROBLEM

**Overall Framework：**

The algorithm adopts a Genetic Algorithm (GA) framework implemented with the DEAP library to solve a multi-depot Vehicle Routing Problem (VRP) under stochastic customer demands.  
Each individual represents a permutation of customer nodes, indicating the visiting order. The algorithm evaluates each route using Monte Carlo sampling to account for uncertainty in customer demands.  
A penalty-based fitness function ensures that infeasible routes (exceeding vehicle capacity) are heavily penalized. The process iteratively evolves the population through selection, crossover, and mutation until convergence or reaching the predefined generation limit.

**Crossover Operator：**

A **partially matched crossover (PMX)** (tools.cxPartialyMatched)is used to ensure that each customer appears exactly once in the offspring. This method can combine high-quality sub-paths from different parent generations, maintain population diversity, and accelerate convergence.

**Mutation Operator：**

Choose to use (tools.mutShuffleIndexes) in the DEAP package as the selection operator and set indpb=0.05 to introduce random permutations of customer orders. This mutation operator is lightweight but effective for diversifying the population in a multi-objective setting.

**Selection Operator：**

A **tournament selection** in DEAP package (tools.selTournament) is used to select individuals for reproduction, Randomly select a subset of individuals and choose the best one.

**Other Components：**

1. Fitness Evaluation (Monte Carlo Sampling):

Each individual is evaluated multiple times using sampled customer demands from a truncated normal distribution.  
The expected distance and feasibility rate are calculated as:

In the Monte Carlo simulation, each individual performs multiple random samplings and calculates the average distance and feasibility rate:

1. Elitism:

The best individuals are preserved across generations by using (hof = tools.HallOfFame(10))

1. Visualization:

Draw an image based on the optimal solution obtained after multiple iterative calculations to show the results more intuitively.

**Compare with task 1:**

In the deterministic VRP problem, the GA algorithm optimizes based on average customer demand. Vehicles are always within capacity constraints or nearly fully loaded, resulting in shorter routes. The total distance is optimal or close to the theoretical optimal. However, in a random demand VRP, fluctuations in customer demand can lead to vehicles being overloaded or returning to the warehouse prematurely, even requiring multiple replenishments. To ensure feasibility, the GA may adopt a conservative strategy (reducing vehicle loads, prioritizing safety, and avoiding overload penalties). As a result, the average total distance is often longer than in the deterministic case, as the algorithm sacrifices efficiency to mitigate risk.

Comparing the feasibility rates between the two approaches, in a properly designed deterministic VRP, all individuals essentially meet capacity constraints, resulting in a feasibility rate close to 100%.

In a random demand VRP, due to demand fluctuations, each Monte Carlo sampling may result in an individual being overloaded at some point. The GA needs to optimize the average performance over multiple evaluations, rather than the single performance. The feasibility rate is typically less than 100%, depending on vehicle capacity and the magnitude of demand fluctuations.

Comparing route stability: In a deterministic VRP, the route is fixed, and each run yields the same output based on the same input. Stochastic Demand Vehicle-to-Resource (VRP): The same route may perform differently under different demand patterns.

For example, there may be premature returns (due to exceeding capacity) or "empty" segments (due to reduced loads for safety). Consequently, a nondeterministic VRP route is more conservative and dispersed, avoiding single overloads and reflecting a risk-averse strategy.

This contrasts with a deterministic VRP, which aims for the shortest route but may face the risk of actual overloads in reality.

