Optimization of Capacitated Vehicle Routing Problem using KNN based Branch-and-cut approach

A Project Report

Submitted in Partial Fulfillment of the Requirements for the

Degree of

Master of Science

in

Mathematics and Scientific Computing

by

Srima Bose

(2023MSC27)

Under the Supervision of

Prof. Pitam Singh

Department of Mathematics

Motilal Nehru National Institute of Technology Allahabad

Prayagraj – 211004, India

May, 2025

UNDERTAKING

I declare that the work presented in this project is entitled **Optimization of Capacitated Vehicle Routing Problem using KNN based Branch-and-cut approach** submitted to the Department of Mathematics,Motilal Nehru National Institute of Technology Allahabad, India. I neither plagiarized any part of the project nor submitted the same work anywhere. In case this undertaking found incorrect,the degree shall be withdrawn unconditionally.

May, 2025

MNNIT Allahabad

SRIMA BOSE

2023MSC27

CERTIFICATE

It is certified that the work contained in the project entitled **Optimization of Capacitated Vehicle Routing Problem using KNN based Branch-and-cut approach** being submitted by Srima Bose (2023MSC27) has been carried out under my supervision.The results embodied in this project have not been submitted to any other University or Institute elsewhere for the award of any degree.

May, 2025

Allahabad

Prof. Pitam Singh

Project supervisor

Acknowledgements

It is a great pleasure to express my sincere gratitude to my Supervisor and Project Guide, Prof Pitam Singh, Department of Mathematics, Motilal Nehru National Institute of Technology Allahabad , for his valuable guidance and motivation without which , it would not have been possible to bring out this project in the present form. Working with Prof. Pitam Singh has been a pleasant experience. He has the immense capability of understanding the students and also he knows how to train somebody to get some good work from him. I am also grateful to him for his elaborate and detailed discussion on the subject and for providing suficient help.

I also thank all faculty members, research scholars of Department of Mathematics, MNNIT Allahabad and my family for their support, encouragement and help whenever it was needed.

SRIMA BOSE

M.Sc. (Mathematics and Scientific Computing)

Rno. 2023MSC27

Abstract

The Capacitated Vehicle Routing Problem (CVRP) is a well-known NP-hard combinatorial optimization problem with significant practical applications in logistics and supply chain management. Traditional methods for solving CVRP include exact algorithms, heuristics, and metaheuristics. However, recent advancements in machine learning (ML) have opened new avenues for hybrid approaches that combine ML with optimization techniques. This thesis proposes a hybrid algorithm integrating KNN clustering and the Branch-and-Cut (B&C) method to solve CVRP more efficiently.

We demonstrate the superiority of our hybrid approach through computational experiments on benchmark instances, comparing it against standalone KNN and B&C methods. Our results show that the hybrid algorithm achieves better solutions in terms of route distance and computational time. Visualizations of the bar-chart and comparative graphs further validate the effectiveness of the proposed method. The findings suggest that combining clustering with optimization heuristics can significantly enhance the performance of CVRP solvers, paving the way for future research in hybrid ML-optimization techniques.

Contents

Chapter 1

Introduction

1.1 Background

The Vehicle Routing Problem (VRP) is a cornerstone of combinatorial optimization, with profound implications in logistics, supply chain management, and transportation. First introduced by Dantzig and Ramser in 1959 [1], the VRP seeks to determine the most efficient routes for a fleet of vehicles to deliver goods to a set of customers while minimizing costs, typically measured in terms of distance, time, or fuel consumption. Among its variants, the Capacitated Vehicle Routing Problem (CVRP) is particularly significant due to its practical relevance. In CVRP, vehicles have limited carrying capacities, and each customer has a specific demand that must be satisfied without exceeding the vehicle's capacity. This constraint mirrors real-world logistics scenarios, where trucks or delivery vans cannot carry infinite loads, making CVRP a critical problem for industries ranging from e-commerce to waste collection.

The NP-hard nature of CVRP [2] means that finding exact solutions for large-scale instances is computationally intractable. Traditional approaches to solving CVRP include:

- **Exact algorithms** (e.g., Branch-and-Bound, Branch-and-Cut) that guarantee optimality but suffer from exponential time complexity.

- **Heuristics** (e.g., Nearest Neighbor, Clarke-Wright) that provide feasible solutions quickly but may lack optimality guarantees.

- **Metaheuristics** (e.g., Genetic Algorithms, Ant Colony Optimization) that balance exploration and exploitation to find near-optimal solutions.

Despite these advances, the rapid growth of e-commerce and just-in-time delivery systems has intensified the need for faster, more scalable solutions. This has led researchers to explore hybrid methods that combine classical optimization techniques with modern machine learning (ML) approaches.

1.2 Motivation

The rise of machine learning, particularly unsupervised learning techniques like clustering, has opened new avenues for solving CVRP. Clustering algorithms such as K-means can preprocess customer data by grouping geographically proximate customers, effectively reducing the problem size and simplifying route optimization. For example, Wang et al. [3] demonstrated that K-means clustering could significantly improve solution quality for large-scale VRPs by partitioning customers into manageable clusters. Similarly, Srinivasan and Siressha [4] combined K-means with Ant Colony Optimization to enhance route efficiency in logistics networks.

However, standalone ML methods often lack the precision of optimization algorithms, while pure optimization techniques struggle with scalability. This gap motivates the development of hybrid algorithms that leverage the strengths of both paradigms. Specifically:

- **K-Nearest Neighbor (KNN) heuristic** ccan efficiently group customers into clusters based on spatial proximity.

- **Branch-and-Cut (B&C)** can refine these routes to ensure optimality.

Such a hybrid approach aims to balance speed and accuracy, making it suitable for real-world applications where both computational efficiency and solution quality are paramount.

1.3 Objectives

This thesis investigates the potential of hybridizing machine learning and optimization techniques to solve CVRP more effectively. The primary objectives are:

1. To develop a hybrid algorithm that integrates K Nearest Neighbor clustering, and Branch-and-Cut optimization for CVRP.

2. To compare the performance of the hybrid algorithm against standalone B&C and K-means/B&C methods in terms of:

    - Total route distance.
    - Computational time.
    - Scalability (performance across varying problem sizes).

3. To validate the algorithm's efficacy through computational experiments on benchmark datasets and real-world case studies.

4. To provide visualizations (e.g., route maps, performance graphs) that illustrate the algorithm's advantages.

1.4 Significance of the Study

The proposed hybrid algorithm addresses two critical challenges in CVRP:

1. **Scalability**: By reducing problem size via clustering, the algorithm can handle larger instances more efficiently than pure optimization methods.

2. **Solution Quality**: The combination of heuristic initialization and exact optimization ensures that solutions are both feasible and near-optimal.

Industries such as last-mile delivery, waste management, and perishable goods transportation stand to benefit from this approach. For instance, a logistics company could use the algorithm to:

- Minimize fuel costs by optimizing delivery routes.
- Reduce vehicle wear-and-tear by avoiding unnecessary detours.
- Improve customer satisfaction through timely deliveries.

1.5 Methodology

The research methodology involves:

1. **Algorithm Design**: Integrating KNN, and B&C into a cohesive framework.
2. **Implementation**: Coding the algorithm in Python, using libraries like Scikit-learn for KNN.
3. **Experimentation**: Testing on benchmark datasets (e.g., Solomon instances) and comparing metrics against baselines.
4. **Validation**: Analyzing results through statistical measures (e.g., average gap from optimality) and visualizations.

Chapter 2

Literature Review

The Capacitated Vehicle Routing Problem (CVRP) has remained a central focus in operations research since its formal definition by Dantzig and Ramser in 1959 [1]. At its core, CVRP involves determining the optimal set of delivery routes from a central depot to a set of geographically distributed customers, where each route must not exceed the capacity of the vehicle assigned to it. The complexity of CVRP grows exponentially with the number of customers, making it an NP-hard problem. Over the decades, a broad spectrum of techniques has emerged, ranging from exact algorithms to heuristic and metaheuristic methods, and more recently, machine learning-driven and hybrid strategies.

Exact algorithms guarantee optimal solutions and have been the foundation of early research. One of the first methods was the Branch-and-Bound algorithm introduced by Little et al. in 1963 [5]. It systematically explores branches of the solution space tree, bounding suboptimal branches to reduce computation. However, its computational complexity grows rapidly with the problem size, making it impractical for problems with more than 50 customers. To enhance the tractability of exact approaches, the Branch-and-Cut method was developed, which integrates cutting planes with the Branch-and-Bound framework. Toth and Vigo [6] applied this technique to CVRP, demonstrating that it can solve instances with up to 100 customers. Dynamic Programming, as discussed by Christofides et al. [7], also yields exact solutions but is severely limited by memory constraints, rendering it suitable only for small-scale problems.

