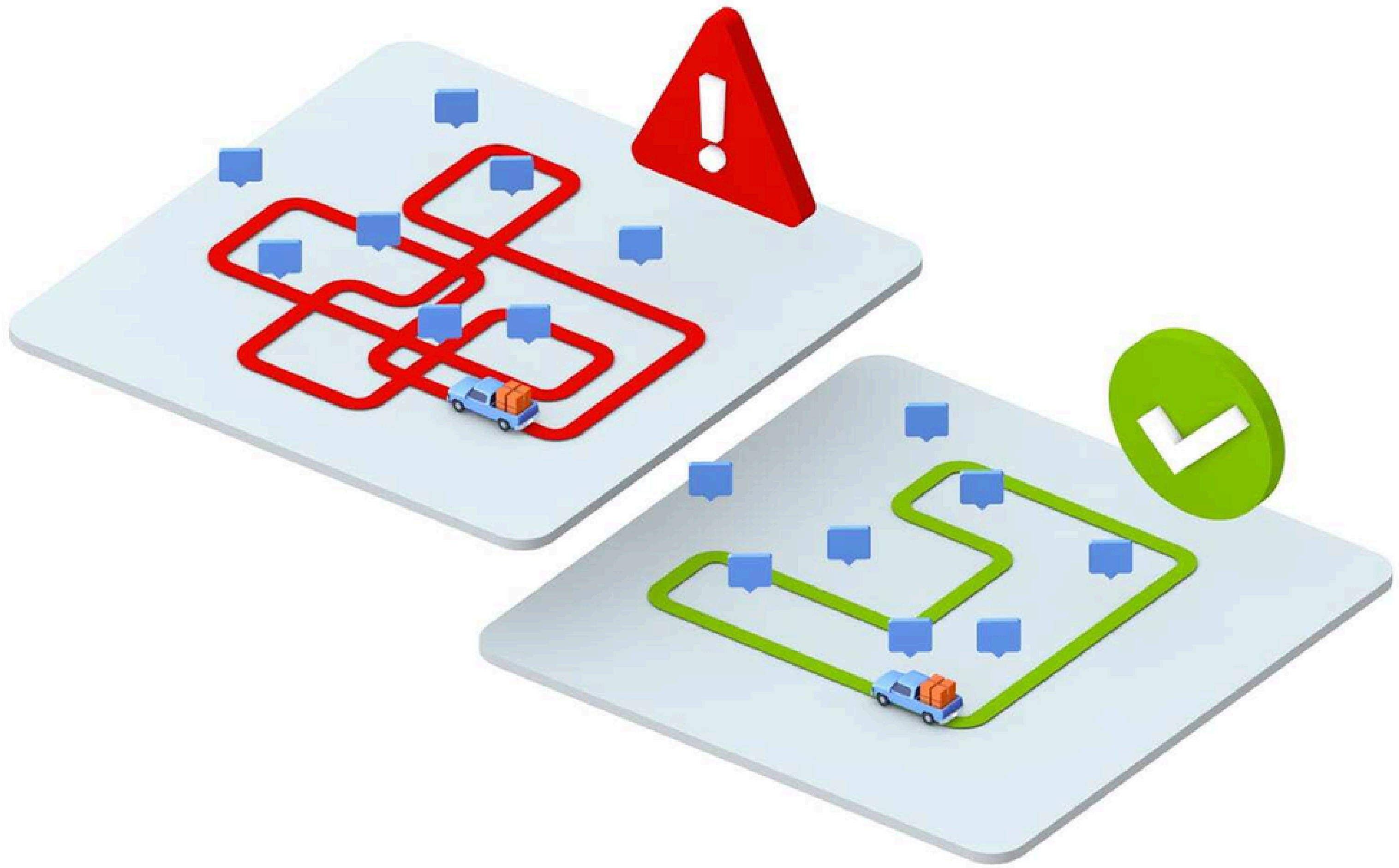
IIT MADRAS
DUAL DEGREE PROJECT
END REVIEW

ENGINEERING DESIGN
IDDD DATA SCIENCE



Vehicle Routing

Employee Pick Up Problem

Presentation by :

REVATI SAWANT
ED19B025

Abstract

Employee pick up problem addresses the transportation of employees from their respective places to factory or an office. This is a challenging problem for every organization particularly when number of people employed is large. Today, companies prefer to outsource these operations to service providers.

The number of employees and their boarding points would change frequently which necessitates rerouting of the buses. From the service provider perspective, the objective is to provide minimum cost (distance) service. However the rerouting could result in increased travel time for some of the employees.

We consider the objective of minimizing the total travel distance by the buses and understand the role of stochasticity to formulate a relevant problem.

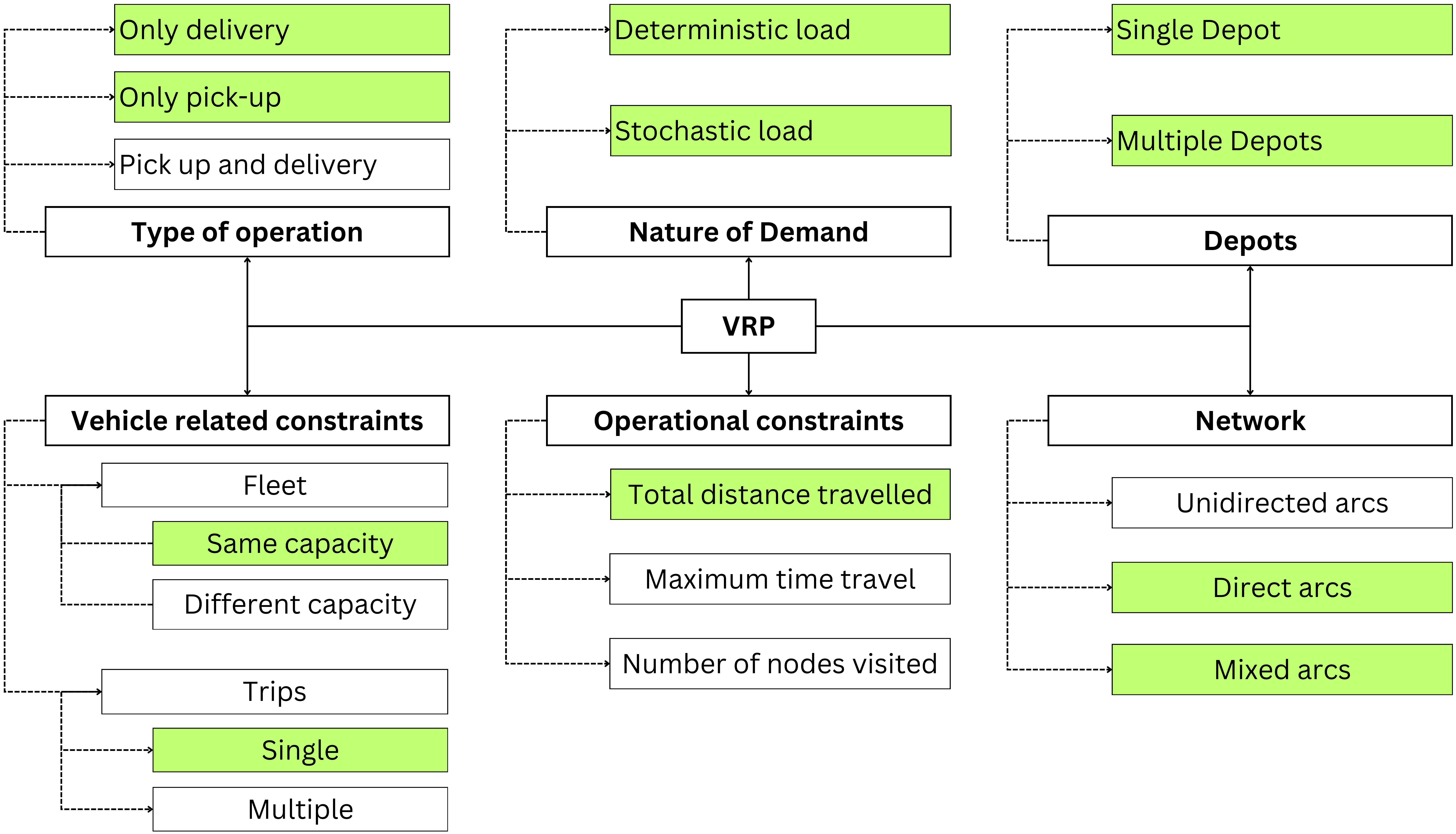
Several models and their performances are documented in the following report with an intent to implement this on real data from a leading automotive manufacturer.

Introduction

Transportation spending is generally observed to be within **top five facility service expenditure** for companies. Minimizing the transportation cost is one of the key objectives that the company looks at while the **employee looks at satisfaction** that comes from travelling minimum distance. There needs to be a good trade off between the transportation cost and employee satisfaction.

Increase in the number of employees and attrition will change the number of boarding points as well as the number of people boarding the buses frequently. This becomes a challenge to the service provider and **rerouting has to be done often**. This would cause some employees to spend more time in the vehicle and sometimes can lead to a route with increased travel times.