## C. LARGE-SCALE OPTIMIZATION PROBLEM

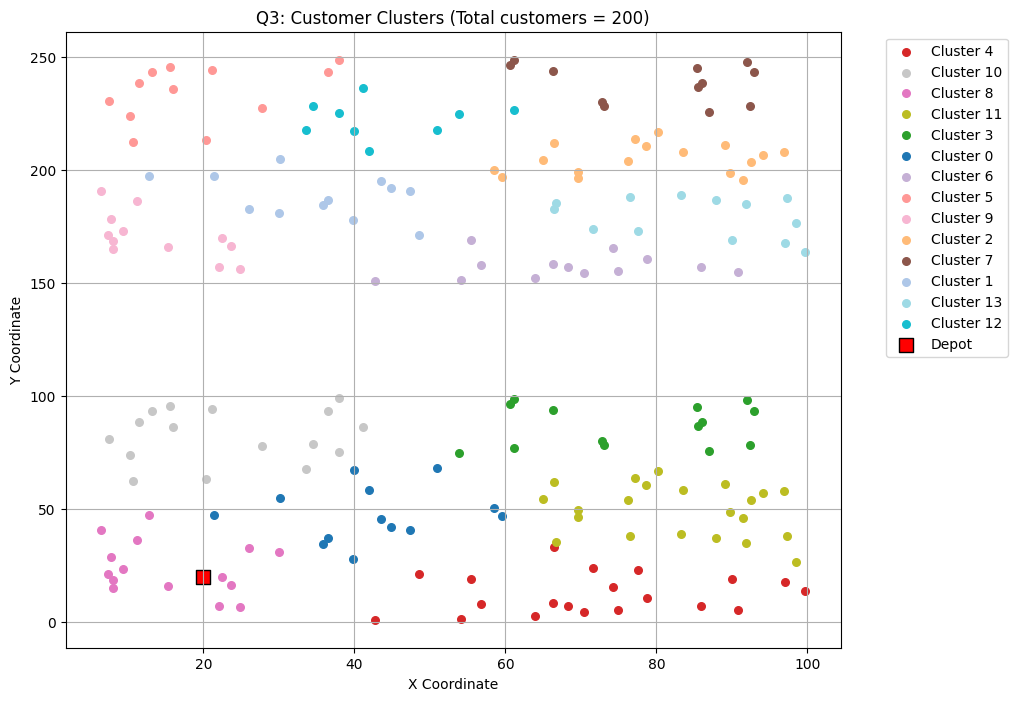
**Overall Framework：**

To improve computational efficiency and reduce the complexity of large-scale vehicle routing problems, customer locations are clustered before route optimization. The K-Means algorithm is applied, with the number of clusters heuristically determined as:

where is the total number of customers. The algorithm first attempts to use sklearn.KMeans; if the library is unavailable, a simplified Lloyd-based K-Means implementation is used as a fallback.

In each iteration, customer points are assigned to the nearest cluster center based on Euclidean distance, and centers are updated as the mean of all points within each cluster. The process continues until the cluster assignments converge or a maximum number of iterations is reached.

After clustering, each customer is labeled according to its cluster, enabling independent route planning within clusters. This decomposition reduces problem scale while preserving spatial coherence between routes.



**Crossover Operator：**

A **partially matched crossover (PMX)** (tools.cxPartialyMatched)is used to ensure that each customer appears exactly once in the offspring. This method can combine high-quality sub-paths from different parent generations, maintain population diversity, and accelerate convergence.

**Mutation Operator：**

Choose to use (tools.mutShuffleIndexes) in the DEAP package as the selection operator and set indpb=0.05 to introduce random permutations of customer orders. This mutation operator is lightweight but effective for diversifying the population in a multi-objective setting.

**Selection Operator：**

A **tournament selection** in DEAP package (tools.selTournament) is used to select individuals for reproduction, Randomly select a subset of individuals and choose the best one.

**Other Components：**

1. Customer Clustering (K-Means):

To reduce the computational complexity of large-scale vehicle routing problems and improve optimization efficiency, customers are first clustered using the K-Means algorithm. The heuristic for determining the number of clusters is described above.

After clustering, customers are assigned cluster labels, which lay the foundation for the subsequent optimization process based on a genetic algorithm (GA). This decomposition breaks down the large vehicle routing problem (VRP) into multiple smaller, more manageable subproblems, improving scalability and convergence stability.