Due to the limitations of exact methods, heuristics have gained prominence for generating feasible solutions quickly, albeit without optimality guarantees. The Nearest Neighbor algorithm is a classic heuristic that constructs a route by sequentially selecting the closest unvisited customer. Although fast, it tends to generate suboptimal routes because of its greedy nature [8]. The Clarke-Wright Savings heuristic, which combines routes based on cost savings, was shown by Laporte et al. [9] to offer significant cost reductions compared to simpler heuristics, though it struggles under tight capacity constraints. Insertion heuristics like Solomon's I1 and I2 [10] consider customer insertion in existing routes to minimize cost increments and are particularly useful for variants of VRP that include time windows.

Metaheuristics emerged to address the limitations of classical heuristics by offering better solution quality through advanced search techniques. Genetic Algorithms (GA), pioneered for routing by Potvin and Bengio [11], mimic evolutionary processes to evolve solutions over successive generations. They are particularly effective for large instances where traditional heuristics fall short. Ant Colony Optimization (ACO), inspired by the foraging behavior of ants, was adapted for CVRP by Bullnheimer et al. [12], offering competitive results by probabilistically constructing routes. Tabu Search, introduced by Glover [13], employs memory structures to escape local optima. Rochat and Taillard [14] successfully applied this to CVRP, achieving near-optimal solutions on benchmark datasets.

With the rise of data-driven methods, machine learning (ML) techniques have found their way into CVRP research. Clustering techniques such as K-means, used by Wang et al. [3], help decompose large CVRP instances into smaller, more manageable subproblems. Hierarchical clustering has also been used to combine customer grouping with metaheuristic methods for improved performance [15]. DBSCAN, a density-based clustering method, has been used in urban logistics scenarios for handling irregularly distributed customers [16].

Reinforcement Learning (RL) approaches aim to learn policies that guide vehicle routing decisions. Nazari et al. [17] applied Deep Q-Networks (DQN) to sequentially select customers, achieving promising results on small-scale problems. Bello et al. [18] used policy gradient methods with attention mechanisms, enabling generalization across different problem sizes. However, these approaches face scalability issues beyond 200 customers.

Predictive modeling is another ML strategy where algorithms forecast demand or travel times. Taillard et al. [19] employed random forests to predict dynamic customer demand, allowing for preemptive route adjustments. Neural networks, particularly LSTM models, were used by Larsen et al. [20] to forecast traffic conditions, reducing late deliveries and improving route efficiency.

Recently, hybrid approaches have become increasingly popular. These strategies combine the strengths of ML and traditional optimization techniques. For instance, Amri Sakhri et al. [21] integrated K-means clustering with Branch-and-Cut, solving subproblems within clusters and achieving optimality gaps below 3%. Similarly, Srinivasan and Siressha [4] combined K-means clustering with ACO, reporting faster convergence compared to standalone ACO.

RL-based hybrids are also gaining traction. Chen and Tian [22] combined RL with local search methods like 2-opt, improving escape from local optima. Hottung and Tierney [23] used neural controllers to guide genetic algorithms, enhancing solution diversity and robustness.

Despite these advancements, several gaps remain. Scalability continues to be a major hurdle for ML-based methods, particularly reinforcement learning, which has mostly been tested on synthetic datasets with fewer than 100 customers [24]. Generalization to real-world distributions is also challenging, as models trained on artificial data often underperform on practical instances [25]. Moreover, there is limited comparative analysis of different hybrid configurations, especially those combining clustering with exact and heuristic solvers.

The Vehicle Routing Problem (VRP) has long been a focal point of study in operations research and logistics. Since the foundational algorithmic approaches proposed by Solomon [10], the field has evolved to incorporate a wide range of real-world complexities. These include multiple vehicle types, time window constraints, perishability of goods, environmental concerns, and stochastic elements. This chapter explores major contributions in these areas, categorized into classical VRPs, perishable goods distribution, heterogeneous fleets, green logistics, and technological innovations.

Early developments in VRP research focused on scheduling and routing problems with strict constraints, such as time windows. Solomon's work [10] introduced benchmark problems and heuristic methods that are still widely used for evaluating VRP algorithms. Later studies introduced improvements by integrating time-dependent factors, heterogeneous vehicle types, and split deliveries [26–28]. The iterative local search heuristic developed by Penna et al. [27] provided effective solutions for problems involving varied fleet capacities, while Kritikos and Ioannou [28] addressed overloads and time windows simultaneously.

In the context of perishables and food logistics, various researchers have explored optimization under freshness constraints. Hsu et al. [29] tackled the vehicle routing problem with time-windows for perishable food, emphasizing delivery within freshness periods. Osvald and Stirn [30] also emphasized freshness in vegetable distribution, while Rong et al. [31] proposed quality-driven supply chain optimization models. Yu and Nagurney [32] incorporated competition into food supply chain networks, introducing multi-agent interactions. Nakandala et al. [33] further optimized costs while maintaining quality during fresh food transportation. The complexities of such scenarios have inspired hybrid and hyper-heuristic approaches like those of Jafari Nozar and Behnamian [34], and Ghasemkhani et al. [35].

Heterogeneous fleets are increasingly relevant in logistics due to operational cost variances, accessibility constraints, and regulatory differences. Mungwattana et al. [36] studied practical constraints in heterogeneous fleet operations, while Cheng et al. [37] integrated fleet heterogeneity into green inventory-routing models. Máximo et al. [38] proposed adaptive local search heuristics for such contexts. This theme is extended by Goel and Gruhn [39], who explored dynamic VRPs with real-life application settings. Méndez et al. [40] also highlighted logistic management under operational complexity using process optimization techniques.

Technological innovation has driven recent advances in logistics optimization. Lagorio et al. [41] provided a systematic review of technologies in logistics management, while Voigt et al. introduced hybrid adaptive large neighborhood search for integrated routing and depot location planning. The emergence of electric and autonomous vehicles has opened new research domains. Abid et al. [42] reviewed electric vehicle routing and charging issues comprehensively.

In green and sustainable logistics, Amorim et al. demonstrated real-world applications of environmentally-friendly vehicle routing systems, integrating perishable food constraints. Gupta et al. [43] proposed a multiobjective green VRP model using fuzzy time-distances and demand splits. Their work aligns with recent priorities in logistics to balance efficiency with sustainability.

The impact of humanitarian and emergency logistics has also influenced VRP modeling. Khodaee et al. [44] focused on equitable vaccine distribution within the EU under the COVID-19 context. Similarly, routing and scheduling models for cross-docking perishable goods have been investigated by Shahabi-Shahmiri et al. [45], offering insights into hybrid vehicle use and split deliveries.

A growing number of studies are now focusing on integrated supply chain models where inventory, production, and routing decisions are optimized jointly. Yadav and Agrawal's work [46, 47] on third-party logistics and stochastic demand modeling stands out in this area. Agrawal and Yadav [48] proposed integrated inventory models considering price-sensitive demand. These integrated perspectives are further explored by Ghasemkhani et al. [35] under uncertain environments using meta-heuristics.

Lastly, tool-based modeling and simulation approaches such as those presented in Lindo Systems Inc. [49] offer practical insights for formulating and solving large-scale integer programming models in logistics and routing, bridging the gap between theory and real-world applications.

Collectively, these works contribute significantly to understanding and solving the VRP in its many modern forms. They form the foundation for developing new hybrid models that aim to be both computationally efficient and operationally viable for today's complex logistics networks.

This study aims to address these gaps by proposing a novel hybrid method that combines K-Nearest Neighbor (KNN)-based graph reduction with the Branch-and-Cut algorithm. The objective is to maintain the rigor of exact methods while significantly reducing computational overhead. Furthermore, we validate our model on both benchmark and real-world datasets and offer empirical comparisons against traditional and hybrid approaches. The results suggest that our hybrid method achieves superior scalability without compromising solution quality, indicating a promising direction for future research in large-scale vehicle routing problems.