Optimization of the routes considering all the above factors would be an important research problem for organizations


Only delivery
Only pick-up
Pick up and delivery
Type of operation
Deterministic load
Stochastic load
Nature of Demand
Single Depot
Multiple Depots
Depots
VRP
Vehicle related constraints
Fleet
Same capacity
Different capacity
Trips
Single
Multiple
Operational constraints
Total distance travelled
Maximum time travel
Number of nodes visited
Network
Unidirected arcs
Direct arcs
Mixed arcs

Methodology

INBUILT CVRP FUNCTION 01

1. Implemented using Pulp model: It uses Branch and Bound programming solver written in C++
2. Low computational power works for small datasets

TABU SEARCH: CVRP 02

1. Initial solution using nearest neighbor heuristic. Algorithm implemented using 4 swap methods

GENETIC: CVRP 03

1. Initial genetic population. Algorithm implemented using two point crossover and mutation

Objective: Minimise travel distance given node locations & demands, vehicle capacity & number

MODEL COMPARISONS 04

1. Comparing tabu serach and genetic models over a bunch of CVRP datasets to evalute the performance
2. Inference table has been provided for the same

MULTI DEPOT 05

1. Implemented using PyGad library where number of depots = number of vehicles
2. Each vehicle picks up the optimal depot depending on the route it has to follow

STOCHASTIC VRP 06

1. Implemented algorithms to handle stochasticity of demands using the idea of expected values
2. Further extended to understand it’s application for employee pick up problem

Pulp model

Data + Distance fn → Distance Matrix → 6 constraints using Pulp → Output: # Vehicles + Distance

DATA

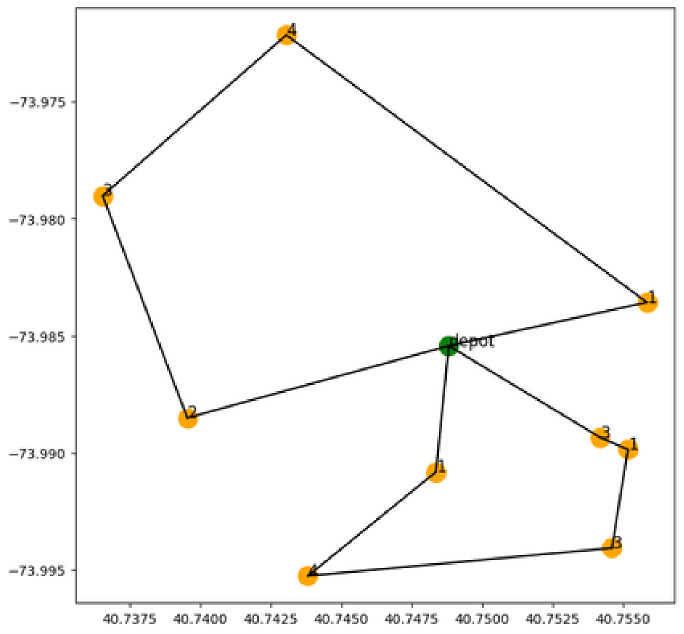
1. Parameters:
    a. Vehicles: 2
    b. Vehicle capacity: 40
    c. Node point: 15
    d. Depot: 1
    e. Demand: 0-10 employees
2. Euclidean distance method

OBSERVATIONS

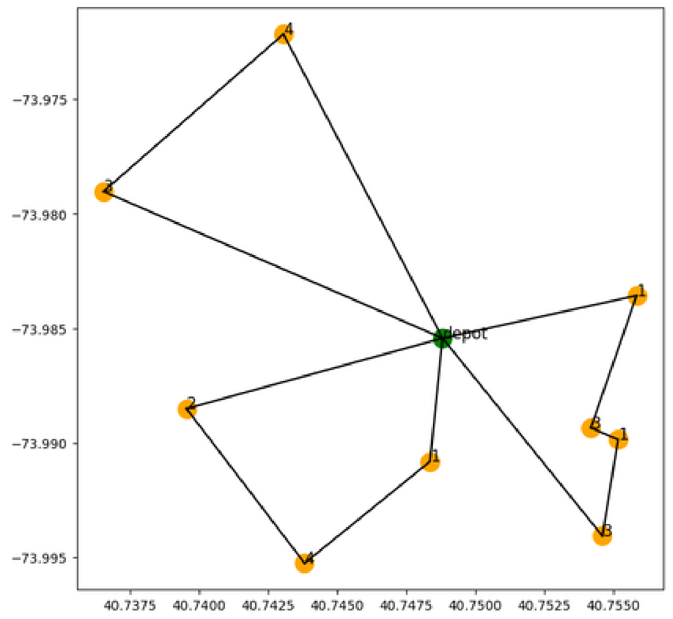
1. Low computational power:
    a. 10 nodes, 2 vehicles - instantaneous
    b. 10 nodes, 3 vehicles
    c. 14 nodes 2 vehicles
    d. 14 nodes 3 vehicles
    e. 15 nodes, 2 vehicles - 9 mins to process

PLOTS

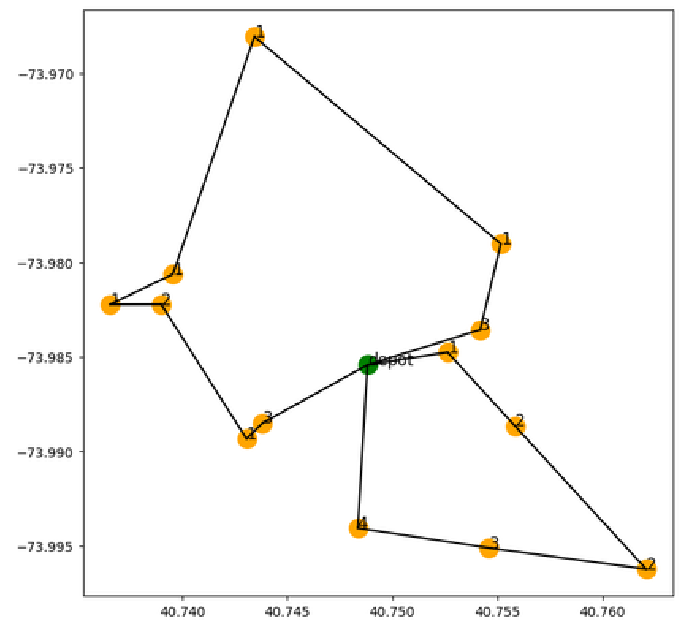
Plot (a)



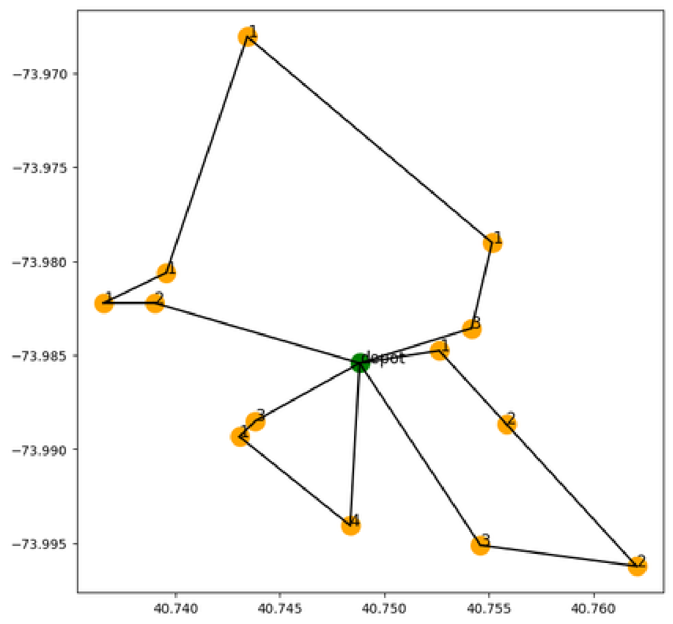
Plot (b)



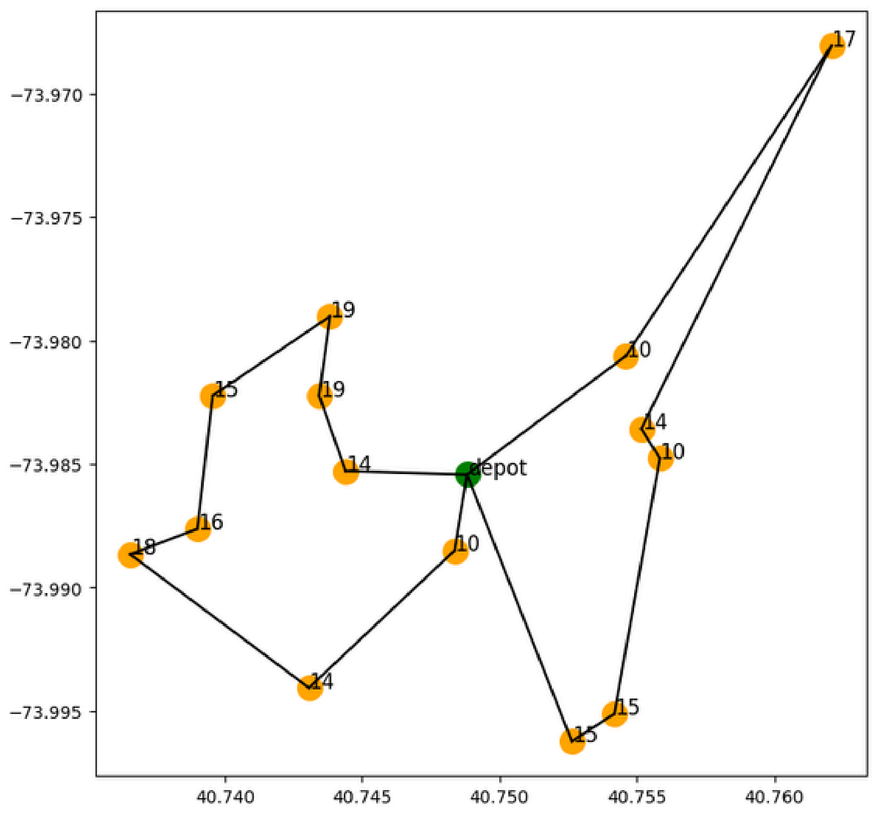
Plot (c)



Plot (d)



Plot (e)



Tabu Search

Initial solution using NNH → Randomly choose from 4 swap methods → Stopping criteria: based on iterations → Objective function: $\sum_{k=1}^{K}[d(k)+p(E_l(k))]$

Vertex reassignment, Vertex swap, 2-opt, Tail swap

Total iterations + max iterations without improvement + max runs

DATA

1. Parameters:
   a. Vehicles: 45
   b. Vehicle capacity: 46
   c. Node point: 136
   d. Depot: 1
   e. Demand: 0-20 employees
2. Geographic distance

OBSERVATION

Method 1: Tabu applied within a route

Node interchanges within a route:

1. Initial solution = NNH
   a. Best distance: **2469**

Method 2: Tabu applied across routes

4 methods of node interchanges across routes:

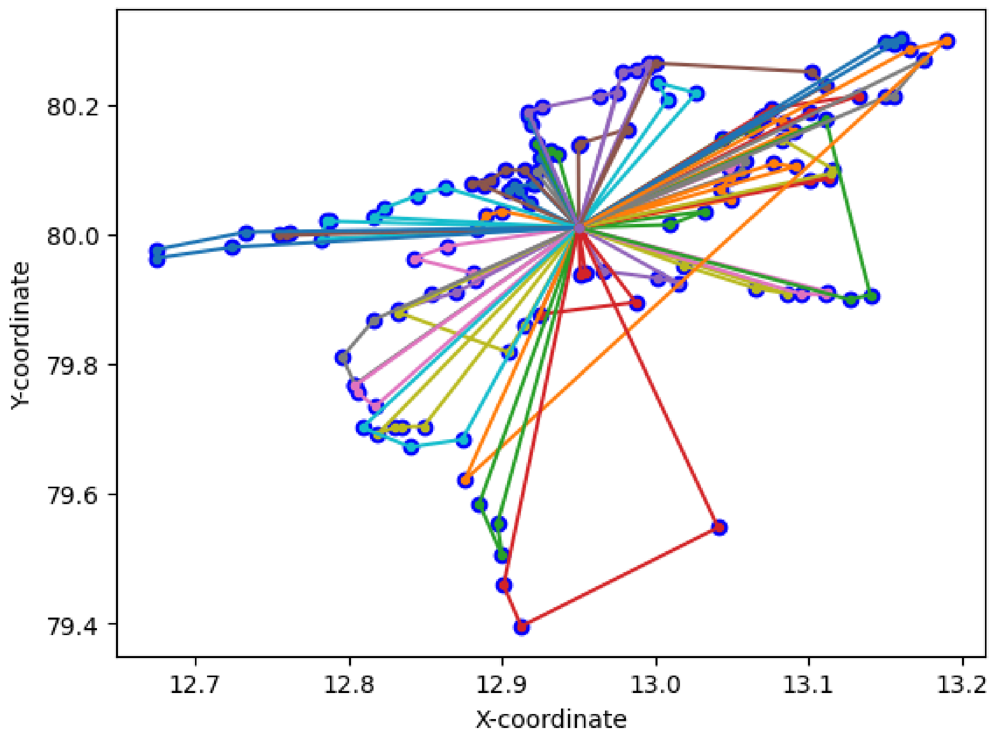
1. Number of iterations in one run = 500
2. For 1st run initial solution is NNH
   a. For consecutive runs initial solution = best solution of previous run
   b. Stops when improvement <= 5Km for 5 consecutive runs (after 14 runs)
   c. Best distance: **2210**

Method 3: Tabu with GA as initial solution

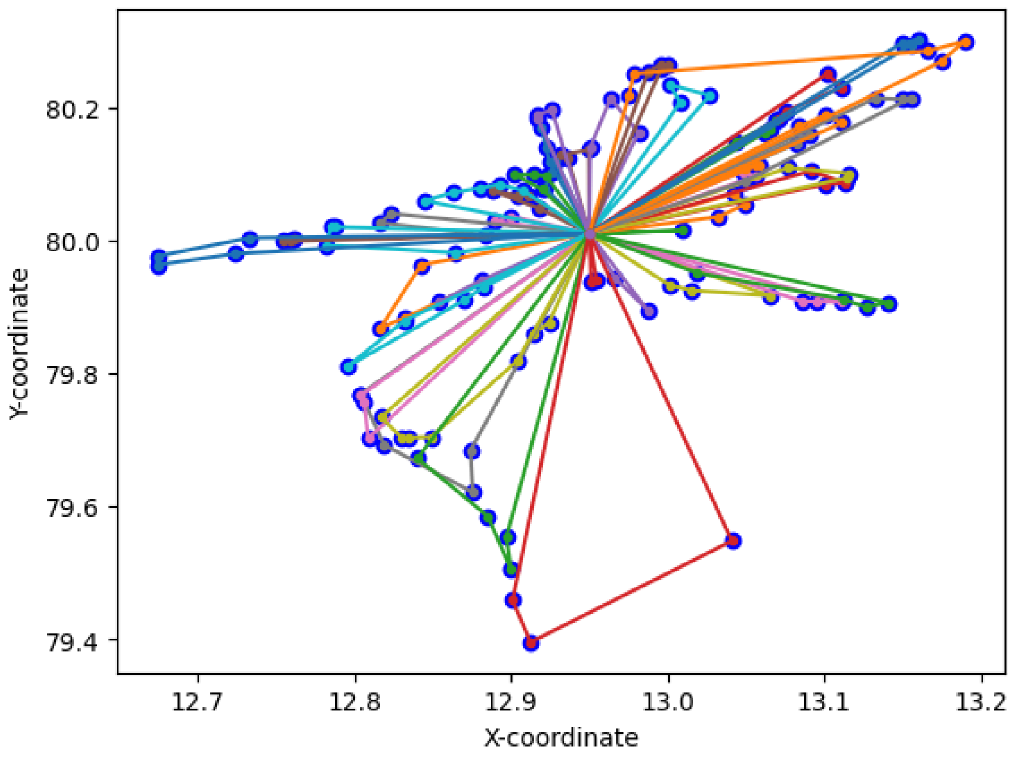
Similar to Method 2, except for the fact that:

1. Initial solution = Genetic algorithm solution generated via randomized initial Population
2. Best distance: **2193**

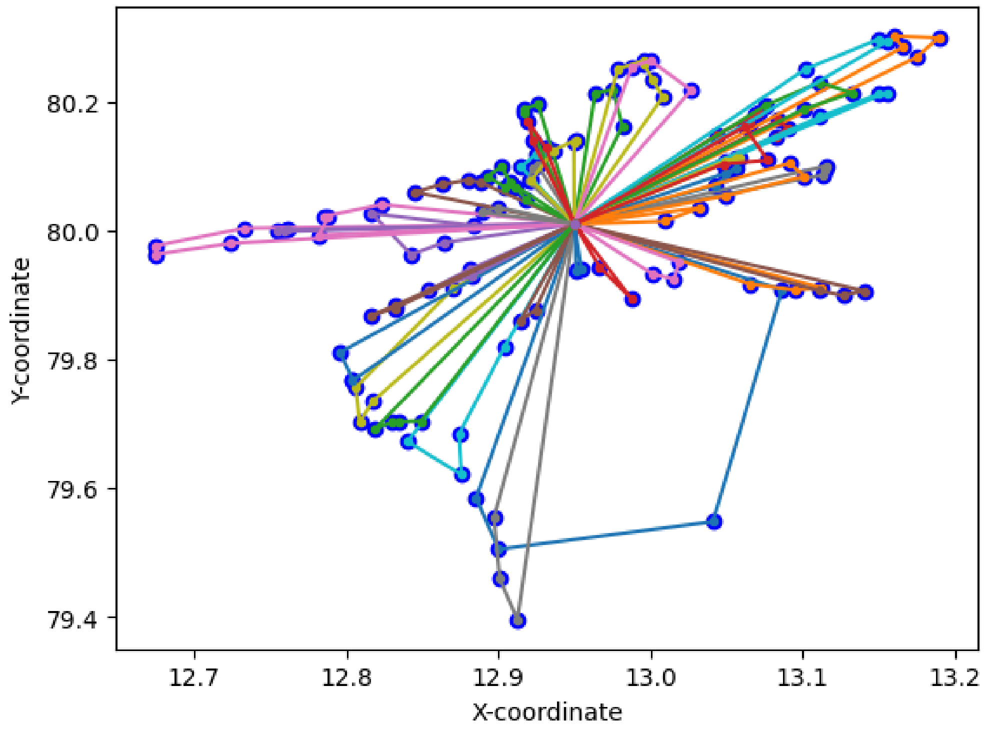
Method 1



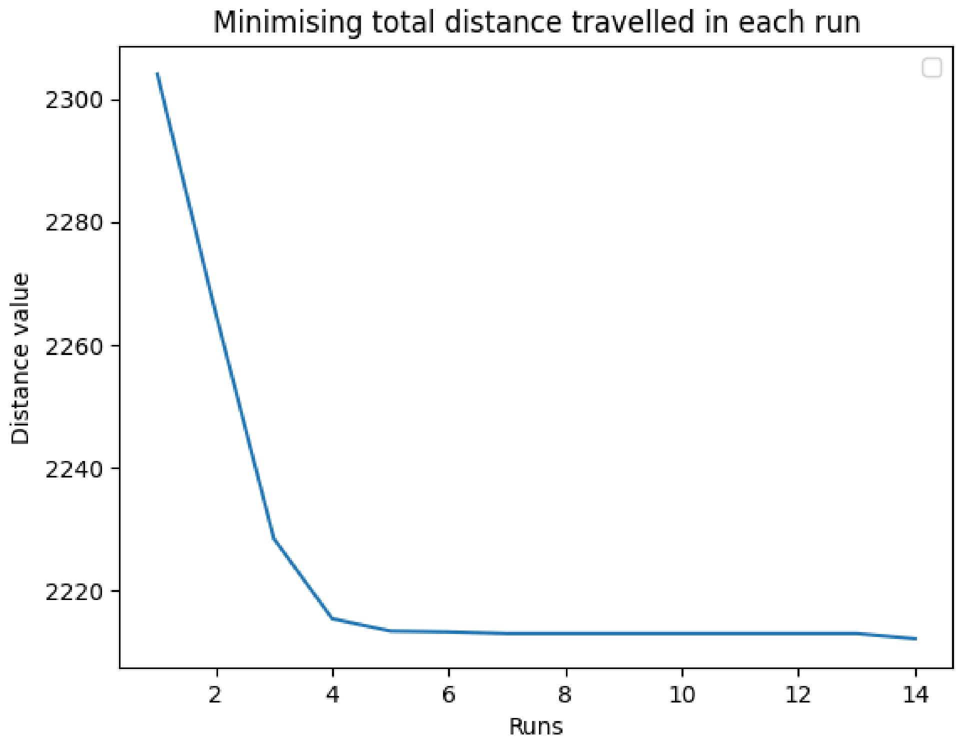
Method 2



Method 3



Convergence of Tabu Search



Genetic

Genetic population OR Randomly generate → Randomly select 2 parents → Offsprings: 2 point crossover and/ or mutation → Objective function: $\sum_{k=1}^{K}[d(k) + p(E_l(k))]$

Appends solutions which are > 3 times the best Tabu solution

Depends on crossover rate Duplicate nodes are corrected using NNH

Fitness(offspring) < Fitness (parent 1 or parent 2), Replace the max fitness of the population with offspring

DATA

1. Parameters:
   a. Vehicles: 45
   b. Vehicle capacity: 46
   c. Node point: 136
   d. Depot: 1
   e. Demand: 0-20 employees
2. Geographic distance between nodes
3. Number of generations: 500
4. Population size: ~10K