1. Elitism:

The best individuals are preserved across generations by using (hof = tools.HallOfFame(10))

1. Visualization:

After convergence, the optimized routes are visualized to highlight the effectiveness of clustering and the resulting route organization. Each cluster is shown in a distinct color, with the depot marked as a red square. Solid lines represent intra-cluster routes, while dashed lines indicate transitions to and from the depot. This visualization intuitively demonstrates the spatial coherence and the role of clustering in improving route structure.

1. Result Preservation:

After optimization, the best individuals, cluster maps, and route segments are stored for subsequent analysis. This ensures that the optimized solution and cluster structure can be reloaded or visualized later and can provide a better initial population the next time the GA algorithm is run.

**Compare with classical VRP：**

In vehicle routing planning (VRP), clustering methods (such as K-Means) have advantages over the traditional method of treating all customers as a single set.

By grouping geographically close customers, the overall large-scale VRP can be broken down into multiple smaller subproblems. This not only reduces computational complexity but also allows for localized path optimization before integrating the optimization into a global solution. Furthermore, genetic algorithms are more efficient when searching within clusters because intra-cluster paths are short and well-connected, thus reducing the search space, accelerating convergence, and ultimately achieving a better solution.

For real-world problems with larger data volumes, applying the optimization algorithm independently to each cluster allows the method to handle very large VRPs, significantly improving scalability.

## D. MULTI-OBJECTIVE OPTIMIZATION PROBLEM

**Overall Framework：**

The multi-objective VRP extends the classical single-objective model by optimizing two conflicting goals simultaneously: minimizing the total travel distance and maximizing the feasibility rate under stochastic customer demands. The algorithm uses the NSGA-II framework implemented via DEAP, which maintains a Pareto-optimal set of solutions representing different trade-offs between the two objectives. Two solution methods are used: (1) a weighted-sum GA that scalarizes the objectives as to produce single best solutions under different trade-offs; and (2) an NSGA-II implementation (via DEAP) that evolves a Pareto front of non-dominated solutions, preserving diversity with crowding distance and revealing the full trade-off surface between distance and efficiency.

**Crossover Operator：**

Same as Task 1.

**Mutation Operator：**

Same as Task 1.

**Selection Operator：**

The selection operator is tools.selNSGA2, which sorts individuals based on non-dominated fronts and crowding distance. It ensures that both convergence and solution diversity along the Pareto front are maintained.

**Other Components：**

Use this function () to perform fitness evaluation, and try using different values ​​several times.

## E. PICKUP AND DELIVERY PROBLEM

**Overall Framework：**

Based on the classic VRP problem, set 30% of custs are randomly selected and their demand is set to negative, which is considered as replenishment.

**Crossover Operator：**

Same as in Task 1.

**Mutation Operator：**

Same as in Task 1.

**Selection Operator：**

Same as in Task 1.

**Other Components：**

The fitness function (eval\_vrp\_pickup\_delivery) evaluates a mixed pickup-and-delivery route: starting from the main depot, the vehicle visits customers sequentially and adjusts load based on signed demands; if any demand or load change exceeds capacity, it detours to the nearest depot for reloading or unloading with penalties.

# III. Experimental results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Q1(gen=6000) | Q2(gen=800) | Q3(gen=5000) | Q4（w=0.5） | Q5 |
| cxpb=0.7, mutpb=0.25, | Single：2081.93  All depot:1844.10 | 2275.19 | 4699.67 | * f1=2552.58, f2=-5956.75 * f1=2197.52, f2=-5868.70 | 1625.07 |
| cxpb=0.8, mutpb=0.25, | Single：2099.77  All depot:1746.29 | 2536.45 | 4675.03 | * f1=2452.23, f2=-4775.33 * f1=2384.85, f2=-6456.44 | 1799.88 |
| cxpb=0.7, mutpb=0.3, | Singlet：2036.95  All depot:1795.66 | 2401.52 | 4705.40 | * f1=2570.61, f2=-6347.44 * f1=2272.37, f2=-5721.00 | 1691.77 |
| cxpb=0.75, mutpb=0.25, | Single：1959.48  All depot:1825.66 | 2252.86 | 4898.70 | * f1=2444.99, f2=-3611.70 * f1=2191.16, f2=-4958.04 | 1780.91 |