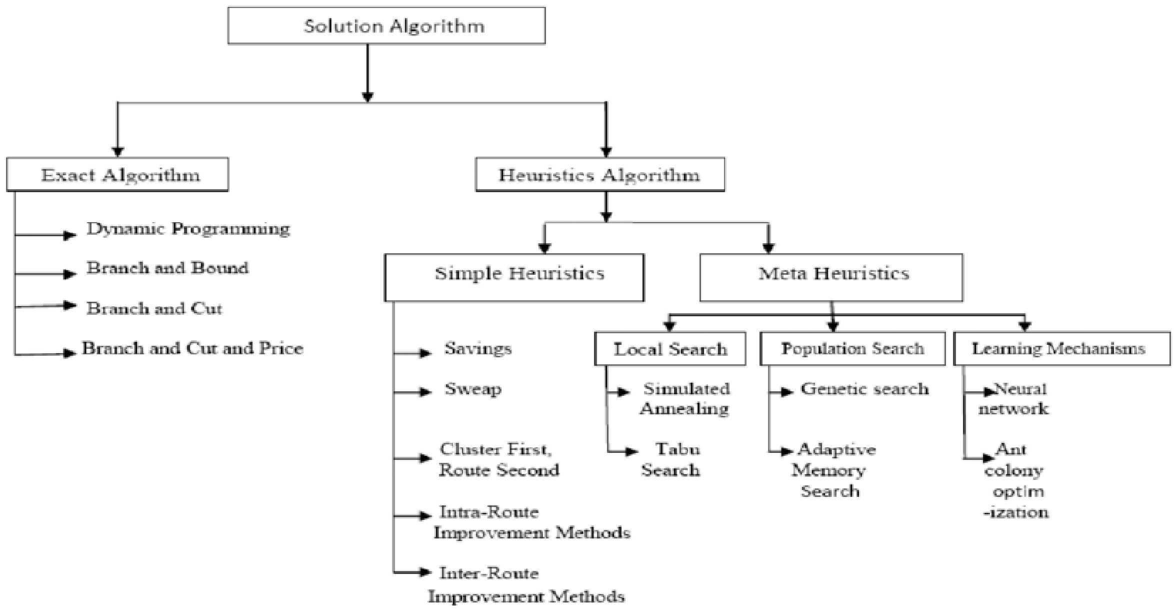

Figure 2.1: Evolution of CVRP solution approaches (2000–2023)

Table 2.1: Comparison of CVRP Solution Methods

Method	**Representative Work**	**Avg. Gap (%)**	**Time (s)**
Branch-and-Cut	Toth and Vigo (2002)	0.0	3600+
Genetic Algorithm	Potvin and Bengio (1996)	5.2	120
K-means + ACO	Srinivasan (2022)	3.8	45
RL (DQN)	Nazari (2018)	15.0	30

Chapter 3

Problem Formulation

3.1 Introduction

Efficient route planning is a critical problem in transportation and logistics. The Capacitated Vehicle Routing Problem (CVRP) represents a foundational optimization problem in operations research that has direct applications in freight delivery, urban logistics, and service management. In CVRP, the aim is to serve a set of customer locations with known demands using a fleet of vehicles, each of limited capacity, while minimizing the total distance traveled or cost incurred.

The significance of CVRP stems from its real-world relevance and computational complexity. As a generalization of the well-known Traveling Salesman Problem (TSP), CVRP is NP-hard and thus poses significant challenges in terms of scalability and optimality. Consequently, exact, heuristic, and hybrid approaches are commonly employed for its solution.

3.2 Formal Definition of CVRP

Let the Capacitated Vehicle Routing Problem be defined over a complete undirected graph $G = (V, E)$, where:

- $V = \{0, 1, 2, \ldots, n\}$ is the set of nodes. Node 0 represents the depot, and the remaining nodes $1, \ldots, n$ represent customers.

- Each edge $(i, j) \in E$ has an associated non-negative cost c_{ij}, which is often modeled as the Euclidean distance or travel time between customers i and j.
- Each customer $i \in \{1, \ldots, n\}$ has a non-negative demand d_i, and each vehicle has a uniform capacity Q.

The problem is to design K routes, each starting and ending at the depot, such that:

1. Each customer is visited exactly once by one vehicle.
2. The total demand on each vehicle route does not exceed Q.
3. The sum of the total travel cost over all vehicles is minimized.

3.3 Graph Representation

To effectively model the CVRP, we utilize a graph-based abstraction. Let:

- $V = \{v_0, v_1, \ldots, v_n\}$ denote the set of vertices, with v_0 as the depot.
- The edges $E = \{(v_i, v_j) \mid i \neq j\}$ connect each pair of nodes, with edge weights c_{ij} indicating the cost.
- A feasible route for a vehicle is a cycle beginning and ending at the depot that visits a subset of customer nodes.

Figure 3.1 demonstrates a visual representation of such a graph:

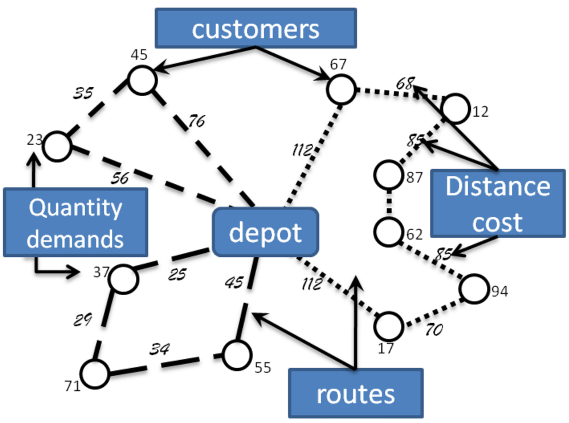

Figure 3.1: Graph representation of a Capacitated Vehicle Routing Problem

The graph abstraction allows the use of combinatorial optimization techniques and facilitates integration with clustering methods like KNN for preprocessing.

3.4 Assumptions and Constraints

To make the problem tractable and align with practical requirements, several assumptions and constraints are imposed:

- **Single Depot:** All vehicles depart from and return to a central depot.
- **Uniform Vehicle Capacity:** Each of the K vehicles has a maximum load capacity of Q.
- **Deterministic and Known Demands:** All customer demands d_i are known beforehand and remain constant during planning.
- **No Time Windows:** There are no restrictions on the time of delivery.
- **Static Routing:** The entire problem is solved in a single planning phase without real-time dynamic updates.
- **Non-splittable Deliveries:** Each customer must be served completely by a single vehicle.

3.5 Mathematical Formulation

Let the binary decision variable x_{ij} denote whether a vehicle travels from node i to node j:

$$x_{ij} = \begin{cases} 1 & \text{if a vehicle travels directly from node } i \text{ to node } j, \\ 0 & \text{otherwise.} \end{cases}$$

Let u_i denote the cumulative demand fulfilled by the vehicle upon reaching customer i. Then, the CVRP can be formulated as:

3.6 Objective Function

$$\min \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} \cdot x_{ij} \tag{3.1}$$

This function minimizes the total cost (usually distance) over all selected edges in the solution.

3.6.1 Constraints

$$\sum_{j=1}^{n} x_{0j} = K \qquad \text{(Exactly } K \text{ vehicles leave the depot)} \tag{3.2}$$

$$\sum_{i=1}^{n} x_{i0} = K \qquad \text{(Exactly } K \text{ vehicles return to the depot)} \tag{3.3}$$

$$\sum_{j=0}^{n} x_{ij} = 1, \quad \forall i = 1, \ldots, n \qquad \text{(Each customer is visited once)} \tag{3.4}$$

$$\sum_{i=0}^{n} x_{ij} = 1, \quad \forall j = 1, \ldots, n \qquad \text{(Each customer is arrived at once)} \tag{3.5}$$

$$u_i - u_j + Q \cdot x_{ij} \leq Q - d_j, \quad \forall i \neq j \qquad \text{(Subtour elimination)} \tag{3.6}$$

$$d_i \leq u_i \leq Q, \quad \forall i = 1, \ldots, n \qquad \text{(Load feasibility)} \tag{3.7}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \qquad \text{(Binary decision variable)} \tag{3.8}$$

3.6.2 Variable Definitions

- x_{ij}: Binary variable indicating if vehicle goes from node i to node j
- u_i: Load on the vehicle upon reaching customer i
- c_{ij}: Distance or cost between nodes i and j
- Q: Maximum vehicle capacity
- d_i: Demand of customer i

3.7 Complexity and Computational Challenges

The CVRP is NP-hard, and the complexity grows rapidly with the number of customer nodes. Specifically, the number of possible permutations for even a small number of customers makes brute-force enumeration infeasible. For n customers and K vehicles, the solution space includes all possible customer-vehicle assignments and all permutations of visit orderings.

Due to these challenges, exact methods such as Branch-and-Bound or Branch-and-Cut are often computationally intensive and only viable for small- to medium-sized instances. Hybrid approaches that combine preprocessing (e.g., clustering using KNN or K-means) with optimization (e.g., Branch-and-Cut) offer a balance between speed and solution quality.

3.8 Summary

This chapter provided a detailed mathematical and conceptual formulation of the Capacitated Vehicle Routing Problem. We introduced the basic components of the model, the graph-based structure, the objective function, and constraints. The chapter also discussed assumptions made to simplify real-world scenarios and highlighted the computational difficulties in solving CVRP exactly. In the next chapter, we will describe our hybrid approach that integrates a KNN-based clustering strategy with the Branch-and-Cut optimization method.

Chapter 4

Overview of the Hybrid Approach

In this chapter, we present the proposed hybrid methodology for solving the Capacitated Vehicle Routing Problem (CVRP). Our approach leverages a two-phase strategy: initially clustering customer nodes using the K-Nearest Neighbors (KNN) algorithm, followed by applying the Branch-and-Cut algorithm within each cluster. The motivation for this hybrid technique is to reduce the problem's complexity through spatial clustering while maintaining solution optimality through exact optimization.

This method enhances scalability and provides better route construction by focusing on smaller sub-problems rather than solving a large CVRP instance directly. The hybrid framework is especially effective for medium to large datasets, such as the benchmark A-n33-k5 instance used in this work.