OBSERVATION

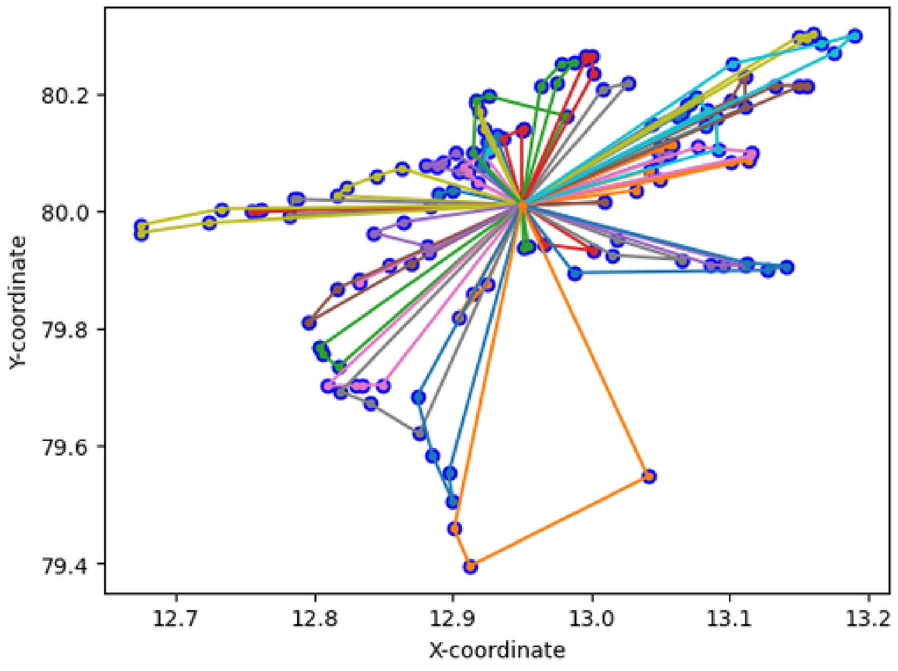
Method 1:
Population : Tabu search population
Best distance: **2210**

Method 2:
Population : Randomly generated & feasible
Best Distance: **4421** = 500 iterations
Best Distance: **4409** = 1000 iterations

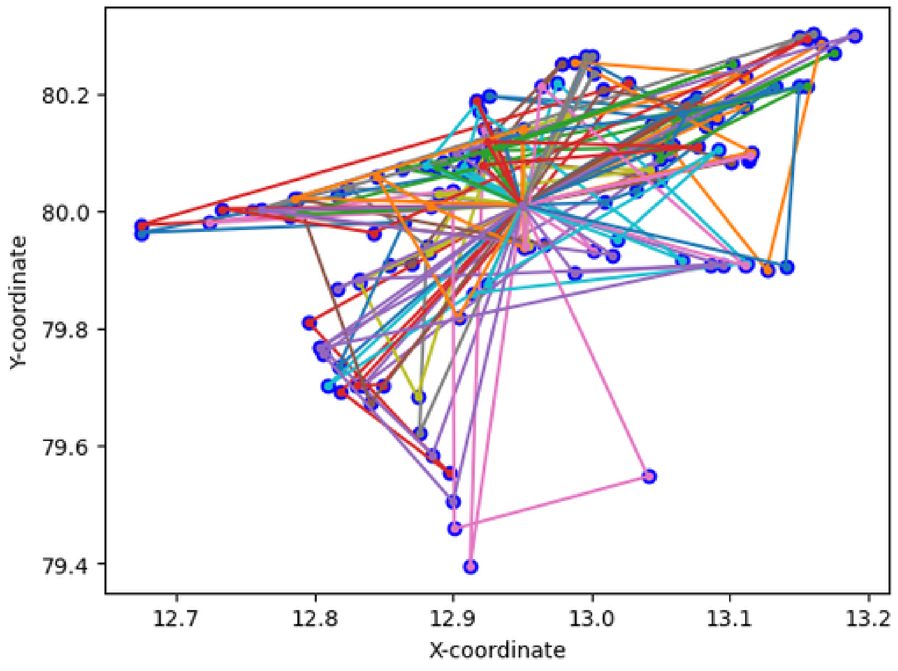
Method 3:
Pygad with randomnly generated population
Best distance: **4382**

PyGad with Tabu search as initial population
Best Distance: **2597**

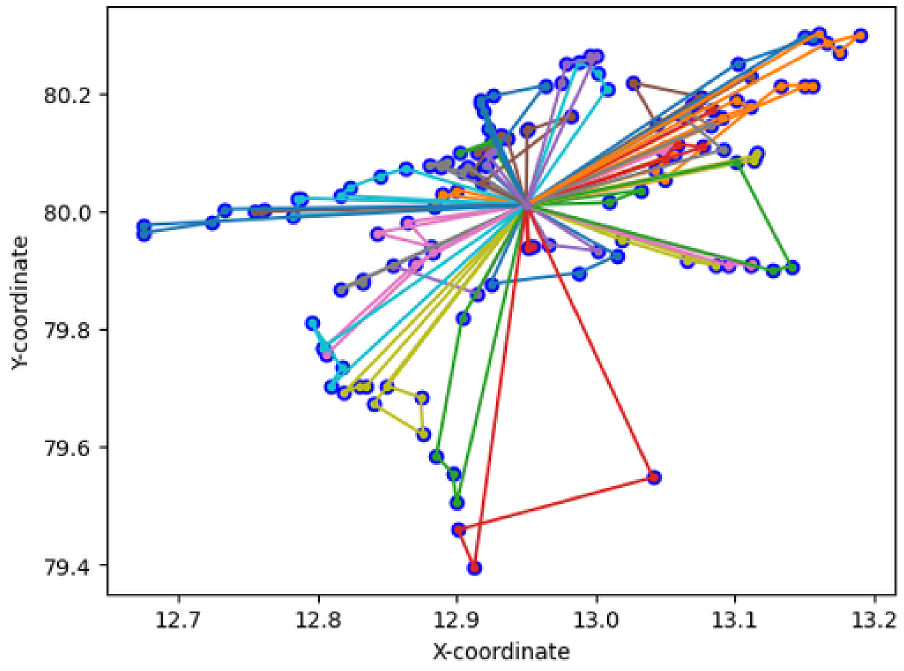
Method 1



Method 2



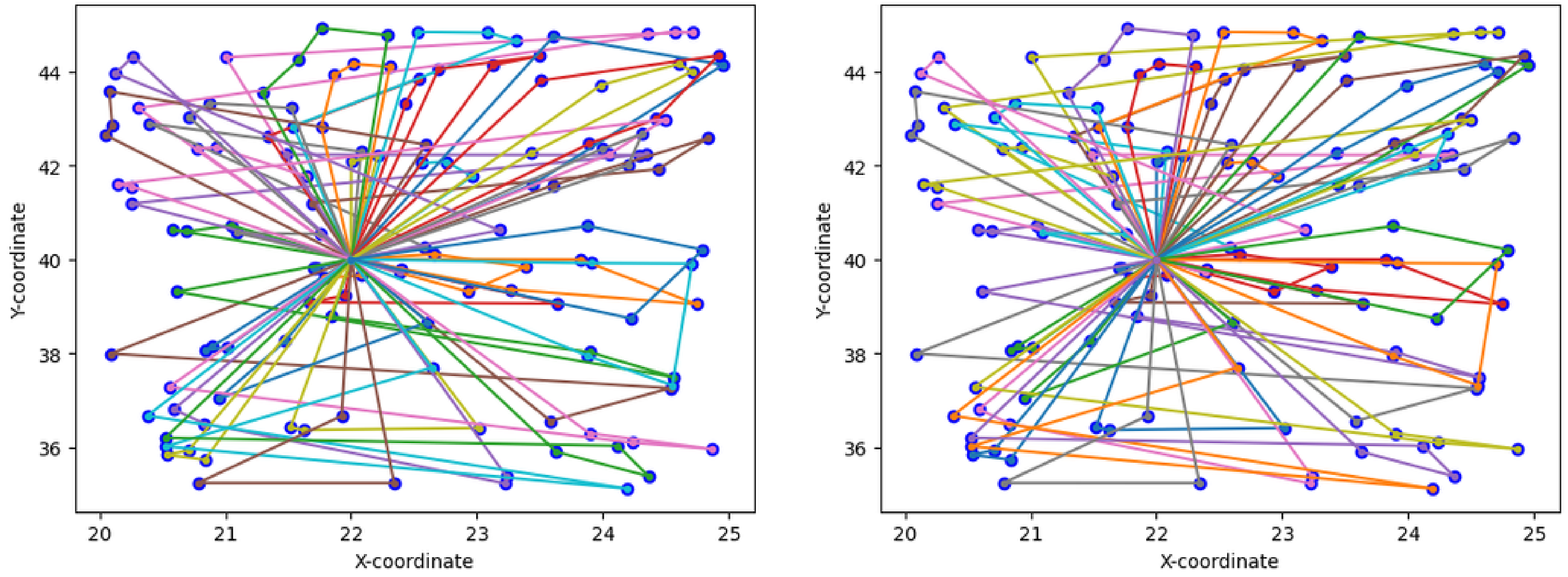
Method 3



Model comparison

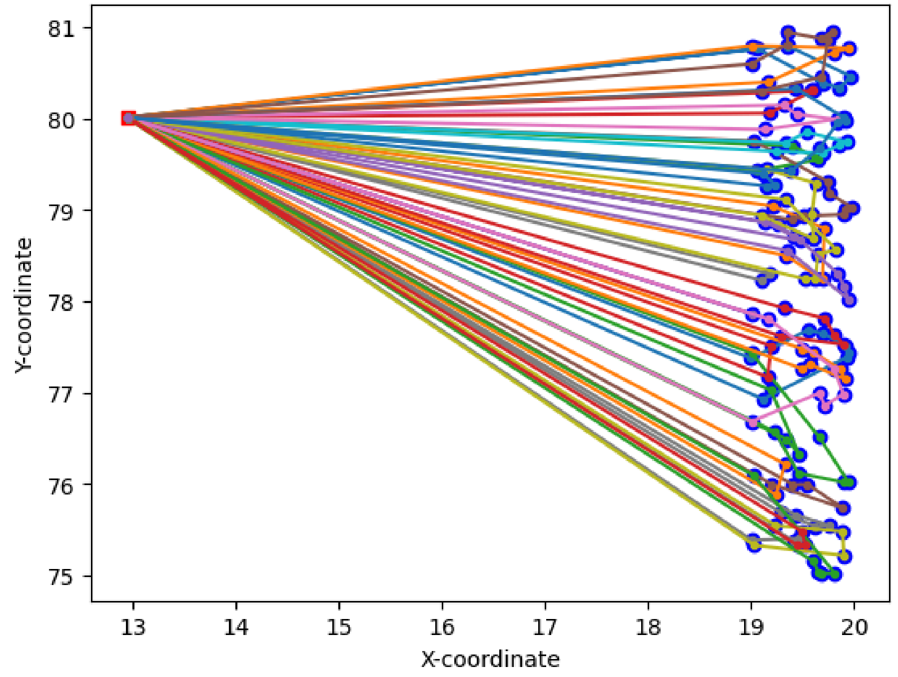
Inferences:

1. For the same result genetic algorithm took more time as compared to tabu search
2. Tabu search gave better results when GA solution was used as the initial solution instead of the NNH = Overall best performance model among all the P&Cs
3. GA gave better solutions when few of the optimal tabu search solutions were used as its genetic population.



