Vehicle Route Optimization for Ethiopian Pharmaceutical Delivery Using Reinforcement Learning Model

By

Beyene Dabi Neguse

A Thesis Submitted to the Department of Computer science and Engineering,

School of Electrical Engineering and Computing

Presented in Partial Fulfillment for the Degree of Masters of Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia
June 2022

Vehicle Route Optimization for Ethiopian Pharmaceutical Delivery Using Reinforcement Learning Model

By: Beyene Dabi Neguse

Advisor: Dr. Ketema Adere (Ph.D.)

A Thesis Submitted to the Department of Computer science and Engineering, School of Electrical Engineering and Computing

Presented in Partial Fulfillment for the Degree of Masters of Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

ADAMA, ETHIOPIA
June 2022

APPROVAL SHEET

The advisor of the thesis entitled "**Vehicle Route Optimization for Ethiopian Pharmaceutical Delivery Using Reinforcement Learning Model" and developed by Beyene Dabi Neguse.** Hear by certifying that the recommendation and Suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis

Ketema Adere (Ph.D.)		
Advisor	Signature	Date

We, the undersigned, members of the board of Examiners of the thesis by **Beyene Dabi Neguse** have read and evaluated the thesis entitled "**Reinforcement Learning Model for Solving the Vehicle Routing Problem in Pharmaceutical Delivery"** and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Masters of Science in Computer Science and Engineering.

Chairperson	Signature	Date
Internal Examiner	Signature	Date
External Examine	Signature	Date

Final approval and acceptance of the thesis are contingent upon submission of its Final copy to the office of postgraduate Studies (**OPGS**) through the Department Graduate Council (**DGC**) and School Graduate Committee (**SGC**)

Department Head	Signature	Date
School Dean	Signature	Date
Office of postgraduate studies. Dean	Signature	Date

DECLARATION

I hereby declare that this MSc thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis have been duly acknowledged.

Beyene Dabi Neguse		
Name	Signature	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that we have read the revised version of the thesis entitled "**Vehicle Route Optimization for Ethiopian Pharmaceutical Delivery Using Reinforcement Learning Model**" prepared under my/our guidance by Beyene Dabi Neguse Submitted in partial fulfillment of the requirement for the degree of master's degree of Computer Science and Engineering Therefore, I recommend submitting the revised of the thesis to the department following the applicable procedures.

Ketema Adere (Ph.D.)		
Advisor	Signature	Date

ACKNOWLEDGEMENT

First and above all, I would like to thank to the Almighty of God whose eternal and undying love kept me and for infinite mercy throughout my life and during this thesis work. Next, I would like to thank my advisor Dr. Ketema Adere for his effortless support and valuable comments throughout this thesis work Next, I would like to thank Mr. Endris M. instructor in ASTU who helped me and also Mr. Ayele T. in the data collection process and intelligent advice about the proposed topic.

Finally, I would like to thank my beloved mother Lommi Dechasa and father Dabi Niguse who sacrificed a lot for me. Also, I would like to thank my fiancée Nardos B. who has stood by me through all my travails, my absences, my fits of pique and impatience. She gave me support and help, discussed ideas and prevented several wrong turns. And to all my family members, friends and classmates without them this wouldn’t be possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ML	Machine Learning
RL	Reinforcement Learning
VR	Vehicle Routing
VRP	Vehicle Routing Problem
TSP	Travelling Salesman Problem
VRPPD	Vehicle Routing Problem with Pickups and Deliveries
CVRP	Capacitated Vehicle Routing Problem
GVRP	Green Vehicle Routing Problem
GA	Genetic Algorithm
VRPTW	Vehicle Routing Problem with Time Windows
PSO	Particle swarm Optimization
ACO	Ant Colony Optimization
EPSS	Ethiopian Pharmaceutical Supply Services
SDVRP	Split Delivery Vehicle Routing Problem
NP	Nondeterministic Polynomial
MDP	Markov Decision Process
ANN	Artificial Neural Networks
TS	Tabu Search
RNN	Recurrent Neural Network
ANN	Artificial Neural Networks
DHF	Direct Health Facility
IDHF	Indirect Health Facility
VRP20	Vehicle Routing Problem instances of 20 node
VRP50	Vehicle Routing Problem instances of 50 node
VRP100	Vehicle Routing Problem instances of 100 node