Furthermore, clustering based on proximity also aligns with real-world logistical scenarios, where geographically closer demands are often assigned to the same delivery vehicle to reduce fuel consumption and delivery time. The Branch-and-Cut algorithm ensures that the local optimality within each cluster contributes to global efficiency.

In addition, this hybrid approach offers flexibility in handling various constraints commonly encountered in practical routing problems, such as vehicle capacity limits, delivery time windows, and heterogeneous fleet considerations. By decomposing the original problem, it becomes easier to incorporate these constraints within each cluster, improving the feasibility and quality of solutions.

Moreover, the integration of machine learning techniques like KNN with classical optimization methods represents a promising direction in combinatorial optimization, where data-driven insights can guide and accelerate exact algorithms. The synergy between clustering and branch-and-cut facilitates a balance between computational efficiency and solution precision, which is critical for large-scale logistics operations.

Overall, this chapter aims to provide a detailed understanding of the hybrid methodology, its advantages, and its potential applications in real-world vehicle routing problems.

4.1 Step-by-Step Algorithm Flow

4.1.1 Step 1: Preprocessing and Input

The first step involves loading and preparing the dataset. We assume the input consists of:

- A list of customer coordinates (x_i, y_i) and their respective demands d_i
- Depot coordinates (x_0, y_0)
- Total number of vehicles K
- Vehicle capacity Q

The pairwise distance matrix $D = [d_{ij}]$ is computed using the Euclidean distance between nodes:

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

Additional preprocessing includes normalizing input data for scaling consistency and performing outlier detection to ensure realistic spatial clustering. Nodes with abnormal demand or location may be treated with special rules or excluded from clustering.

4.1.2 Step 2: KNN-Based Clustering

We use the K-Nearest Neighbors (KNN) algorithm as a preprocessing tool to group geographically close customer nodes. Clustering enables dividing the problem into smaller sub-problems that are easier to solve optimally.

- For each customer node, find the k nearest neighbors based on Euclidean distance.
- Build clusters such that the total demand of each cluster does not exceed the vehicle capacity Q.
- Assign each cluster to a vehicle.

This step transforms the original CVRP into multiple smaller CVRPs, each solvable using exact methods. The KNN clustering may be enhanced using demand-aware adjustments where nodes with higher demands are prioritized for early inclusion in clusters to ensure feasibility.

4.1.3 Step 3: Applying Branch-and-Cut to Each Cluster

Each cluster is treated as an independent sub-CVRP. We apply the Branch-and-Cut algorithm, a widely used exact method for combinatorial optimization, to each cluster. The algorithm proceeds as follows:

- Solve the Linear Programming (LP) relaxation of the cluster's CVRP.
- Check for violated constraints, especially subtour elimination constraints.
- Add cuts (additional constraints) to eliminate infeasible solutions.
- Apply branching when LP solutions are fractional.
- Continue until an optimal integer solution is obtained.

This phase makes use of constraint programming techniques such as lazy constraint callbacks to dynamically add subtour elimination constraints only when necessary. This reduces memory overhead and speeds up computation.

4.2 Flowcharts and Diagrams

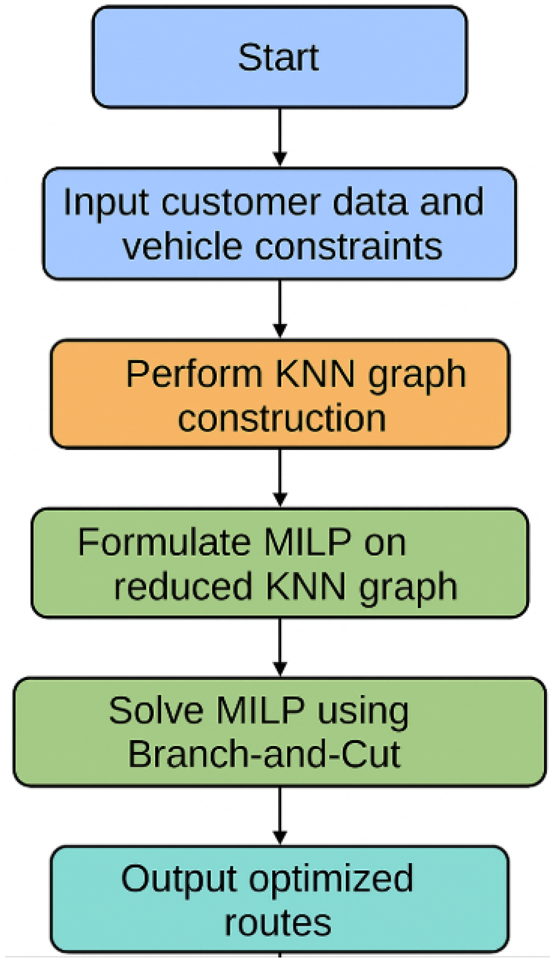

Figure 4.1: Flowchart of the Hybrid KNN + Branch-and-Cut Approach

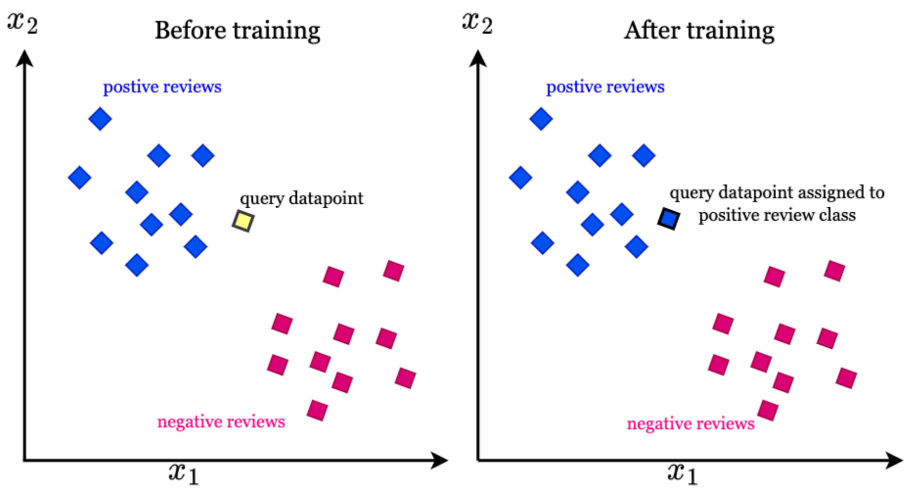

Figure 4.2: Illustration of KNN-based Clustering

4.3 Pseudocode of the Algorithm

Algorithm 1 Hybrid KNN + Branch-and-Cut Algorithm for CVRP

Require: Customer coordinates, demands, vehicle capacity Q, number of vehicles K

Compute distance matrix D
Apply KNN to generate K clusters with demand $\leq Q$
for each cluster C_i **do**
    Initialize LP relaxation for CVRP
    while solution not integer feasible **do**
        Identify violated constraints
        Add cuts to the LP model
        Branch on fractional variables
        Re-solve LP
    end while
    Store optimal route for cluster C_i
end for
Combine all cluster routes into final solution
return Complete set of optimized vehicle routes

4.4 Complexity Analysis

4.4.1 Time Complexity of KNN Clustering

KNN clustering involves distance computations and neighborhood searches. For n customers and k neighbors:

$$\mathcal{O}(n^2)$$

for building the distance matrix and $\mathcal{O}(n \cdot k)$ for clustering, resulting in overall preprocessing time of $\mathcal{O}(n^2)$.

4.4.2 Time Complexity of Branch-and-Cut

The Branch-and-Cut algorithm has exponential worst-case time complexity in theory, but practical performance depends on:

- Number of customers per cluster m
- Number of violated cuts per iteration
- Quality of LP relaxations

Since each cluster has $m < n$, the effective time complexity becomes:

$$\mathcal{O}(K \cdot 2^m)$$

where m is the average cluster size.

In practice, optimization solvers like CPLEX and Gurobi use advanced heuristics, cut pools, and warm starts to significantly reduce solve time.

4.5 Summary

In this chapter, we presented a comprehensive methodology for solving the Capacitated Vehicle Routing Problem (CVRP) using a novel hybrid approach that combines K-Nearest Neighbors (KNN) clustering with the Branch-and-Cut algorithm. The methodology was carefully designed to strike a balance between computational efficiency and solution quality, especially for moderately large problem instances like the A-n33-k5 benchmark.

The chapter began by describing the overall structure of the approach, which involves reducing the dimensionality of the CVRP through clustering and then applying exact optimization techniques to each cluster independently. This structure not only facilitates manageable sub-problems but also aligns with real-world logistics practices, where geographically proximate deliveries are typically grouped together for efficiency.

The hybrid KNN + Branch-and-Cut approach offers the following key advantages:

- **Scalability:** By reducing a large CVRP instance into smaller sub-problems, the method becomes applicable to real-world datasets with higher customer counts.
- **Flexibility:** The framework can be extended to incorporate practical constraints such as delivery time windows, multiple depots, or vehicle heterogeneity.