When GA and Tabu search gave the same optimal solutions

Dataset No.	Tabu Search Soln.	Genetic Algo. Soln.
Original Dataset	2210	4421
data_1	20372	45066
data_2	40980	119453
data_3	56207	96482
data_4	32080	78741
data_5	50809	50809



One of the interesting plots where depot is far away from the employee settlement

Multi depot

PyGad Libary used with custom initial population → Nodes assigned to vehicles = 45 different routes → Each route picks up the best depot node → Objective function: to minimise distance

List of depot locations provided to the code

DATA

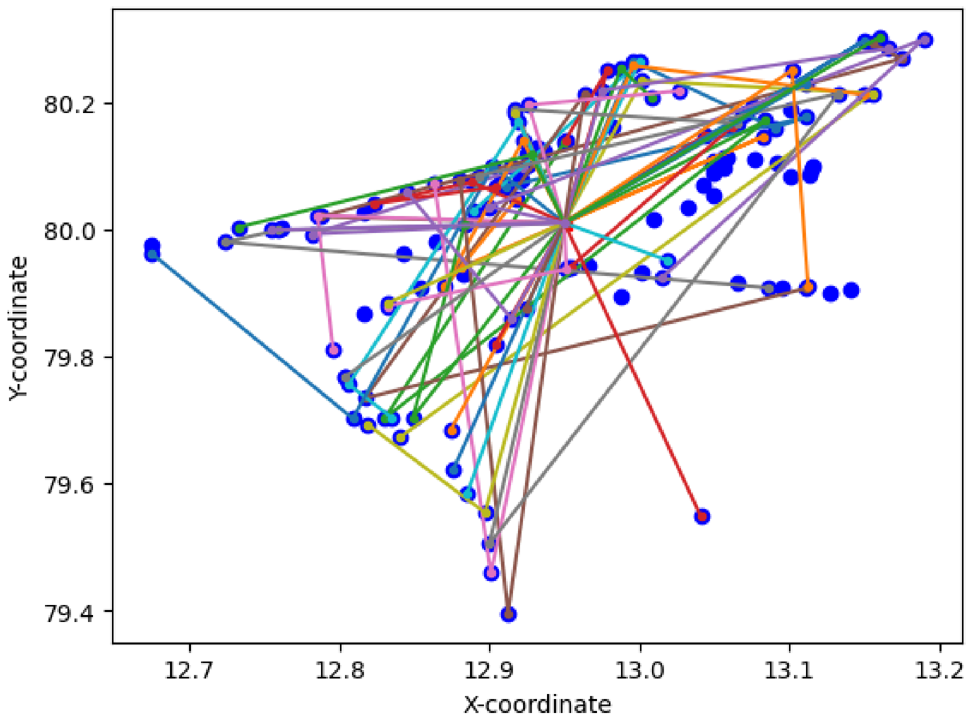
1. Parameters:
   a. Vehicles: 45
   b. Vehicle capacity: 46
   c. Node point: 136
   d. Depot: 45
   e. Demand: 0-20 employees
2. Geographic distance between nodes

```
# Create Genetic Algorithm instance
ga_instance_multi = pygad.GA(num_generations=10,
                             num_parents_mating=10,
                             fitness_func=fitness_func_multi,
                             sol_per_pop=50,
                             num_genes=num_nodes,
                             gene_type=int,
                             parent_selection_type="tournament",
                             K_tournament=10,
                             crossover_type="two_points",
                             crossover_probability = 0.8,
                             initial_population = custom_initial_population,
                             mutation_type="swap",
                             mutation_percent_genes=10,
                             mutation_num_genes=2,
                             mutation_probability = 0.8,
                             keep_parents=0)
```

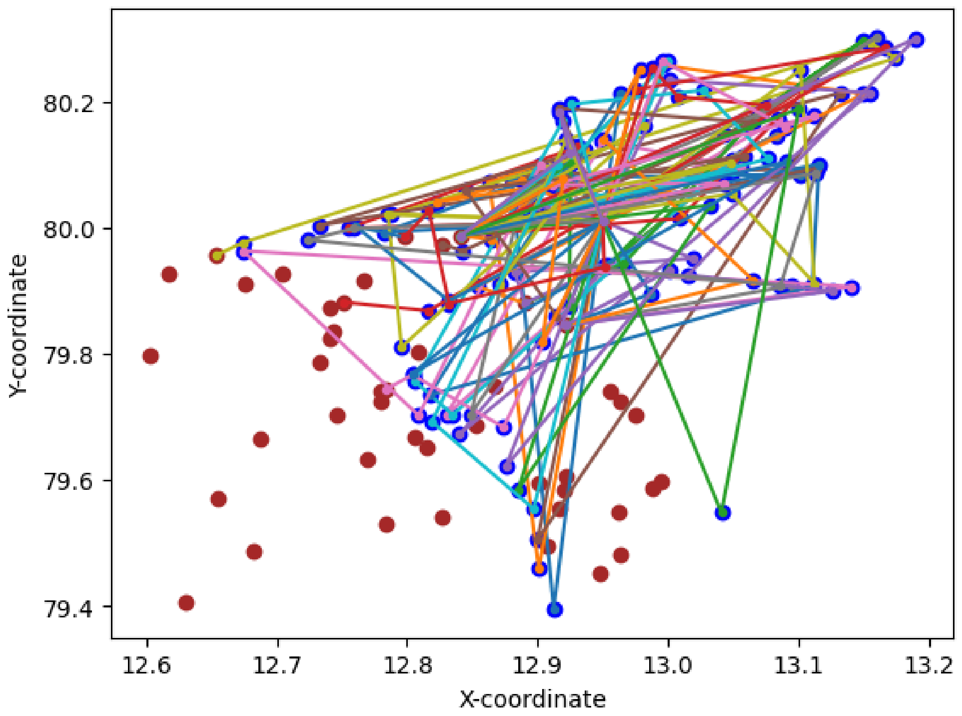
OBSERVATIONS

Method 1:
Depots: Starting points of a route

Best distance: **4131**



Method 2:
Depots: Randomly generated points
Each route chooses the best possible depot such that the total route distance is minimised for its journey.
Best distance: **4639**



Stochastic VRP

Problem definition:

1. Each node is a customer with some demand.
2. Demand at each location is not known at the time the tour is designed but is known to follow a certain probability distribution. E.g. Normal, uniform
3. One vehicle with some fixed capacity is used. The vehicle has to come back to the depot in case of restocking.

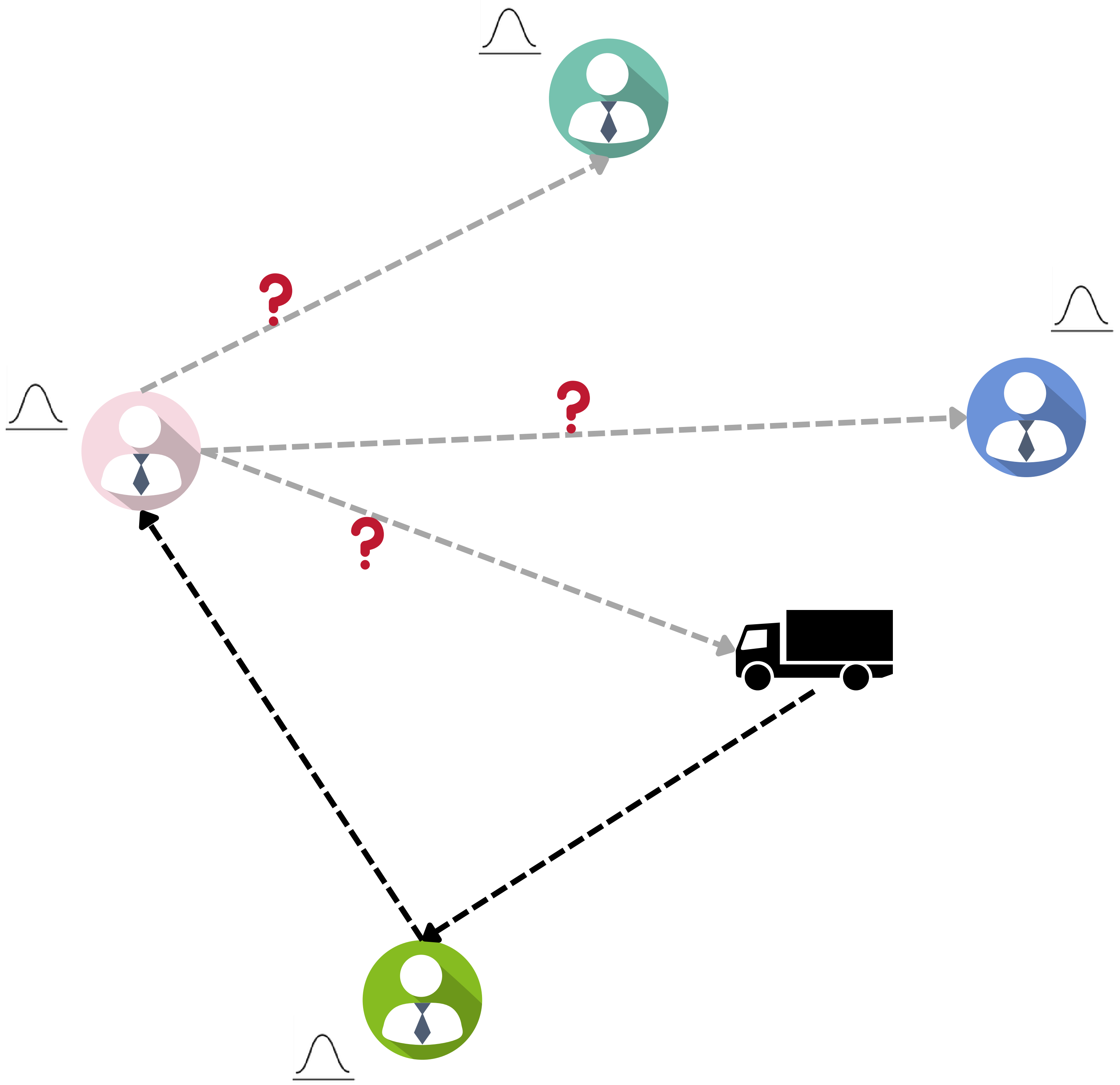
Why can't the routes be designed after knowing the demand:

1. Not enough resources
2. Effort does not justify the gained benefits
3. Other priorities: Regularity/ personalization of service (by having the same vehicle & driver visit the same customer)

Difficult to learn the demand on a day before actually visiting the customer

Applications:

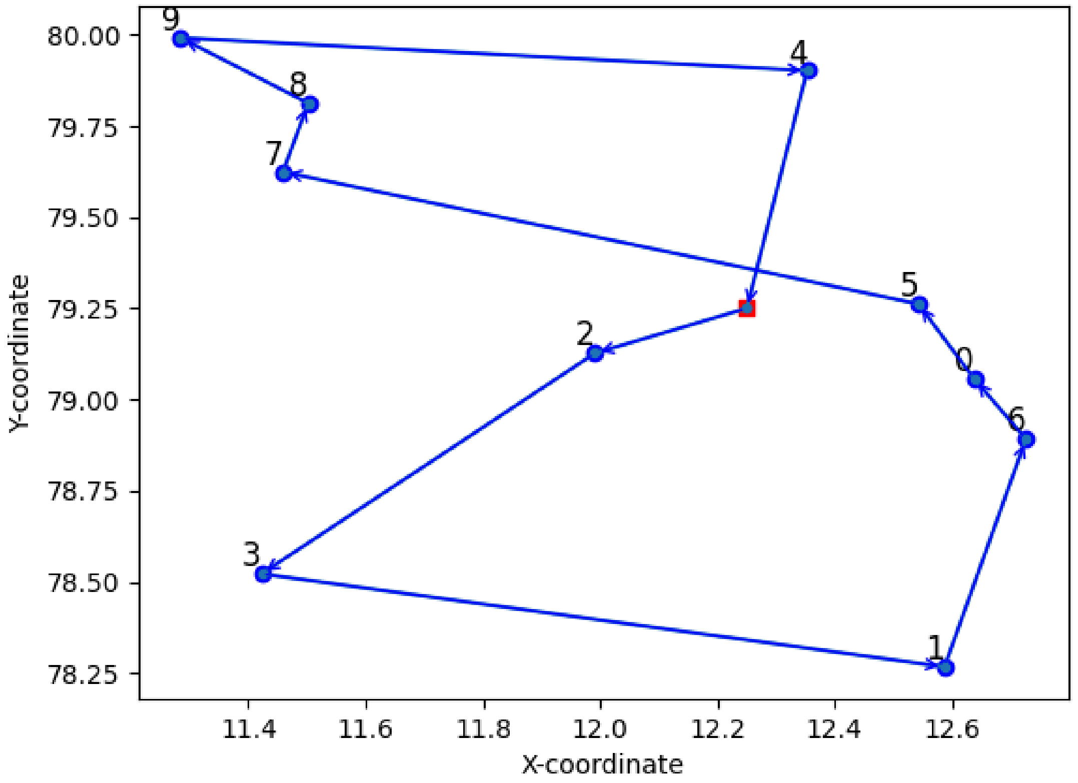
- Central bank collect money from branches
- Post office distribution of pacakges
- Retail logistics - demand at stores/ warehouses
- Healthcare logistics



Method 1 : Demand based routing

DATA

1. Parameters:
   a. Vehicles: 1
   b. Vehicle capacity: 50
   c. Node point: 10
   d. Depot: 1
   e. Demand: [10 to 20] with equal probability
2. Geographic distance between nodes



Priori customer sequence

Steps

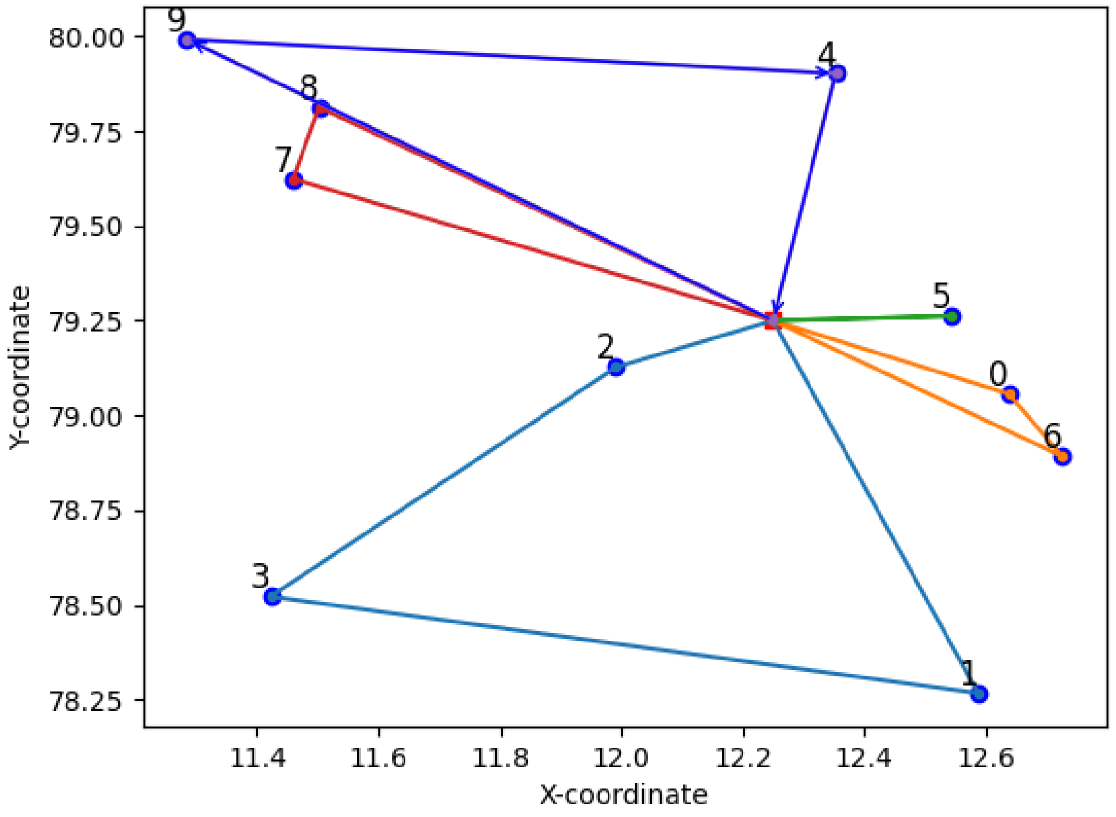
Determine a fixed priori sequence among all customers based on previosuly known demand:

1. Most occuring demand value
2. Mean demand value

Dealing with stochasticity:

1. Strategy A: when customer demands are not known before hand
   a. Visit all nodes in priori sequence return back to depot and start again if demand > capacity of vehicle
2. Strategy B: customer demands are known before starting the tour
   a. Apply optimality of VRP for nodes with D > 0 and skip the rest, return to depot and start again if demand > capacity of vehicle

After introducing stochasticity



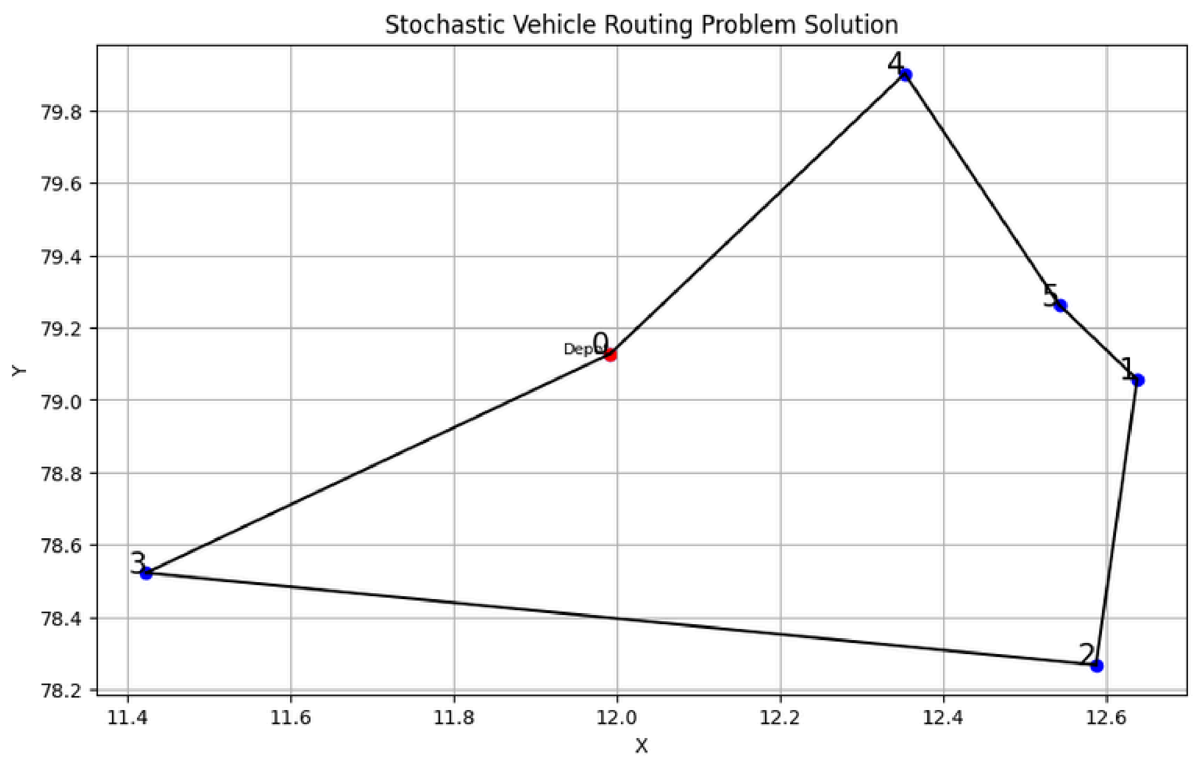
Method 2.1 : EV with scenario probabilities

DATA

1. Customers = 5; Depot = 1
2. Vehicle = 1; Vehicle Capacity = 100;
3. Lower bound = 20 vehicle cannot approach next node with items <=20)
4. Demand scenarios = 3, with respective probabilty of each:
   a. scenarios = [[10, 20, 15, 25, 30], [15, 25, 20, 30, 35], [20, 30, 25, 35, 40]]
   b. probabilities = [0.1, 0.4, 0.5]

Steps

1. Distance matrix:
   a. Node to node direct travel cost is same as distance
   b. Node to depot travel cost in order to refill is double the travel distance - due to logistics considerations
   c. E.g. : Depot = D
      i. If optimal route is D-A-B-D ------- Cost = D-A + A-B + B-D
      ii. If optimal route is D-A-D-B-D ---- Cost = D-A + 2(A-D) + 2(D-B) + B-D
2. As there are only 5 nodes. 5! = 120 permutations are tested.
3. **Minimisation function: Travel cost + Probability of a scenario * additional cost for that scenario**
4. Output gives the sequence of nodes in which the vehicle should be travelling for incurring minimal cost



Scenarios & their probabilities based method

```
Total cost: 883.6159378512295
Best route: (0, 3, 2, 1, 5, 4, 0)
```