Abstract

This study considers the route optimization of EPSS pharmaceutical delivery in Ethiopia. The agency has 19 hubs across the country. Due to its centralized distribution of pharmaceutical products from a single hub in the cluster, it is experiencing issues as the number of facilities throughout the cluster rapidly increases. The goal is to lower transportation costs while maintaining a high degree of customer satisfaction. As a result, the focus is on the vehicle routing problem (VRP) within each facility of this enormous distribution chain. This study uses real data from the EPSS catchment area. One of the efforts to improve the quality of service is to provide optimization of the distribution process. Optimization can be done by adding a clustering approach to the existing reinforcement learning algorithm. The large coverage distribution by one vehicle takes more time and more cost. In the process of optimizing the route facility clustering method which has produced the clustered facilities. Then the clustered facilities are inserted into the RL in order to provide the best-optimized route. Furthermore, the delivery cost will be calculated based on the distance covered by the vehicle based on the optimal route. The results of this study are examined with the proposed 3 types of instances which are VRP20, VRP50, and VRP100. The researcher performed a statistical analysis of the results from various aspects and designed two experiments to evaluate. Finally, the researcher concludes the applicability approach for the selected type of problem and suggest directions in which researchers can improve the route optimization approach.

Keywords: Vehicle Routing Problem, VRP20, VRP50, VRP100, RL

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

In the world of industrial business, one of the activities that takes place is the distribution of products to customers. The challenge in logistics activities is to deliver products to customers at the right time and in the right place. Distribution is an important aspect of logistics and serves as a vital connection in the supply chain between the organization and its clients. A vehicle is required in a distribution process activity to transport the product and distribute it to multiple destination locations based on client demand [1].This distribution is one of the key activities performed by pharmaceutical companies and plays an important role in the effectiveness of business [2]. However, it is acknowledged that the traditional approach based on set routes does not meet the needs of health facilities and may be inefficient for pharmaceutical distribution in some circumstances [3]. To solve this delivery problem, Vehicle Routing Problem (VRP) is formulated. VRP aims to find an optimal routing from the depot to multiple customers so as to minimize the cost [4].

VRP is a combinatorial optimization problem that has been studied for decades in applied mathematics and computer science, with a wide range of exact and heuristic solutions available. [5]. VRP is a well-known NP- hard problem which has introduced by Dantzig and Ramser [6]. It is known to be considerably more computationally difficult than the Traveling Salesman Problem (TSP) [7]. When commodities are moved from a depot to serve individual clients who are dispersed, VRP occur. and taking into account how the cost of the distribution might be lowered while increasing net income [8].

The key challenge with VRP is determining the best path for delivering the product. Optimal routes serve the purpose of minimizing the overall transportation cost, minimizing the number of vehicles, minimum distance travelled, minimum travel time, or other objectives [9]. However, the type of constraints that must be adhered to, as well as the dynamic and/or stochastic characteristics present [10] , must be considered during route optimization. As a result, there are numerous variations of the problem.

VRP has a wide range of practical applications in logistics and transportation. The major versions of VRP are as follows, depending on the application: [4] it is capacitated vehicle routing problem (CVRP); it is split delivery VRP (SDVRP); it is VRP with time windows (VRPTW); and it has VRP pick and delivery (VRPPD). Among those VRP variants, the study focused only on CVRP. In this VRP problem, each customer is associated with quantities representing one demand to be delivered to the customer. It must ensure that the overall delivery on a route does not exceed the vehicle capacity at any point along the route, in addition to the constraint that the entire delivery on a route cannot exceed the vehicle capacity. Split deliveries are not permitted at this location [11]. Since companies have to use available vehicles for delivery efficiently, [2] the challenge involved in the distribution system is to minimize the total travel distance of the route.

