Optimization of Capacitated Vehicle Routing Problem using KNN based Branch-and-Cut Approach

Under the Supervision of Prof. Pitam Singh
Presented by: Srima Bose

M.Sc. Mathematics and Scientific Computing
Motilal Nehru National Institute of Technology, Allahabad

May 22, 2025

Introduction to Vehicle Routing Problem (VRP)

- VRP is a classical combinatorial optimization problem in logistics.
- Objective: Determine optimal routes for a fleet of vehicles to serve a set of customers.
- Constraints include vehicle capacity, delivery time windows, and route length.
- First introduced by Dantzig and Ramser in 1959.
- VRP aims to minimize total delivery cost or distance while satisfying customer demands.
- Widely applicable in transportation, distribution, and supply chain management.

Diagram of VRP Application

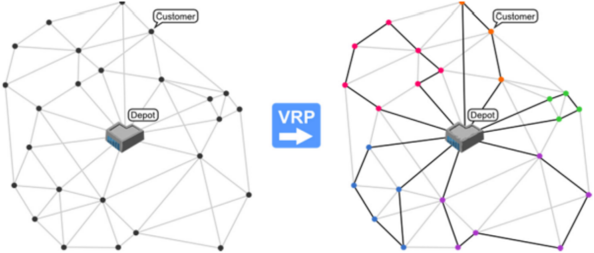


Figure: VRP Application

Types of Vehicle Routing Problems (VRP)

- **Capacitated VRP (CVRP):** Vehicles have limited capacity.
- **VRP with Time Windows (VRPTW):** Deliveries must be made within specific time frames.
- **VRP with Pickup and Delivery (VRPPD):** Goods are picked up and delivered to different locations.
- **Open VRP:** Vehicles do not return to the depot after completing routes.
- **Multi-Depot VRP (MDVRP):** Multiple depots are used for dispatching vehicles.

VRP Models and Industry Applications

- **Capacitated Vehicle Routing Problem (CVRP):**
  - Used in logistics, parcel delivery, and retail distribution.
  - Example: Amazon, FedEx delivery trucks with limited capacity.
- **VRP with Time Windows (VRPTW):**
  - Applied in courier services and home healthcare.
  - Example: Delivering packages within promised time slots.
- **Pickup and Delivery VRP (PDVRP):**
  - Used in ride-sharing, waste collection, and courier return services.
  - Example: Uber, Lyft route planning; recycling services.
- **Multi-Depot VRP (MDVRP):**
  - Employed in companies with multiple warehouses or depots.
  - Example: Large retail chains managing multiple distribution centers.
- **Open VRP:**
  - Used when vehicles don't return to the depot, e.g., one-way trips.
  - Example: Food delivery services and some public transport routing.

What is Capacitated VRP (CVRP)?

Problem Definition

A special case of VRP where:

- Each vehicle has limited capacity Q
- Each customer must be visited exactly once
- Route demand $\leq Q$ (capacity constraint)
- Minimize total travel cost/distance

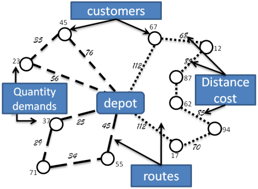


Figure: CVRP

Why Choose CVRP for My Project?

- CVRP is a fundamental and widely applicable VRP variant with strong real-world relevance.
- Many industries rely on capacitated vehicles, making this problem highly practical.
- CVRP balances complexity and tractability — challenging enough to innovate, yet solvable with hybrid methods.
- Existing research provides a strong foundation, enabling meaningful comparison.
- Opportunity to improve optimization by combining ML (KNN) and Branch-and-Cut algorithms.
- Potential for significant cost savings and efficiency improvements in logistics.
- Personal interest in tackling complex combinatorial optimization problems.