Method 2.2 : EV with node probabilities

```
node_parameters = {
    1: {'mean': 40, 'std_dev': 5},
    2: {'mean': 50, 'std_dev': 7},
    3: {'mean': 30, 'std_dev': 10},
    4: {'mean': 25, 'std_dev': 4},
    5: {'mean': 35, 'std_dev': 9}
}
```

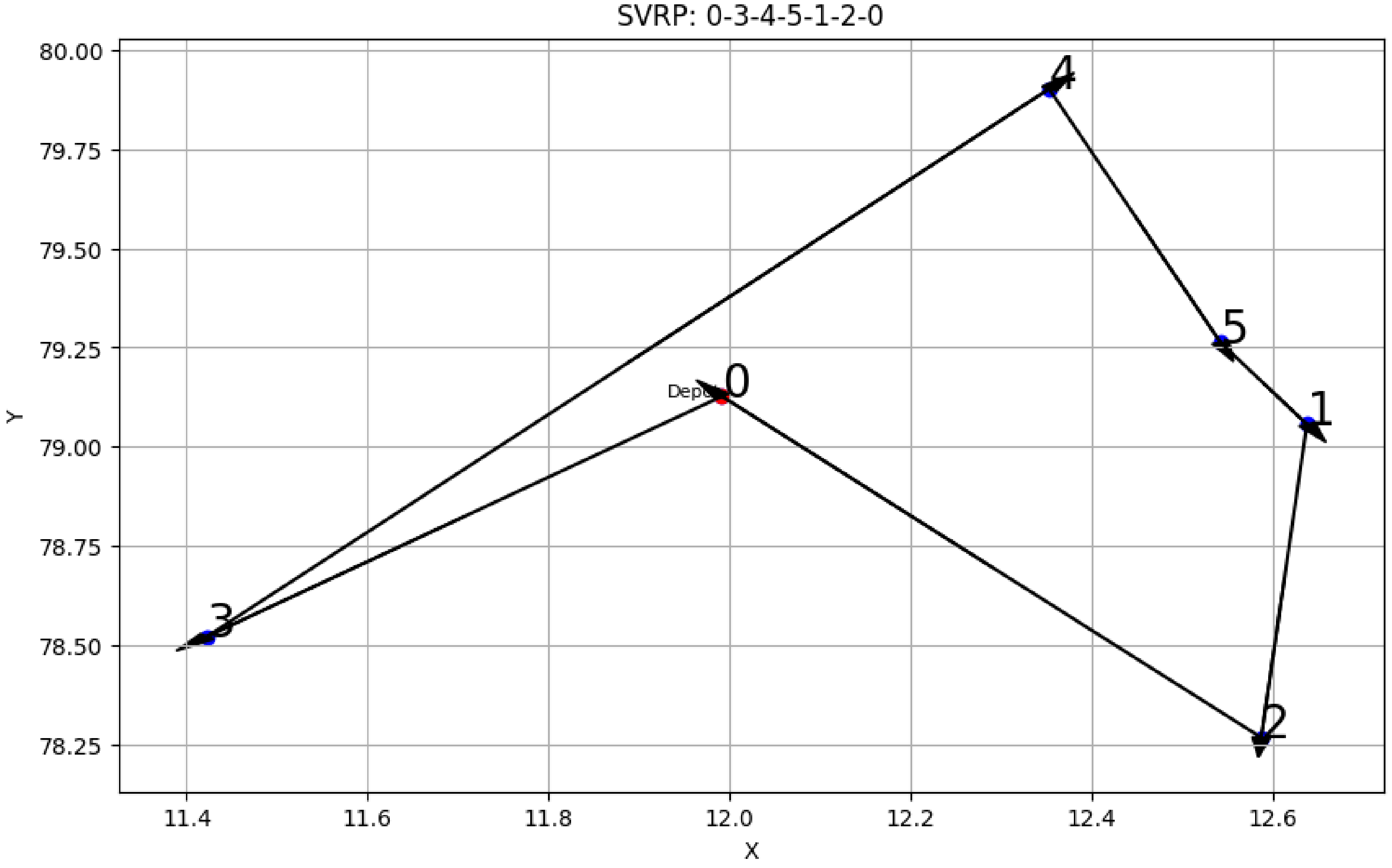
Demand = normal distribution

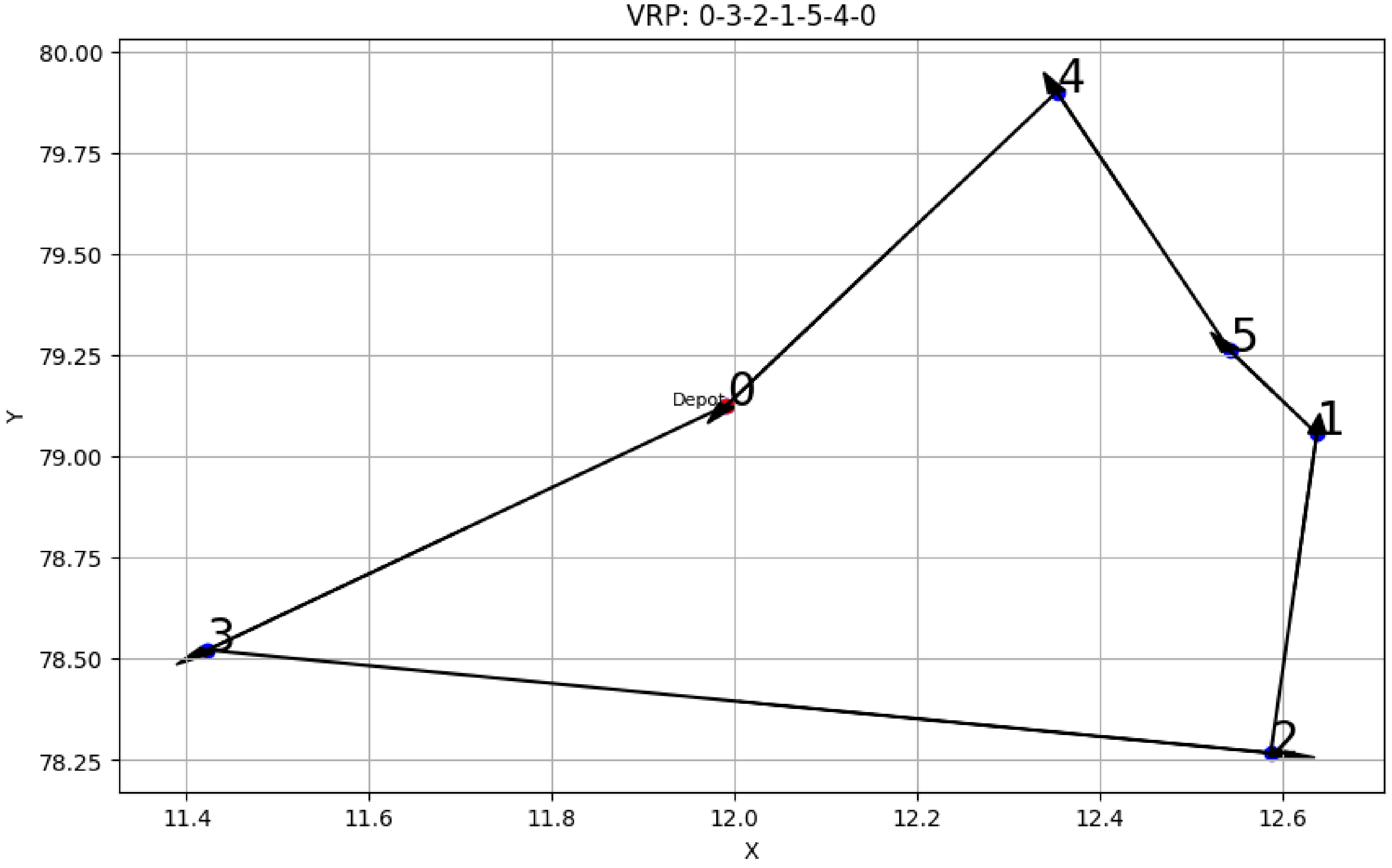
→ Demand scenarios created for each node and probability is calculated
E.g. For Node 1: Demand = 40, probability = 0.5

→ JPD = MULTIPLY(Node probabilities)
E.g. [40,50,30,25,35] then
JPD = 0.5^5 = 0.03125
Calculated for ~200 scenarios

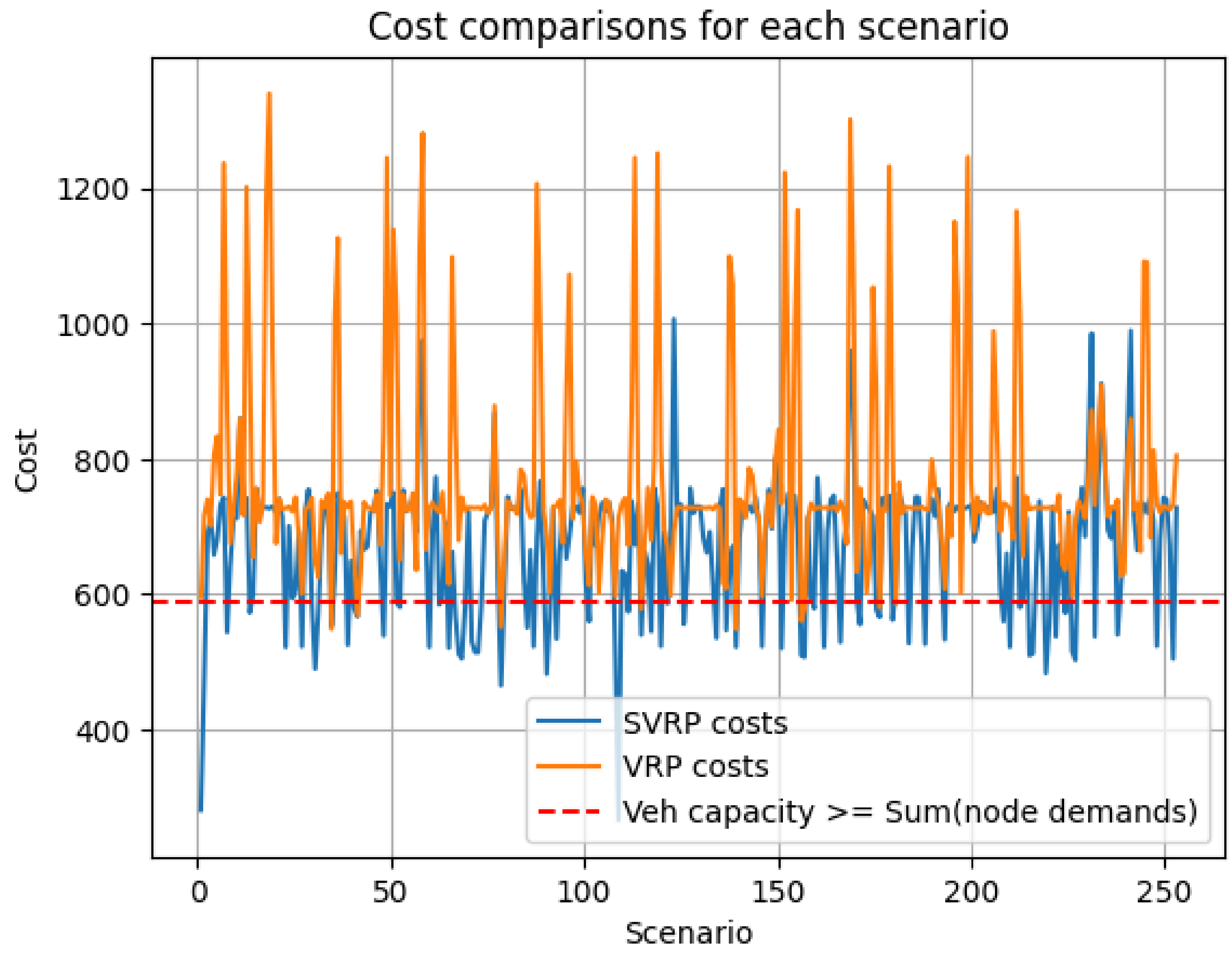
→ Normalisation is done such that the SUM (probabilities of all scenarios) = 1