Since the first VRP presented by Dantzig and Ramser [12]–[15], researchers have investigated a variety of solutions to VRP [8] such as exact algorithms and heuristic algorithms. Meta-heuristics are a kind of heuristics, which have widely been applied to VRPs. Moreover, because of its wide range of applications and NP-Hardness, which prevents the calculation of globally optimal solutions for large-scale and real-world issue cases. As a result, there is still room for further work in this area. Therefore, the main contribution of this thesis will be proposing a clustering approach for pharmaceutical delivery-based case studies based on EPSS custom datasets.

1.2. Motivation of the Study

It is now well established from a variety of studies indicates that a number of solutions have been recognized for route optimization problem but most of the researches done for transportation and traffics problem only. Over the past decade, most research in VRP has emphasized the use of heuristic algorithm significantly. However, because of its wide range of applications and NP-Hardness, which prevents the calculation of globally optimal solutions for large-scale and real-world issue cases. As a result, there is still room for further work in this area. Also, use of traditional system based on fixed routes makes delivering pharmaceutical products difficult. The above issue motivates the researcher to propose a solution to VRP. The main motivation of this work is studying and propose a clustering approach to optimize route.

1.3. Statement of the Problem

Due to the expanding global population, demand for drugs increased, so pharmaceutical distribution is one of the fastest-growing sectors and has become much more important. Costs in the distribution chain have contributed to the growing costs of medicines, according to research conducted by the European Union (EU) [16]. Ethiopian Pharmaceutical Supply Service (EPSS) is a government agency in charge of managing the supply chain for public health commodities at a low cost and expanding their reach in Ethiopia [17]. Like any company distributing a product, pharmaceutical societies offer product delivery services to ensure efficient delivery and fulfill the demand for pharmaceutical products. But there is a difference between the pharmaceuticals distribution and the general product distribution system since health facilities do not have a large surface area for storing large quantities of drugs. This leads to frequent and small volume delivery of pharmaceutical products.

Therefore, delivery vehicle route planning can be done in order to deliver products to multiple health facilities simultaneously in a cost-effective way. Concerning the VRP problem, the existing literature on routing optimization suggests that various problems can be solved using various methods such as integer programming, dynamic programming, particle swarm optimization, genetic algorithm, ant colony optimization, etc. [8], [18]–[21]. Generally, these methods are used for optimizing the cost associated with the problem to get the optimum combination of route and vehicle. The network which is constant with the weights of its links is called a static network. The weight of all ties is regarded as fixed in this state, and the problem is solved globally to find the optimal path between two nodes [22]. To solve issues in a distribution network, it is crucial to correctly identify active components, classify problems, and determine whether required parameters are appropriate. Given the limitations of the methods available, this research needs to emphasize developing a clustering approach to solve real problems with real-world instances.

According to [23], it has been observed that most of the recent literature focuses on Real VRPTW has some uncertain and/or fuzzy constraints than classical VRP. Furthermore, there is no baseline for a more realistic version of VRP, which has been confirmed. VRP arises in many applications, according to [24] , because of its wide range of applications and NP-Hardness, which prevents the calculation of globally optimal solutions for large-scale and real-world issue cases. Due to its

flexibility and computational efficiency as well as being very fast and capable of handling very large networks, the static network is still the preferred tool for strategic transport planning [22]. Still now, there happens no single study based on the distribution system of EPSS and its applicability with route optimization methods. As a result, there is still room for further work in this area. Therefore, this study is motivated to enhance the performance of routing by design a clustering approach for pharmaceutical delivery-based case studies based on EPSS custom datasets.

1.4. Research Questions

This study fills the gap and can support and improve the Ethiopian pharmaceutical delivery approach by optimizing routes.