Historical Background of VRP

- The Vehicle Routing Problem (VRP) was first introduced by Dantzig and Ramser in 1959.
- Initially formulated to optimize gasoline deliveries to service stations.
- Early studies focused on exact methods like branch-and-bound and linear programming.
- Over time, heuristics and metaheuristics (e.g., genetic algorithms, simulated annealing) became popular to solve larger instances.
- The VRP has evolved into multiple variants addressing real-world constraints (e.g., time windows, multiple depots).
- Recent advances include incorporating AI and Machine Learning for improved solution quality and speed.

Evolution from VRP to CVRP

- VRP was introduced to optimize routing of vehicles serving multiple customers.
- Basic VRP focused on minimizing total distance or cost without vehicle capacity limits.
- Real-world constraints led to development of Capacitated Vehicle Routing Problem (CVRP).
- CVRP adds vehicle capacity limits, ensuring demand assigned to a route doesn't exceed vehicle load.
- CVRP better models logistics challenges in delivery, distribution, and supply chain industries.
- This evolution allows more practical and applicable routing solutions.

Classical Solution Techniques

Clarke-Wright Savings Algorithm

- A heuristic method introduced in 1964.
- Begins with one route per customer and merges routes based on cost savings.
- Efficient for generating good initial solutions for VRP and CVRP.

Sweep Algorithm

- Uses polar coordinates to group customers by angle from the depot.
- Routes are formed by "sweeping" through customers in angular order.
- Effective for geographically clustered customer locations.

Clarke-Wright Savings Algorithm - Flowchart

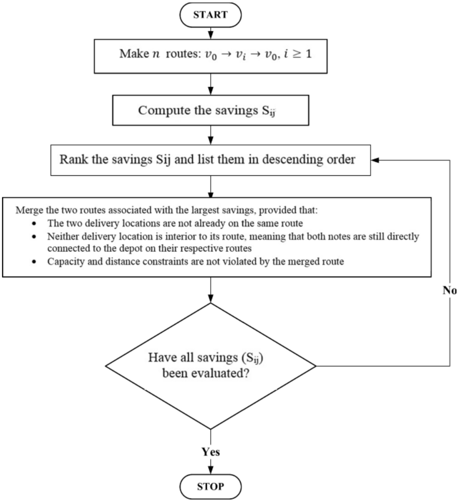


Figure: Clarke-Wright Savings

Sweep Algorithm - Flowchart

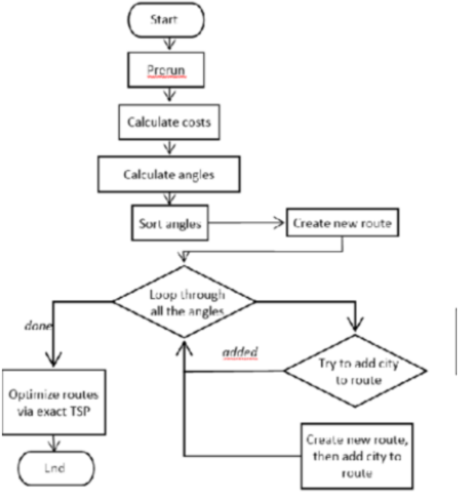


Figure: Sweep

Literature Review I

- **Sakhri et al. (2022)** proposed a hybrid clustering algorithm using K-Means and KNN to solve the CVRP [1].
- **Konstantakopoulos et al. (2020)** developed a hybrid method combining clustering and routing heuristics [2].
- **Dorigo and Gambardella (1997)** introduced Ant Colony System, which has inspired many metaheuristics for routing problems [3].
- **Hassanzadeh et al. (2022)** presented a hybrid Ant Colony Optimization and Simulated Annealing technique for CVRP [4].
- **Zhao et al. (2022)** designed a hybrid Genetic Algorithm for CVRP with time windows [5].
- **Liu et al. (2022)** used a Hybrid Particle Swarm Optimization approach for CVRP [6].
- **Sharma and Gupta (2023)** combined Genetic Algorithm with local search to effectively solve CVRP [7].