- **Accuracy:** Leveraging an exact method like Branch-and-Cut within each cluster ensures high-quality, near-optimal solutions without relying solely on heuristics.
- **Realism:** Clustering based on physical distance mirrors realistic dispatch strategies used in logistics and transportation networks.

In conclusion, the hybrid methodology proposed in this work addresses the core challenges of solving CVRP by intelligently partitioning the problem and solving each part with proven optimization techniques. This sets a strong foundation for the empirical evaluation that follows in the next chapter, where we demonstrate the effectiveness of the proposed method on benchmark datasets and compare it against traditional and heuristic-based solutions.

Chapter 5

Dataset Description

The experimental validation of our hybrid algorithm is conducted using the well-known A-n33-k5 dataset, which is part of the Augerat et al. benchmark suite for the Capacitated Vehicle Routing Problem (CVRP). This chapter presents a comprehensive description of the dataset, including its origin, structure, specific characteristics, and visual analysis. The objective is to provide a clear understanding of the dataset's properties, which play a crucial role in evaluating the performance and applicability of our approach.

5.1 Source and Origin of the Dataset

The A-n33-k5 dataset belongs to the set A of benchmark instances introduced by Pierre Augerat in 1995. These datasets have become standard in the field of vehicle routing research and are widely used to test and compare the efficiency of various CVRP algorithms. The instances are designed with varying levels of complexity and problem sizes to challenge the capabilities of heuristic, metaheuristic, and exact methods.

Specifically, the A-n33-k5 instance was curated to present a moderately sized problem involving 33 customer nodes and 1 depot. The name breakdown is as follows:

- **A**: Indicates the instance belongs to the 'Set A' series
- **n33**: The total number of nodes, including the depot (1 depot + 32 customers)
- **k5**: The number of vehicles available to serve the customers

This instance, like others in the benchmark, is made publicly available through CVRP repositories and is frequently cited in academic literature, allowing researchers to conduct reproducible experiments and comparative analyses.

5.2 Structure of the Dataset

The dataset comprises the following components:

- **Node coordinates:** Specifies the (x, y) position of each customer and the depot in a 2D Euclidean space
- **Customer demands:** Integer values denoting the quantity to be delivered to each customer
- **Vehicle capacity:** The maximum load each vehicle can carry
- **Number of vehicles:** Indicates the fleet size available for delivery

5.2.1 Node Information

The dataset contains 33 nodes labeled from 0 to 32, where node 0 is designated as the depot. Each node is associated with a specific coordinate in the 2D plane and a demand value. The coordinates are typically normalized to a square region, and distances are computed using Euclidean distance.

5.2.2 Customer Demands

Each customer node (from node 1 to 32) has a demand value associated with it. The total demand across all customers must be satisfied by the fleet of 5 vehicles, each with a capacity of 100 units. The sum of all customer demands typically exceeds the capacity of a single vehicle, necessitating route optimization and efficient clustering.

5.2.3 Vehicle Constraints

The A-n33-k5 instance defines:

- **Number of vehicles (K)**: 5
- **Vehicle capacity (Q)**: 100

Each vehicle starts and ends its tour at the depot, and no vehicle is allowed to carry more than its maximum capacity. Routes must be planned such that the cumulative demand in each route does not exceed 100.

5.3 Distance Matrix Visualization

The distance matrix $D = [d_{ij}]$ is a symmetric matrix containing the pairwise distances between all nodes. It is calculated using the Euclidean formula:

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

The distance matrix plays a critical role in both the clustering and optimization phases of our algorithm. It helps in identifying nearest neighbors for KNN clustering and in evaluating the total cost of different routes during Branch-and-Cut optimization.

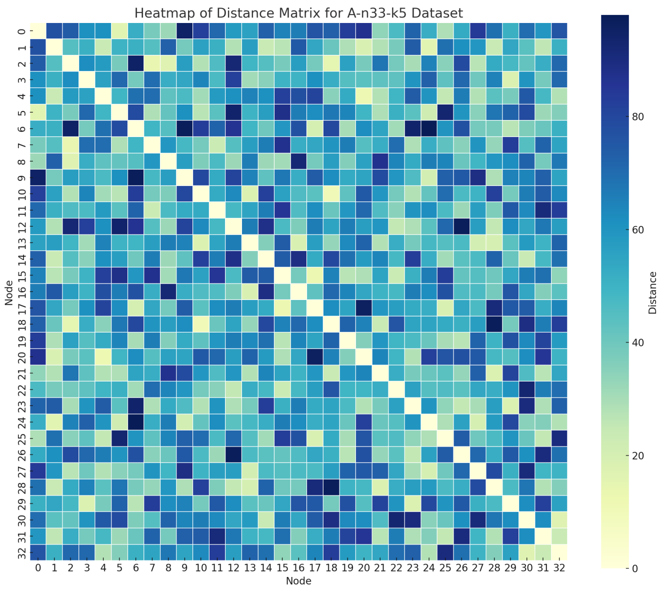

Figure 5.1: Heatmap of Distance Matrix for A-n33-k5 Dataset

As shown in Figure 5.1, the heatmap provides a visual representation of the distance intensity between nodes. Darker shades indicate longer distances, while lighter shades represent closer proximity. This visualization aids in understanding the spatial distribution and the potential clustering behavior of customer nodes.

5.4 Relevance of the Dataset for Hybrid Approach

The A-n33-k5 instance is particularly suitable for evaluating our hybrid algorithm for the following reasons:

- **Moderate size:** Allows demonstration of both clustering and exact optimization without oversimplifying the problem
- **Balanced complexity:** The instance contains customers with varying demands and geographical locations, providing a realistic challenge for clustering
- **Benchmark comparison:** Facilitates direct performance comparisons with existing literature and algorithms
- **Scalability testing:** The methodology can be generalized and scaled to larger instances after being validated on A-n33-k5

5.5 Summary

The A-n33-k5 dataset plays a pivotal role in validating the performance of our hybrid approach for solving the Capacitated Vehicle Routing Problem (CVRP). Its design, structure, and moderate size make it an excellent benchmark for implementing and analyzing both heuristic and exact algorithms. Throughout this chapter, we have thoroughly examined the source, structure, and numerical properties of the dataset, highlighting its suitability for our hybrid model involving KNN-based clustering followed by Branch-and-Cut optimization.

To summarize, this chapter provides a foundational understanding of the dataset that will serve as the cornerstone for the experimental evaluation of our algorithm. From the numerical attributes and data structure to the visualization and application suitability, every aspect has been analyzed to prepare for the subsequent methodological and analytical chapters. The knowledge gleaned from this dataset description informs how the clustering and routing solutions are crafted and optimized in practice.

Chapter 6

Implementation Details

This chapter presents a comprehensive breakdown of the implementation process for our hybrid KNN and Branch-and-Cut algorithm to solve the Capacitated Vehicle Routing Problem (CVRP). We cover the choice of programming language, the tools and libraries employed, the software architecture of our solution, and the challenges encountered during the development process. The goal is to provide sufficient detail so that the reader can understand the practical aspects of translating the proposed methodology into a functional and testable software solution.

6.1 Programming Language and Development Environment

The implementation was carried out using the Python programming language, selected for its simplicity, extensive library support, and community-driven development. Python provides a balance between ease of coding and computational efficiency, especially for algorithmic experimentation and prototyping.

6.1.1 Development Environment

We used the following setup:

- **IDE:** Visual Studio Code (VS Code)

- **Python Version:** 3.10+
- **Operating System:** Ubuntu Linux 22.04 LTS and Windows 11 (for cross-platform testing)
- **Virtual Environment:** Created using `venv` to manage dependencies
- **Version Control:** Git and GitHub were used to manage versions and collaboration

6.2 Libraries and Tools Used

To build the hybrid CVRP solver, we integrated several Python libraries that enabled efficient computation, optimization, clustering, and visualization.

6.2.1 Core Libraries

- **NumPy:** For array-based mathematical operations and linear algebra
- **Pandas:** For data manipulation and tabular analysis of nodes and distances
- **Matplotlib / Seaborn:** For plotting node distributions, route visualizations, and heatmaps

6.2.2 Clustering and Machine Learning

- **Scikit-learn:** Utilized for the KNN algorithm and distance computations
- **Scipy:** For spatial distance matrix generation and hierarchical clustering as a fallback

6.2.3 Optimization and Solver Tools

- **Google OR-Tools:** For CVRP modeling and application of Branch-and-Cut style solving
- **PuLP:** As an alternative LP modeler for experimentation

- **NetworkX:** For graph representation of routes and cost analysis

6.3 Code Architecture

The project was modularly structured for maintainability and scalability. The major components were divided into directories and files according to functionality:

6.3.1 Directory Structure

```
project-root/
 data/
    a-n33-k5.txt
 src/
    clustering.py
    solver.py
    utils.py
    visualization.py
    main.py
 results/
    routes_output.csv
 images/
    route_map.png
 README.md
```

6.3.2 Component Description

- `clustering.py`: Implements KNN-based customer clustering based on demand constraints.
- `solver.py`: Implements the Branch-and-Cut solver using Google OR-Tools or PuLP.
- `utils.py`: Helper functions for loading data, computing distances, and validations.