To address the problems stated, the following questions designed

1. What is the major limitation in the current EPSS to optimize distribution route?
2. What is impact of clustering on route optimization?
3. What metrics and how can evaluate the performance of the proposed approach?

1.5. Objectives of the Study

1.5.1. General Objectives

The general objective of this research work is to investigate and design RL-based cluster-first route second approaches for solving vehicle routing problem in Ethiopia pharmaceuticals delivery.

1.5.2. Specific Objectives

To achieve the general objective, the following specific objectives are designed as follows.

- To review existing literatures related to ML and VRP
- Collect VR data in order to apply a machine learning algorithm.
- To design RL-based cluster-first route second approaches that can incorporate with the route data-based solution.
- Preparing experimental setup
- To test and evaluate the approach using clustered and un-clustered dataset.

1.6. Scope and Limitation of the Study

1.6.1. Scope of the Study

As mentioned in the above the aim of this study is to optimize pharmaceuticals delivery route based on route data which include distance of health facility from each other and from depot. In a data the working time and other constraints are excluded in this study. This study focuses only in distance data because the as it is known the major challenge here is to find the shortest route that can be used for delivery of pharmaceuticals product. During experimentation the study analyze the data by selecting different algorithms that can incorporate to the data based on literature and show their results with recommendation.

1.6.2. Limitation of the Study

This study is unable to encompass the entire VRP variants which are raised in literature. This study will not consider investigation of parameter such as capacity and delivery time. For those issue our study was limited.

1.7. Beneficiary of the Study

The main significance of this work is to optimize route data that could outperform the delivery of pharmaceutical products and minimize cost of delivery. This strategy should have been effective for any Ethiopian pharmaceutical sector, given a reasonable amount of route data. The second goal of this work was to learn about computational optimization techniques. As the world becomes more connected due to globalization and the Internet, the ability to track and measure these connections increases. In today's society more, data is being recorded than ever, yet the information being gleaned from these records is in its infancy. Learning the techniques that can be used to optimize routes was the ultimate objective, due to the importance it will play in the future of Ethiopian pharmaceutical delivery.

1.8. Organization of the Thesis Work

In below Figure we explain the frame of the thesis in a simple way to give the reader a general appearance. Chapter 1 will give the reader information about the problem, background and motives for study. This chapter is also included the significance of the study. Chapter 2 reviews literatures

that have studied about VRP that have done ever in the past. Chapter 3 clarify methodologies used for route optimization. Chapter 4 Implementation, experimentations and comparison of the results obtained. Also, the results of approach discussed. In the last chapter conclusions and recommendations are takes place and by stressing the thesis statement, and to leave some final ideas to the reader.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORK

This chapter provides a detailed review of the relevant literature to gain perception on this study. In the section some related works in the area of vehicle routing problem and its basic version investigation presented. The review includes the concept of vehicle routing problems, vehicle routing problems models, machine learning, machine learning algorithms, and related work.

2.1. Pharmaceutical Delivery in EPSS

The supply chain is the network of interconnected organizations that manufacture, manage, and distribute a certain product. Supply chain management involves the planning and directing of all operations included in sourcing and procurement. Collaboration and coordination between supply chain members, which might include departments inside the supply chain organization, suppliers, intermediates, third-party service providers, and customers, are all part of supply chain management. Supply chain management, in essence, integrates supply and demand management within and between businesses [25].

The government of Ethiopia is looking to avail pharmaceuticals at an affordable cost and increase its reachability to its citizens with the help of pharmaceuticals Supply agency [17]. The Ethiopian Pharmaceutical Supply Service (EPSS) is responsible for supply chain management of public health commodities in Ethiopia. The agency has 19 branch warehouses that serve over 3,800 health facilities, which serve 105 million people across nine regional states and two administrative states. EPSS is in charge of commodity distribution throughout a 1.1 million km2 area (425,000 square miles). Due to their distance or a lack of roads or other necessary access to distribute health commodities, some communities are difficult to reach. Understanding the coverage levels and gaps in EPSS's branch network is critical for EPSS and the Ministry of Health (MoH) to improve Ethiopians' access to important health commodities [26].