Mathematical Approaches for CVRP

Branch-and-Bound

- Systematically explores all possible routes by dividing the problem into smaller subproblems (branching).
- Uses bounds to eliminate suboptimal solutions early, reducing computation time.
- Guarantees optimal solution but may be computationally expensive for large instances.

Brach-and-Cut

- Enhances Branch-and-Bound by adding cutting planes (linear inequalities) to tighten the formulation.
- Efficiently solves medium-sized instances of CVRP.
- Particularly effective when combined with modern solvers like CPLEX and Gurobi.

Machine Learning in Routing Problems

K-means Clustering

- Groups customers based on geographical proximity.
- Reduces problem size by creating smaller, localized subproblems.
- Improves computational efficiency before routing optimization.

K-Nearest Neighbours

- Identifies the closest nodes (customers) to construct greedy routes.
- Works well as a heuristic initializer or for prioritizing node visits.
- Helps in estimating neighborhood structure to reduce computation.

Summary of Literature Gaps

- Most classical methods (e.g., Clarke-Wright, Sweep) are efficient but fail to scale well for large, real-world CVRP instances.
- Exact methods (Branch-and-Bound, Branch-and-Cut) guarantee optimality but become computationally expensive as problem size increases.
- Metaheuristics like Genetic Algorithms and Ant Colony Optimization offer good solutions but require fine-tuning and may lack consistency.
- Limited studies integrate Machine Learning techniques with exact methods for CVRP.
- Most ML-based approaches focus on clustering or prediction, but not on full-route optimization in tandem with traditional solvers.
- Few works evaluate the effectiveness of combining KNN with Branch-and-Cut for route construction and optimization.

These gaps motivated our research to explore a hybrid ML + exact approach for scalable and efficient CVRP solutions.

Problem Definition of CVRP

Objective: Design optimal routes for a fleet of vehicles to deliver goods to a set of customers, minimizing total travel cost while satisfying capacity constraints.

Given:

- A central depot.
- A set of n customers, each with a known demand.
- A fleet of k identical vehicles with limited carrying capacity.
- A distance matrix between all locations.

Constraints:

- Each customer is visited exactly once by one vehicle.
- The total demand on any route must not exceed vehicle capacity.
- All vehicles start and end at the depot.

Goal: Minimize the total distance traveled by the vehicles.

Dataset Used: A-n33-k5

Dataset Name

A-n33-k5 (from Augerat's Set A – CVRP benchmark)

Key Characteristics:

- **Number of Customers:** 33
- **Number of Vehicles:** 5
- **Vehicle Capacity:** 100 units
- **Depot Location:** Node 1
- **Customer Data:** Coordinates, demand, service constraints
- **Objective:** Minimize total distance while meeting capacity constraints

Why This Dataset?

- Widely used in academic benchmarking for CVRP.
- Ideal for testing classical and hybrid approaches.
- Enables direct performance comparison with existing solutions.

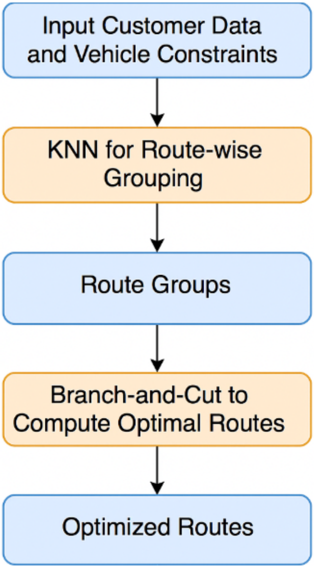
Objectives of the Thesis

- To understand the theoretical and practical aspects of the Capacitated Vehicle Routing Problem (CVRP).
- To explore and analyze traditional and modern solution techniques for CVRP.
- To design and implement a hybrid algorithm combining:
  - *K-Nearest Neighbour (KNN)* for intelligent customer grouping.
  - *Branch-and-Cut* for solving optimized delivery routes.
- To test and evaluate the hybrid model using benchmark dataset A-n33-k5.
- To compare the performance of the proposed approach with classical methods.
- To highlight the strengths, limitations, and real-world applicability of the proposed solution.