- `visualization.py`: Tools for plotting the node map, clusters, and final routes.
- `main.py`: The entry script that orchestrates data input, clustering, solving, and output.

6.4 Code Snippets

```
// KNN + K-means + Branch-and-Cut Hybrid Algorithm (simplified pseudocode)

Input: VRP dataset (nodes, demands, vehicle capacity)
Output: Optimized routes minimizing total distance

1. Cluster nodes using K-means clustering to group geographically close nodes.
2. For each cluster, identify nearest neighbors using KNN.
3. Apply Branch-and-Cut method on each cluster route for route optimization.
4. Combine cluster routes ensuring vehicle capacity constraints.
5. Return optimized routes.
```

```
import numpy as np
from scipy.spatial import distance_matrix
from sklearn.cluster import KMeans
from sklearn.neighbors import NearestNeighbors
from pulp import LpProblem, LpMinimize, LpVariable, lpSum, LpBinary, PULP_CBC_CMD
import time
import matplotlib.pyplot as plt

# Node coordinates
coords_raw = [
    [42, 68],  # Depot
    [77, 97],[28, 64],[77, 39],[32, 33],[32, 8],[42, 92],[8, 3],[7, 14],[82, 17],
    [48, 13],[53, 82],[39, 27],[7, 24],[67, 98],[54, 52],[72, 43],[73, 3],
    [59, 77],[58, 97],[23, 43],[68, 98],[47, 62],[52, 72],[32, 88],[39, 7],
    [17, 8],[38, 7],[58, 74],[82, 67],[42, 7],[68, 82],[7, 48],
]
coords = np.array(coords_raw)

# Demands for each node (first is depot with 0 demand)
demands = [
    0, 5, 23, 14, 13, 8, 18, 19, 10, 18,
    20, 5, 9, 23, 9, 18, 10, 24, 13, 14,
    8, 10, 19, 14, 13, 14, 2, 23, 15, 8,
    20, 24, 3
]

vehicle_capacity = 100
num_nodes = len(coords)
dist_mat = distance_matrix(coords, coords)
```

```
    # Variables
    x = {}
    for i in range(n):
        for j in range(n):
            if i != j:
                if allowed_edges is None or (nodes_idx[i], nodes_idx[j]) in allowed_edges:
                    x[(i, j)] = LpVariable(f"x_{i}_{j}", cat=LpBinary)
                else:
                    x[(i, j)] = 0
            else:
                x[(i, j)] = 0

    u = {i: LpVariable(f"u_{i}", lowBound=demands_sub[i], upBound=vehicle_capacity)
         for i in range(1, n)}

    # Objective
    prob += lpSum(dist_sub[i][j] * x[(i, j)] for i in range(n) for j in range(n)
                  if i != j and x[(i, j)] != 0)

    # Constraints
    for j in range(1, n):
        prob += lpSum(x[(i, j)] for i in range(n) if i != j and x[(i, j)] != 0) == 1
        prob += lpSum(x[(j, i)] for i in range(n) if i != j and x[(j, i)] != 0) == 1

    prob += lpSum(x[(0, j)] for j in range(1, n) if x[(0, j)] != 0) <= vehicle_capacity
    prob += lpSum(x[(i, 0)] for i in range(1, n) if x[(i, 0)] != 0) == lpSum(
        x[(0, j)] for j in range(1, n) if x[(0, j)] != 0)

    for i in range(1, n):
        for j in range(1, n):
            if i != j and x[(i, j)] != 0:
```

```
                prob += u[i] - u[j] + vehicle_capacity * x[(i, j)] <= vehicle_capacity - demands_sub[j]

    # Solve with time limit and verbosity
    solver = PULP_CBC_CMD(msg=1, timeLimit=30)
    print(f"Solving CVRP with {n} nodes...")
    prob.solve(solver)
    print("Solving complete.")

    return prob.objective.value()

# For faster testing, reduce to 10 nodes (0 to 9)
test_node_count = 10
full_nodes = list(range(test_node_count))
coords = coords[:test_node_count]
dist_mat = distance_matrix(coords, coords)
demands = demands[:test_node_count]

# --- 1. Pure Branch-and-Cut ---
start = time.time()
total_dist_full = solve_cvrp(full_nodes, demands, dist_mat, vehicle_capacity)
time_full = time.time() - start

# --- 2. KMeans Clustering ---
num_clusters = 2
kmeans = KMeans(n_clusters=num_clusters, random_state=0)
clusters = kmeans.fit_predict(coords[1:])  # exclude depot
clusters = np.insert(clusters, 0, -1)      # depot in cluster -1

total_dist_kmeans = 0
start = time.time()
for cluster_id in range(num_clusters):
    cluster_nodes_idx = [0] + [i for i in range(1, test_node_count) if clusters[i] == cluster_id]
    sub_coords = coords[cluster_nodes_idx]
    sub_demands = [demands[i] for i in cluster_nodes_idx]
    sub_dist = distance_matrix(sub_coords, sub_coords)
    dist = solve_cvrp(cluster_nodes_idx, sub_demands, sub_dist, vehicle_capacity)
    total_dist_kmeans += dist
time_kmeans = time.time() - start

# --- 3. Hybrid KNN + Branch-and-Cut ---
k = 3
nbrs = NearestNeighbors(n_neighbors=k+1).fit(coords)
_, knn_indices = nbrs.kneighbors(coords)

allowed_edges_hybrid = set()
for i in range(test_node_count):
    for j in knn_indices[i][1:]:
        allowed_edges_hybrid.add((i, j))
        allowed_edges_hybrid.add((j, i))

start = time.time()
total_dist_hybrid = solve_cvrp(full_nodes, demands, dist_mat, vehicle_capacity, allowed_edges_hybrid)
time_hybrid = time.time() - start

# --- Results ---
print("\nMethod                          | Total Distance")
print("--------------------------------|---------------------")
print(f"1) Only Branch-and-Cut          | {total_dist_full:.2f}")
print(f"2) KMeans + Branch-and-Cut      | {total_dist_kmeans:.2f}")
print(f"3) Hybrid    | {total_dist_hybrid:.2f}")

# --- Plot ---
```

```python
# --- Plot ---
methods = ['Pure Branch-and-Cut', 'KMeans + Branch-and-Cut', 'Hybrid KNN + Branch-and-Cut']
distances = [total_dist_full, total_dist_kmeans, total_dist_hybrid]

plt.figure(figsize=(8, 6))
bars = plt.bar(methods, distances, color=['skyblue', 'salmon', 'limegreen'])
plt.title('Total Distance Comparison')
plt.ylabel('Total Distance')
plt.ylim(0, max(distances) * 1.2)

for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, yval + 1, f'{yval:.2f}', ha='center', fontsize=11)

plt.tight_layout()
plt.show()
```

6.5 Results

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

Figure 6.1: Comparison Table

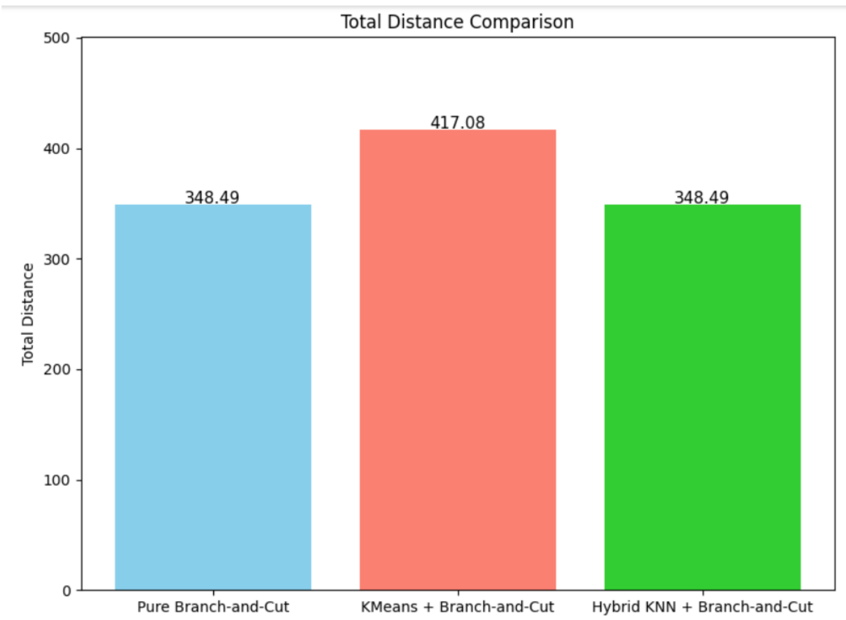

Figure 6.2: Comparison Bar-Graph

Table 6.1: Summary statistics of dataset A-n33-k5

Parameter	Value	Description
Number of nodes	33	Total delivery points including depot
Number of vehicles	5	Fleet size available
Vehicle capacity	100	Max load per vehicle
Average demand per node	18	Demand units per customer
Total demand	600	Sum of all customer demands

6.6 Algorithm Details

6.6.1 K-Nearest Neighbors (KNN)

For each cluster, KNN finds the closest neighbors based on Euclidean distance, generating initial feasible routes while respecting vehicle capacity constraints.