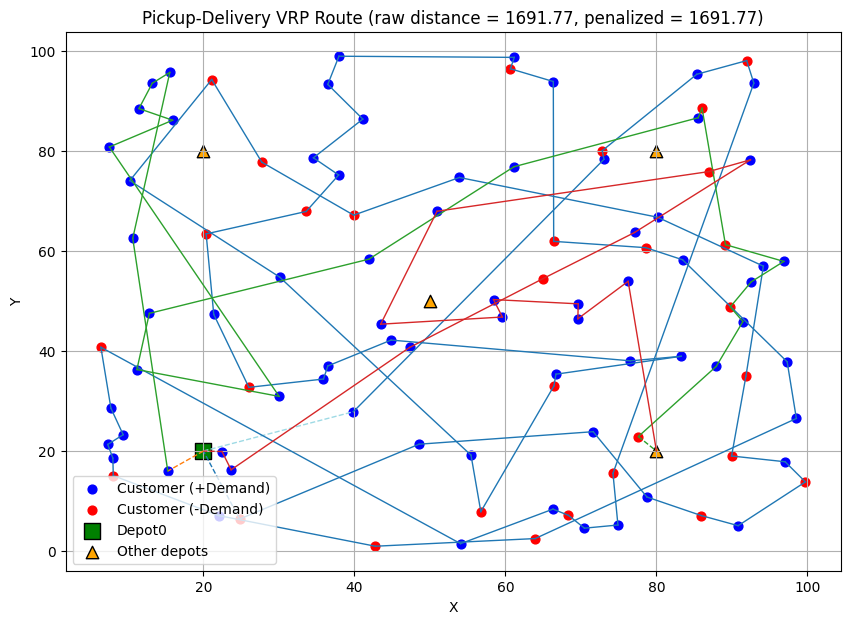
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1(Single depot) | Q1(all) | Q2 | Q3 |
| cxpb=0.7, mutpb=0.25, |  |  |  |  |
| cxpb=0.8, mutpb=0.25, |  |  |  |  |
| cxpb=0.7, mutpb=0.3, |  |  |  |  |
| cxpb=0.75 mutpb=0.25, |  |  |  |  |

It can be seen that when cxpb=0.8, the first two thousand gen of the curve have obvious ups and downs

Q4

|  |  |  |  |
| --- | --- | --- | --- |
| W=0.25 | W=0.75 | W=0.5 | W=1 |

Q5 The Best result:



# IV. Conclusion

In traditional vehicle routing problems, using a GA algorithm and penalizing violations of constraints can yield an optimal solution after multiple iterations.

Q2 and Q3, based on Q1, add dynamic demand based on a normal distribution and more custs, respectively. This requires adjustments to the evaluation function and the use of Monte Carlo simulations to find a relatively stable route that is less likely to exceed weight limits. For Q3, which uses clustering, a GA algorithm is first used within each cluster to find the optimal solution, and then all clusters are combined to find the overall optimal solution. Using clustering makes it easier to solve real-world problems.

Q4 analyzes a dual-objective vehicle routing problem—minimizing total distance traveled and maximizing route efficiency. I used a standard weighted algorithm and the NSGA-II method to balance these two objectives by maintaining a Pareto-optimal solution set.

Q5 is a vehicle routing problem with pickup and delivery tasks (Pickup-and-Delivery VRP), simulating real-world logistics scenarios. GAs can also effectively handle complex constraints involving mixed pickup and delivery demands.

# V. REFERENCES

1. http://vrp-rep.org/
2. https://neo.lcc.uma.es/vrp/vrp-instances/
3. https://github.com/DEAP/deap