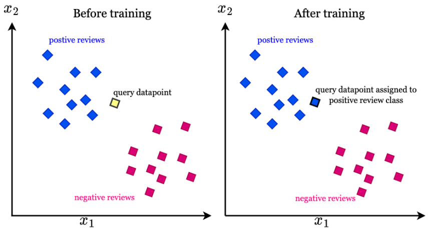
Why a Hybrid Approach?

- Classical methods often struggle with scalability and solution quality for large or complex instances.
- Heuristics are fast but can lead to sub-optimal solutions without global perspective.
- Exact algorithms are accurate but computationally expensive for larger datasets.
- Machine learning offers data-driven insights, but lacks optimization precision.
- **Hybridization combines the strengths of both worlds:**
  - KNN: Leverages spatial proximity for intelligent clustering of customer nodes.
  - Branch-and-Cut: Ensures optimal or near-optimal routing under constraints.
- Aims to improve both solution speed and accuracy.

Overview Diagram of the Proposed Hybrid Model



- K-Nearest Neighbors (KNN) is a simple, instance-based learning algorithm.
- It classifies a data point based on the majority class among its K closest neighbors.
- In clustering context, KNN helps group nodes based on spatial proximity.
- Useful for initial grouping of nodes in Capacitated Vehicle Routing Problem (CVRP).



Why KNN for Initial Clustering?

- Captures local neighborhood structure effectively.
- Reduces search space for route optimization by grouping nearby nodes.
- Simple to implement and computationally efficient for initial clustering.
- Helps improve solution quality by ensuring spatially coherent clusters.
- Works well with mixed clustering methods (e.g., combined with K-means).

Implementation Details of KNN in CVRP

- Compute Euclidean distance between all nodes.

Given a set of nodes (customers and depot) with coordinates (x_i, y_i), the Euclidean distance d_{ij} between nodes i and j is:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Matrix Representation:
For N nodes, the distance matrix D is symmetric ($d_{ij} = d_{ji}$) with zeros on the diagonal ($d_{ii} = 0$):

$$D = \begin{bmatrix} 0 & d_{12} & \cdots & d_{1N} \\ d_{21} & 0 & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & 0 \end{bmatrix}$$

- For each node, find its K nearest neighbors.

For each node i, sort all other nodes by d_{ij} in ascending order and select the top K nodes.

Let $\mathcal{N}_i^{(K)}$ be the set of K-nearest neighbors of node i:

$$\mathcal{N}_i^{(K)} = \{j_1, j_2, \dots, j_K\}, \quad \text{where} \quad d_{ij_1} \leq d_{ij_2} \leq \cdots \leq d_{ij_K}$$

- Assign nodes to clusters based on neighbor relationships.
  - Start with the node farthest from the depot.
  - Assign it and its K-nearest neighbors to a cluster if capacity allows:

$$\mathcal{C}_m = \{i\} \cup \left\{ j \in \mathcal{N}_i^{(K)} \mid \sum_{k \in \mathcal{C}_m} \text{demand}(k) \leq Q \right\}$$

where Q is the vehicle capacity.

- Ensure clusters respect vehicle capacity constraints.

Overview of Branch-and-Cut

- **Branch-and-Cut** is a mathematical optimization technique used to solve integer programming problems like CVRP.
- It combines two powerful strategies:
  - **Branch-and-Bound:** Systematically explores subsets of possible solutions.
  - **Cutting Planes:** Adds linear constraints (cuts) to eliminate infeasible regions without excluding feasible solutions.
- Widely used in combinatorial optimization, especially when the problem involves binary or integer decision variables.
- Efficient in solving large-scale problems with complex constraints.
- Solves CVRP by optimizing route decisions while respecting vehicle capacity constraints and minimizing total travel cost.
- Implemented using solvers like CPLEX, Gurobi, or CBC.