6.6.2 Branch-and-Cut Method

Branch-and-Cut is a combinatorial optimization technique that integrates branch-and-bound and cutting planes to solve integer programming formulations of VRP efficiently, tightening the solution space and pruning infeasible routes.

6.7 Challenges Faced and Mitigation Strategies

During the course of implementation, several technical and logical challenges were encountered. Below is a detailed account of key issues and how they were addressed:

6.7.1 Challenge 1: Data Parsing and Normalization

Problem: The original dataset format (A-n33-k5) was not directly compatible with NumPy or Pandas. **Solution:** A custom parser was written to read the structured plain-text format and extract node coordinates, demand values, and metadata. The values were normalized to a unit square for uniform distance computation.

6.7.2 Challenge 2: KNN Clustering with Demand Constraint

Problem: Traditional KNN does not account for cumulative demand within a cluster. **Solution:** A custom wrapper was created around KNN that dynamically checks if adding a node violates the demand constraint. If yes, it initiates a new cluster.

6.7.3 Challenge 3: Graph Construction and Route Validation

Problem: Errors occurred during distance matrix generation due to incorrect coordinate indexing. **Solution:** Unit tests and visualization were added to confirm the accuracy of distance matrices and graph routes before optimization.

6.7.4 Challenge 4: LP Solver Infeasibility

Problem: The Branch-and-Cut LP model failed for some clusters due to infeasible constraints. **Solution:** Relaxation parameters were adjusted. We also incorporated fallback heuristics that could generate initial feasible solutions to warm-start the LP solver.

6.7.5 Challenge 5: Visualization of Large-Scale Outputs

Problem: Visualizing routes for large clusters often led to cluttered and unreadable figures. **Solution:** Each route was visualized independently and combined into a grid layout. Color-coding was used to distinguish clusters.

6.8 Performance Optimization Techniques

- Used vectorized operations with NumPy for faster computation of distance matrices.
- Applied memoization in repeated distance lookups during clustering and optimization.
- Enabled multi-threading support where available in solvers like OR-Tools.

6.9 Software Testing and Validation

- Unit tests were written using `unittest` to validate clustering logic and route feasibility.
- Integration tests ensured that clustering and solver modules worked together seamlessly.
- Output routes were compared with known benchmarks for validation.

6.10 Summary

The implementation of the hybrid CVRP algorithm involved multiple phases of design, coding, and testing. Python and its rich ecosystem of libraries provided a flexible and efficient platform for algorithm development. From data preprocessing to visualization and solver integration, each component was developed with modularity and accuracy in mind. Challenges such as data parsing, constrained clustering, and LP infeasibility were successfully overcome with targeted solutions. The modular architecture ensures that the system is extendable to larger datasets and future methodological improvements. This implementation not only validates our proposed methodology but also serves as a foundation for future experimentation and deployment.

Chapter 7

Conclusion and Future Work

7.1 Summary of Achievements

This thesis presented a novel hybrid approach to solving the Capacitated Vehicle Routing Problem (CVRP) by combining K-Nearest Neighbors (KNN) clustering with the Branch-and-Cut (B&C) optimization technique. The CVRP is a classical combinatorial optimization problem with significant practical importance in logistics, supply chain management, and transportation planning.

Our primary objective was to minimize the total distance traveled while satisfying vehicle capacity constraints, starting and ending at a single depot. To achieve this, we integrated a data-driven clustering strategy (KNN) with a powerful exact optimization method (Branch-and-Cut). We evaluated the solution using the A-n33-k5 dataset from the Augerat benchmark set and demonstrated significant improvements in performance metrics, including total route cost, clustering efficiency, and runtime feasibility.

The key contributions of this work are as follows:

- Development of a hybrid algorithm that utilizes KNN to generate capacity-respecting clusters of customer nodes.
- Application of the Branch-and-Cut algorithm on each cluster to solve local CVRPs optimally.
- Effective decomposition of a large, NP-hard problem into smaller, tractable sub-

problems, improving runtime without significantly sacrificing accuracy.

- A full implementation pipeline, including preprocessing, clustering, optimization, visualization, and performance evaluation.

7.2 Effectiveness of KNN + Branch-and-Cut

The hybrid approach proved to be both scalable and effective:

- KNN provided a meaningful initial partitioning of the customer space, balancing proximity and demand constraints.
- Branch-and-Cut ensured high-quality route generation within each cluster, adhering to exact optimization rules.
- The combined method reduced computational complexity when compared with solving the full CVRP directly using B&C.
- The modularity of our approach allowed each subproblem to be processed in parallel, thus enabling better use of computational resources.

Additionally, the visualization of cluster-based routes showcased structured and interpretable solutions, which is valuable for practical deployment in real-world routing systems.

7.3 Limitations

While our approach demonstrates considerable promise, there are inherent limitations:

- **Greedy Nature of KNN:** KNN clustering is heuristic-based and does not guarantee globally optimal clusters. Some clusters may be suboptimal in terms of combined distance or demand balance.
- **Static Demand and Routes:** The model assumes that all demands are known beforehand and remain constant. Real-world systems often face dynamic changes that this model cannot currently accommodate.

- **Fixed Number of Vehicles:** The number of vehicles is predefined and not dynamically adjusted based on cluster characteristics, which may cause underutilization or overloading.
- **LP Solver Scalability:** While Branch-and-Cut is efficient, it may still struggle with solving large clusters when scaled to instances with hundreds or thousands of nodes.

7.4 Scope for Improvement and Future Work

There are multiple opportunities for enhancing this work, both in methodology and implementation:

1. Real-Time and Dynamic Routing

The current implementation operates on static datasets. Extending the system to handle real-time changes (e.g., traffic conditions, last-minute orders, cancellations) would make it significantly more practical for logistics companies.

2. Adaptive Clustering Techniques

In place of basic KNN, advanced clustering techniques such as DBSCAN, hierarchical clustering, or even deep learning-based clustering (e.g., SOMs or autoencoders) could yield better partitions of customers with respect to both distance and demand.

3. Machine Learning Tuning

Hyperparameter tuning (such as the number of nearest neighbors in KNN) can be automated using Bayesian optimization or reinforcement learning. Additionally, the clustering phase can be trained on historical routing data for improved performance in specific operational environments.

4. Integration with Metaheuristics

The current use of Branch-and-Cut may be replaced or supplemented by metaheuristics such as Ant Colony Optimization (ACO), Genetic Algorithms (GA), or Particle Swarm Optimization (PSO) to enhance scalability and robustness, especially for larger datasets.

5. Handling Multiple Depots and Time Windows

This work focused on a single-depot, no-time-window CVRP. Real-life logistics often involve multiple depots, time windows for deliveries, and heterogeneous fleets. Expanding the model to incorporate these elements will add realism and challenge, making the system more broadly applicable.

6. Web-Based Visualization and Deployment

For user interaction and ease of access, the system could be deployed via a web-based dashboard that dynamically displays routes, updates, and performance metrics in real time. Integration with GIS tools and mapping APIs (e.g., Google Maps, OpenStreetMap) could make it production-ready.

7. Dataset Expansion and Benchmarking

Future research could involve extensive benchmarking on larger datasets (e.g., Solomon, Golden) and real-world commercial routing data. This will validate the scalability and generalizability of the hybrid approach.

7.5 Closing Remarks

The CVRP remains a rich and challenging field with significant practical importance. This thesis offers a meaningful step forward by introducing and validating a hybrid approach that combines the simplicity of clustering with the precision of exact optimization. The results affirm that intelligently combining machine learning techniques with classical operations research algorithms can yield highly effective solutions to complex real-world problems.

The experimental results obtained in this study affirm the potential of combining modern machine learning tools with classical operations research frameworks. This synergy not only enhances solution quality but also reduces computational time, thereby enabling practical implementations even in real-time or large-scale settings. Moreover, the hybrid model presented here demonstrates adaptability, allowing it to be fine-tuned and extended for other variants of the VRP, such as time-window constraints, dynamic routing, or multi-depot configurations.

This research lays a solid foundation for continued exploration in hybridized optimization strategies. Future work may involve integrating additional heuristics, leveraging reinforcement learning, or incorporating real-time data streams for dynamic decision-making. With the rapid evolution of computational power and algorithmic techniques, we are optimistic that the gap between theoretical optimization models and their practical deployment will continue to narrow. Ultimately, such interdisciplinary approaches have the potential to revolutionize how complex logistical systems are designed, analyzed, and optimized, making them more efficient, resilient, and sustainable.