Compare between SVRP Node sequence Vs. VRP Node sequence (calculated using mean demand values)





Cost savings: SVRP vs. VRP across scenarios

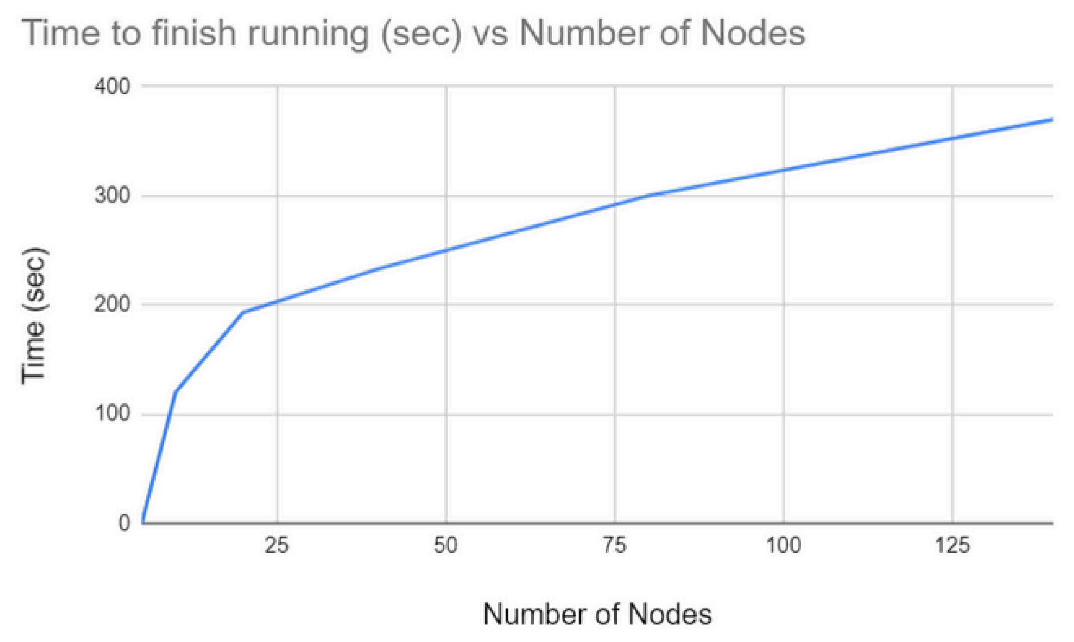


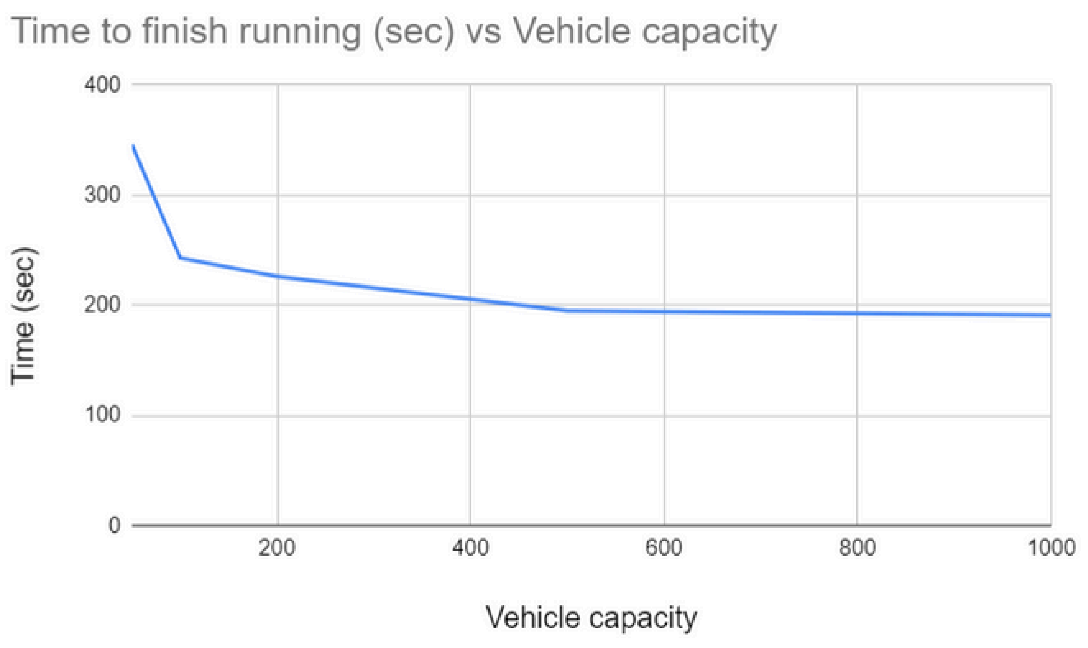
SVRP solution outperforms the VRP for **~51%** of the scenarios

SVRP solution gives similar cost as VRP for **~47%** of the scenarios

VRP solution outperforms SVRP for mere **~2%** of the scenarios

Algorithm performance





% scenarios where SVRP outperforms VRP

Dataset No.	% Scenarios
Original Dataset	~95%
data_1	~97%
data_2	~92%
data_3	~96.5%
data_4	~97.3%
data_5	~96%

Model Pipeline

Tabu Search Algorithm top 100 solutions → Fed in SVRP for the minimisation function → Compared with VRP solution

Extending to Employee Pick Up

Problem Statement

How?

Problem: Assigning airline check-in employees to tasks related to departing flights under uncertain circumstance at an international terminal of a large airport.
Why? The uncertainty of flight departing time mainly stems from delays and traffic control.
Objective: minimizes both staffing costs and risk measurement to respond the time uncertainty

Models:

- 2-stage stochastic model with expected value basis.
- Risk-averse model includes CVaR alongside total expected cost.

Heuristics:

- Sample Average Approximation Method: It approximates the expected objective function, enabling solution with deterministic programs using commercial solvers.
- Decomposition into scenario sub-problems, iteratively updating solutions and dual prices for near-optimal solutions

---

Problem: People Scheduling Service, an Employee Scheduling Algorithm based on Stochastic Workloads at Picnic Technologies - supermarket
Why? The demand for groceries and the workload associated with tasks such as picking and delivering orders can vary unpredictably due to factors like customer preferences, seasonal fluctuations, and unforeseen events.
Objective: minimize expected costs, considering factors such as lost revenue due to incomplete workload, excess personnel costs from overcapacity, and employee satisfaction costs incurred by unfavorable schedules

Models:

1. Mixed integer linear programming: Formulating the scheduling problem as a MILP and solving it directly using an off-the-shelf MILP solver. Will face scalability issues
2. L-shaped algorithm: Two stage stochastic optimisation problem which solves a master problem and subproblems. The master problem incorporates employee satisfaction and scheduling constraints. The subproblems utilise solution of each iteration of master problem alongwith scenario uncertainity data.

Extending to Employee Pick Up

Problem Statement

How?

Problem: The Stochastic Electric-Vehicle Relocation Problem with Dynamic Pricing
Why? The evolving demand for flexible car-sharing services requires the operator to optimize vehicle distribution and profits to address uncertainties in user demand and vehicle availability.
Objective: maximize expected profit considering vehicle relocation costs, charging constraints and revenue from pricing strategies

Models:

1. The SE-VReP-DP model employs a two-stage linear stochastic mixed-integer programming approach. It integrates vehicle relocation and pricing decisions. In the first stage, decisions are made on relocating vehicles and setting prices before customer preferences are known. The second stage incorporates customer behavior scenarios using Sample Average Approximation (SAA) and a Random Utility model to handle uncertainty in preferences.

Problem: A stochastic optimization approach to shift scheduling with breaks adjustments
Why? To efficiently schedule shifts and breaks for service industry workers under uncertain demand.
Objective: minimize total working hours and expected uncovered demand

Models:

1. Two-stage stochastic optimization: In first stage shifts are assigned to workers without knowing the exact demand for each time period. In the second stage, decisions are made after observing the realized demand. Each scenario represents a daily realization of demand with equal probabilities.
2. The Recourse Problem (RP) aims to minimize the expected cost under the true demand distribution, representing the optimal "here-and-now" solution corresponding to stage 1. Additionally, the Wait-and-See Solution (WS) involves solving sub-problems by scenario, assuming perfect information about future demand.

Extending to Employee Pick Up

The need for employees in a factory is influenced by:

1. Factory needs: Demand for that particular month or shift of the day
2. Employee needs: Certain employees might be on leave owing to personal reasons or external reasons, the factory operations should be running even then and re-scheduling with available employees need to be done

These factors influence the number of employees to be picked up from each node. Thus the above problem statements can be extended to the employee pick up problem.

Future work

1. Expand models to capture real-world complexities
    a. Develop algorithms for on-the-fly route adjustments and enable models to adapt to changing conditions seamlessly
    b. Incorporate multiple objectives: Time windows, heterogeneous fleets, driver availability, shift scheduling
2. Integrate advanced optimization techniques
    a. Explore machine learning, deep learning, and reinforcement learning
    b. Balance conflicting objectives (costs, customer satisfaction, environmental impact)
3. Industry standards:
    a. Develop models by incorporating industry specific standards
    b. Develop user-friendly decision support systems, Intuitive visualizations and interactive interfaces

Thank you