How Branch-and-Cut Solves CVRP

- **Step 1:** Formulate the CVRP as an Integer Linear Programming (ILP) problem with objective and constraints.
- **Step 2:** Relax the integer constraints to solve the Linear Programming (LP) relaxation.
- **Step 3:** Identify and add violated constraints (cutting planes), such as subtour elimination constraints.
- **Step 4:** If the LP solution is not integral, perform *branching* by creating subproblems with fixed variable values.
- **Step 5:** Repeat cutting and branching recursively until:
  - An optimal integer solution is found, or
  - Time or memory limit is reached.
- **Benefit:** Efficiently narrows the feasible space and avoids infeasible or fractional solutions.

Brach-and-Cut Algorithm Flowchart

ILP Formulation of CVRP

Variables:

- $x_{ij} = \begin{cases} 1 & \text{if edge } (i,j) \text{ is used} \\ 0 & \text{otherwise} \end{cases}$
- q_i = demand at customer i , Q = vehicle capacity

Objective

Minimize total travel cost while serving all customers:

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij}$$

ILP Formulation (Constraints)

1. **Depart/Return**: exactly K vehicles depart the depot and return to the depot.

$$\sum_{j=1}^{N} x_{0j} = K, \quad \sum_{i=1}^{N} x_{i0} = K$$

2. **Flow Conservation**: Each customer has visited exactly once.

$$\sum_{i=0}^{N} x_{ij} = 1 \quad \forall j \in V', \quad \sum_{j=0}^{N} x_{ij} = 1 \quad \forall i \in V'$$

Capacity Constraints (MTZ)

$$u_i - u_j + Qx_{ij} \leq Q - q_j \quad \forall i, j \in V', \; i \neq j$$

$$q_i \leq u_i \leq Q \quad \forall i \in V'$$

where q_i is demand at node i and Q is vehicle capacity.

Integration of KNN with Branch-and-Cut

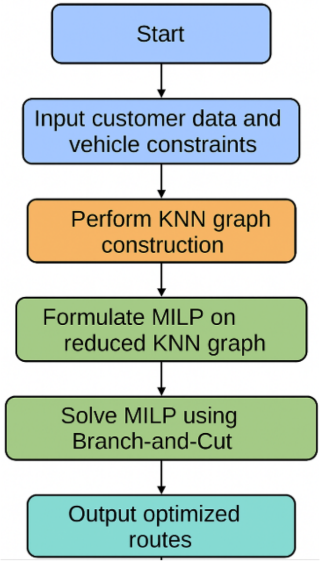
Objective: Reduce problem complexity by focusing only on relevant local connections.

Steps:

1. **K-Nearest Neighbors (KNN) Graph Construction:**
   - For each customer node, identify the k nearest neighbors based on Euclidean distance.
   - Construct a reduced graph where each node is connected only to its k nearest neighbors.
2. **MILP Formulation on Reduced Graph:**
   - Define the CVRP problem using only the edges from the KNN graph.
   - Reduces the number of binary variables x_{ij}, resulting in a smaller search space.
3. **Apply Branch-and-Cut:**
   - Solve the MILP using the Branch-and-Cut algorithm.
   - Cuts are added to remove infeasible solutions and improve convergence.

Result: Faster and more efficient solution without significant loss in optimality.

Hybrid Algorithm Flowchart



Hybrid Algorithm (Part 1/2)

Algorithm 1 KNN Clustering & MILP Setup

Phase 1: KNN Clustering

```
Compute $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \ \forall i, j \in V \cup \{0\}$
for each customer $i \in V$ do
    $\mathcal{N}_i \leftarrow K$ nearest neighbors of $i$ (by $d_{ij}$)
    Create candidate edges $(i, j)\ \forall j \in \mathcal{N}_i$
end for
```