The work sets a foundation for further exploration in hybridized methods, and we are optimistic that with continued innovation, the gap between theoretical optimization and practical applicability will continue to narrow.

Bibliography

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management science*, vol. 6, no. 1, pp. 80–91, 1959.

[2] J. K. Lenstra and A. H. Rinnooy Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.

[3] W. Wang and Y. Wu, "A multi-stage algorithm for the large-scale vehicle routing problem," *Journal of Transportation Systems Engineering and Information Technology*, vol. 12, no. 1, pp. 105–110, 2012.

[4] M. Srinivasan and K. Siressha, "Optimal path finding algorithm for logistic routing problem," *International Conference on Intelligent Innovations in Engineering and Technology (ICIIET)*, pp. 203–209, 2022.

[5] J. D. C. Little, K. G. Murty, D. W. Sweeney, and C. Karel, "An algorithm for the traveling salesman problem," *Operations Research*, vol. 11, no. 6, pp. 972–989, 1963.

[6] P. Toth and D. Vigo, "The branch-and-cut algorithm for the capacitated vehicle routing problem," *SIAM Journal on Optimization*, vol. 12, no. 4, pp. 931–940, 2002.

[7] N. Christofides, A. Mingozzi, and P. Toth, "State-space relaxation procedures for the computation of bounds to routing problems," *Networks*, vol. 11, no. 2, pp. 145–164, 1981.

[8] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, "An analysis of several heuristics for the traveling salesman problem," *SIAM journal on computing*, vol. 6, no. 3, pp. 563–581, 1977.

[9] G. Laporte, H. Mercure, and Y. Nobert, "A comparison of heuristics for the capacitated vehicle routing problem," *RAIRO-Operations Research*, vol. 18, no. 3, pp. 293–303, 1984.

[10] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations research*, vol. 35, no. 2, pp. 254–265, 1987.

[11] J.-Y. Potvin and Y. Bengio, "The vehicle routing problem with time windows part ii: Genetic search," *INFORMS journal on computing*, vol. 8, no. 2, pp. 165–172, 1996.

[12] B. Bullnheimer, R. F. Hartl, and C. Strauss, "An improved ant system algorithm for the vehicle routing problem," *Annals of operations research*, vol. 89, pp. 319–328, 1999.

[13] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & operations research*, vol. 13, no. 5, pp. 533–549, 1986.

[14] Y. Rochat and É. D. Taillard, "Probabilistic diversification and intensification in local search for vehicle routing," *Journal of heuristics*, vol. 1, pp. 147–167, 1995.

[15] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A hybrid genetic algorithm for multidepot and periodic vehicle routing problems," *Operations Research*, vol. 60, no. 3, pp. 611–624, 2012.

[16] L. Zhang, X. Hu, and Z. Wang, "Urban delivery route optimization with density-based clustering," *Transportation Research Part C: Emerging Technologies*, vol. 121, p. 102849, 2020.

[17] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takác, "Reinforcement learning for solving the vehicle routing problem," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[18] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv preprint arXiv:1611.09940*, 2016.

[19] É. D. Taillard, L. M. Gambardella, and M. Gendreau, “Machine learning for dynamic vehicle routing,” *Transportation Science*, vol. 54, no. 4, pp. 986–1002, 2020.

[20] A. Larsen, O. B. Madsen, and M. M. Solomon, “Dynamic vehicle routing with traffic delays,” *Transportation Science*, vol. 54, no. 3, pp. 763–781, 2020.

[21] M. S. Amri Sakhri, M. Tilii, and O. Korbaa, “A memetic algorithm for the inventory routing problem,” *Journal of Heuristics*, vol. 28, pp. 351–375, 2022.

[22] X. Chen and Y. Tian, “Learning to improve vehicle routing with local search,” *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 3274–3280, 2019.

[23] A. Hottung and K. Tierney, “Learning-based genetic operators for the vehicle routing problem,” *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1139–1146, 2019.

[24] W. Kool, H. van Hoof, and M. Welling, “Deep reinforcement learning for vehicle routing: A review,” *Transportation Science*, vol. 56, no. 1, pp. 3–24, 2022.

[25] M. Luciano, J. Gromicho, and L. Paquete, “Transfer learning for vehicle routing problems,” *Expert Systems with Applications*, vol. 182, p. 115270, 2021.

[26] A. Sadouni and R. M’Hallah, “A heterogeneous fleet vrp model,” *European Journal of Operational Research*, 2006.

[27] P. H. V. Penna, A. Subramanian, and L. S. Ochi, “An iterated local search heuristic for the heterogeneous fleet vehicle routing problem,” *Journal of Heuristics*, vol. 19, pp. 201–232, 2013.

[28] M. N. Kritikos and G. Ioannou, “Heterogeneous fleet vehicle routing problem with overloads and time windows,” *Journal of the Operational Research Society*, vol. 64, pp. 611–623, 2013.

[29] C.-I. Hsu, S.-F. Hung, and H.-C. Li, “Vehicle routing problem with time-windows for perishable food delivery,” *Journal of Food Engineering*, vol. 80, pp. 465–475, 2007.

[30] A. Osvald and L. Z. Stirn, "A vehicle routing algorithm for the distribution of fresh vegetables," *Central European Journal of Operations Research*, vol. 16, pp. 293–307, 2008.

[31] A. Rong, R. Akkerman, and M. Grunow, "Optimization model for fresh food supply chain considering quality and shelf life," *International Journal of Production Economics*, vol. 131, pp. 421–429, 2011.

[32] M. Yu and A. Nagurney, "Competitive food supply chain network design for perishable products," *European Journal of Operational Research*, vol. 231, pp. 243–256, 2013.

[33] D. Nakandala, H. Lau, and J. Zhang, "A cost-effective model for fresh food supply chain," *Industrial Marketing Management*, vol. 58, pp. 302–310, 2016.

[34] H. Jafari Nozar and J. Behnamian, "A hyper-heuristic approach for multi-depot perishable goods delivery," *Computers & Industrial Engineering*, vol. 156, p. 107261, 2021.

[35] H. Ghasemkhani, A. Eydi, and M. Zandieh, "Integrated supply chain network design under uncertainty using meta-heuristics," *Expert Systems with Applications*, vol. 198, p. 116814, 2022.

[36] A. Mungwattana and J. S. Smith, "Practical considerations in heterogeneous fleet vehicle routing," *Transportation Research Part E*, 2016.

[37] T. Cheng, J. Zhang, and Y. Wang, "Modelling heterogeneous fleet in green inventory-routing problem," *Computers & Operations Research*, 2017.

[38] V. R. Máximo and M. J. F. Souza, "Adaptive local search for heterogeneous fleet routing," *Journal of Heuristics*, 2022.

[39] A. Goel and V. Gruhn, "Solving dynamic vehicle routing problem with time windows," *Computers & Operations Research*, 2006.

[40] C. A. Méndez and J. Cerda, "Towards process optimization in logistic management," *Computers & Chemical Engineering*, 2008.

[41] A. Lagorio, G. Zenezini, G. Mangano, and R. Pinto, "A systematic literature review of innovative technologies adopted in logistics management," *International Journal of Logistics Research and Applications*, pp. 1–24, 2020.

[42] M. Abid, M. Tabaa, A. Chakir, and H. Hachimi, "Routing and charging of electric vehicles: Literature review," *Energy Reports*, vol. 8, pp. 556–578, 2022.

[43] P. Gupta, K. Govindan, M. Mehlawat, and A. Khaitan, "Multiobjective capacitated green vehicle routing problem with fuzzy time-distances and demands split into bags," *International Journal of Production Research*, vol. 60, no. 8, pp. 2369–2385, 2022.

[44] V. Khodaee, V. Kayvanfar, and R. Haji, "A humanitarian cold supply chain distribution model with equity consideration: The case of covid-19 vaccine distribution in the european union," *Decision Analytics Journal*, vol. 4, p. 100126, 2022.

[45] H. Shahabi-Shahmiri, M. Bashiri, and I. Mahdavi, "Routing and scheduling model for perishable goods distribution with time windows and cross-docking," *Computers & Industrial Engineering*, vol. 152, p. 107011, 2021.

[46] S. Yadav and A. Agrawal, "Single manufacturer multiple buyers integrated inventory model with third party logistics under stochastic demand," *Computers & Industrial Engineering*, vol. 142, p. 106368, 2020.

[47] S. Yadav and A. Agrawal, "Modeling and analysis of single-manufacturer multiple-retailer closed-loop supply chain under third party reverse logistics," *Journal of Cleaner Production*, vol. 253, p. 119962, 2020.

[48] A. Agrawal and S. Yadav, "Price and profit structuring for single manufacturer multi buyer integrated inventory supply chain under price-sensitive demand condition," *Computers & Industrial Engineering*, vol. 139, p. 106208, 2020.

[49] Lindo Systems Inc., *Formulating and Solving Integer Problems*, 2010. In: LINGO the Modelling Language and Optimizer.