Phase 2: MILP Formulation

- Initialize binary variables $x_{ij} \in \{0, 1\}$
- **Objective:** $\min \sum_{i,j} d_{ij} x_{ij}$
- **Constraints:**
  - $\sum_{j \neq i} x_{ij} = 1, \ \forall i \in V$ (Visit each customer once)
  - $\sum_{i \in S} q_i \leq Q, \ \forall S \subseteq V$ (Route capacity)
  - $\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1$ (Subtour elimination)

Hybrid Algorithm (Part 2/2)

Algorithm 2 Branch-and-Cut Procedure

Phase 3: Branch-and-Cut

Solve LP relaxation to get $\bar{x}_{ij}$
while $\exists \bar{x}_{ij} \notin \{0, 1\}$ **do**
    if $\exists$ violated subtour constraint for subset S **then**
        Add cut: $\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1$
    end if
    if no cuts added but solution still fractional **then**
        Select most fractional $\bar{x}_{pq}$ (closest to 0.5)
        Branch:
- ▷ Left child: Set $x_{pq} = 0$
- ▷ Right child: Set $x_{pq} = 1$

    end if
    Re-solve LP with new constraints
end while

Output

return Optimized vehicle routes $\{R_1, ..., R_k\}$

Software & Tools Used

- **Python:** For data preprocessing, distance matrix computation, and clustering.
- **Scikit-learn:** Used for K-Means clustering and K-Nearest Neighbors.
- **Matplotlib & Seaborn:** For data visualization and route plotting.
- **Jupyter Notebook:** For developing and documenting experiments interactively.
- **LaTeX Beamer:** For thesis presentation preparation.

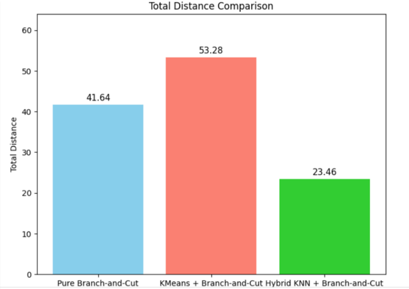
Limitations or Assumptions

- **Homogeneous Fleet:** All vehicles have the same capacity.
- **Static Demand:** Customer demand is known and does not change.
- **Fixed Number of Vehicles:** Predefined maximum number of vehicles used.
- **Euclidean Distances:** Assumes straight-line distances between nodes.
- **Deterministic Inputs:** No uncertainty considered in demands or travel times.
- **Single Depot:** All vehicles start and end at a single depot.

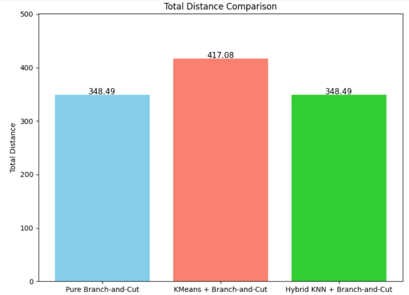
Summary of Methodology

1. **Problem Understanding:** Focused on solving Capacitated Vehicle Routing Problem (CVRP).
2. **Data Collection:** Used benchmark dataset A-n33-k5 from CVRPLIB.
3. **Preprocessing:** Computed distance matrix and demand profile.
4. **Graph Pruning:** Applied K-Nearest Neighbors to reduce edge set.
5. **MILP Formulation:** Defined binary decision variables and CVRP constraints.
6. **Optimization:** Solved the problem using Branch-and-Cut on reduced graph.
7. **Evaluation:** Compared results in terms of route quality and computation time.

Graphical Comparison of the Result



Method	Total Distance
1) Only Brach-and-cut	41.64
2) K-means +Branch-and-Cut	53.28
3) Hybrid KNN + Branch-and-Cut	23.46



Method	Total Distance
1) Only Branch-and-Cut	348.49
2) KMeans + Branch-and-Cut	417.08
3) Hybrid KNN + Branch-and-Cut	348.49

Route Images

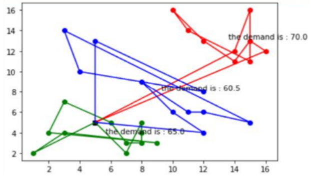


Figure: Vehicles Routes Generated by K-Means Algorithm for the instance of 33 customers

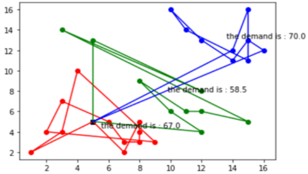


Figure: Vehicles Routes Generated by Branch and cut Algorithm for the instance of 33 customers

Discussion on Results

- **Hybrid approach** (KNN + Branch-and-Cut) significantly improves **computational efficiency**.
- Compared to **pure Branch-and-Cut**, the hybrid method reduces edge complexity while maintaining near-optimal solutions.
- **KMeans + Branch-and-Cut** offers faster solving for clusters but lacks global optimization awareness.
- KNN-based pruning ensures only relevant edges are used, leading to **faster MILP solving** and **lower memory usage**.
- Achieves a good balance between runtime and solution quality on datasets like **A-n33-k5**.
- Results show that the **hybrid method is scalable and practical** for larger VRP instances.

Key Findings

- The proposed **hybrid algorithm (KNN + Branch-and-Cut)** improves solution efficiency without sacrificing accuracy.
- **KNN** reduces graph complexity by focusing on relevant neighboring connections.
- **Branch-and-Cut** ensures optimal sub-tour elimination and constraint satisfaction on the pruned graph.
- Demonstrated **scalability and flexibility** for mid-size to large CVRP datasets.
- Hybrid model is **modular**, allowing easy integration of other clustering or pruning strategies.

Limitations of Current Work

- **Clustering and KNN pruning** may discard globally optimal edges due to local neighborhood focus.
- The model assumes **deterministic demands** and **static travel times**.
- Only tested on a limited dataset (**A-n33-k5**); performance on more complex networks not explored.
- **No dynamic re-routing** or real-time adjustments considered.
- Current MILP formulations have **limited scalability** for extremely large datasets (e.g., greater than 500 nodes).
- Further tuning of **K** in KNN and number of clusters in KMeans could enhance performance.

Scope of Future Work

- Extend hybrid approach by integrating other ML models (e.g., reinforcement learning).
- Incorporate dynamic and real-time data for adaptive routing solutions.
- Explore multi-objective optimization including cost, time, and environmental impact.
- Apply the approach to larger, more complex datasets and real-world scenarios.
- Develop user-friendly software tools or platforms for practical deployment.
- Investigate hybrid algorithms combining other heuristics and exact methods.

References I

L. Sakhri, K. Rahmani, and A. Benyettou, "Capacitated vehicle routing problem based on a hybrid clustering algorithm using a novel k-means and knn," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 729–746, 2022.

D. Konstantakopoulos, L. Gkatzikis, and I. Koutsopoulos, "A hybrid method for solving the capacitated vehicle routing problem," *Computers & Operations Research*, vol. 113, p. 104801, 2020.

M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53–66, 1997.

A. Hassanzadeh, E. B. Tirkolaee, and I. Mahdavi, "A novel hybrid algorithm based on ant colony optimization and simulated annealing for the capacitated vehicle routing problem," *Engineering Applications of Artificial Intelligence*, vol. 108, p. 104564, 2022.

L. Zhao, X. Li, and T. Wang, "A hybrid genetic algorithm for the capacitated vehicle routing problem with time windows," *Applied Soft Computing*, vol. 113, p. 107904, 2022.

Y. Liu, Y. Zhang, and L. Yang, "Hybrid particle swarm optimization algorithm for capacitated vehicle routing problem," *Expert Systems with Applications*, vol. 187, p. 115922, 2022.

N. Sharma and R. Gupta, "An efficient hybrid approach for solving capacitated vehicle routing problem using genetic algorithm and local search," *Journal of Computational Science*, vol. 68, p. 101890, 2023.

Thank You!