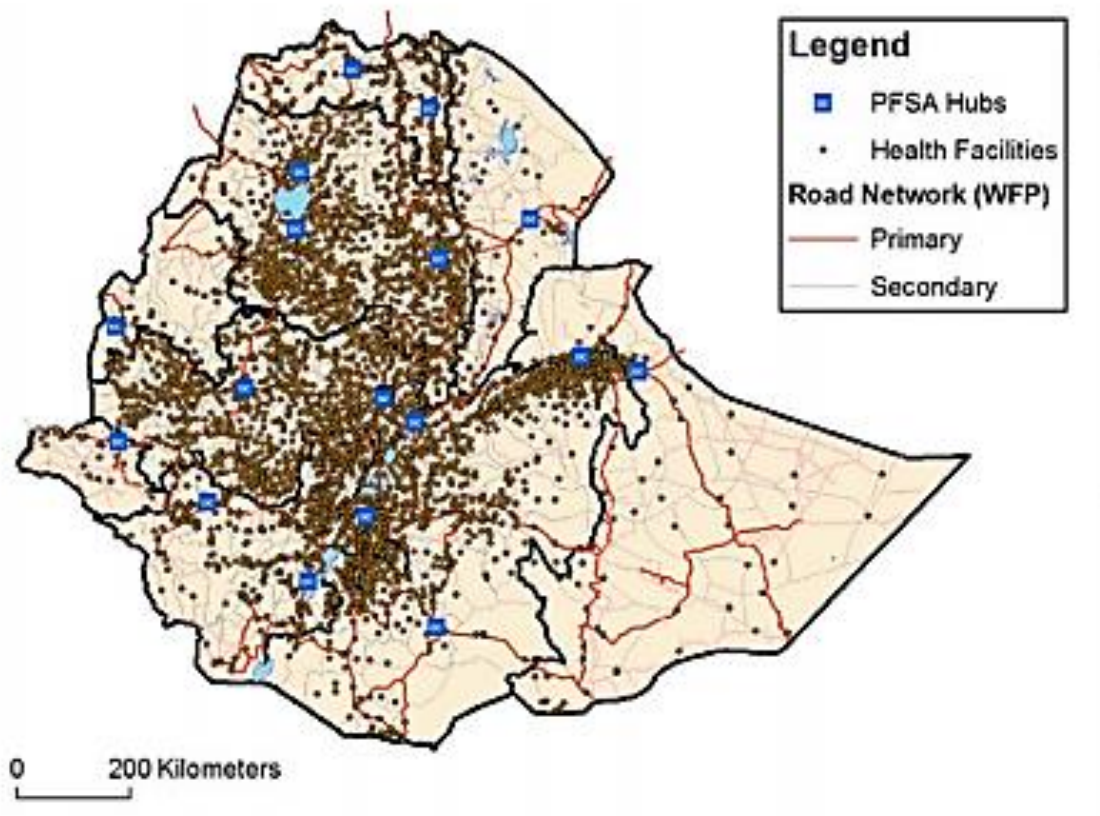


Figure 2.1: EPSS hubs and health facility distribution [25]

According to [27] In planning for direct delivery to woredas, a route optimization activity, using LLamasoft software, was undertaken with the support of John Snow, Inc (JSI). Detailed route maps were developed for each hub. However, application of these maps by hubs was inconsistent with many hubs preferring to use their own maps. A key operational step is just beginning: the current intention is to expand direct vaccine delivery from the current 180 health facilities being supplied to all health facilities, ideally with refrigerated vehicles. As of April 2019, EPSS was planning on an aggressive rollout of direct delivery to 1200 health facilities.

2.2. Vehicle Routing Problems

The Vehicle Routing Problem (VRP) usually occurs when there is movement of goods from a depot to serve specific customers dispersed. And taking into consideration how cost associated with the distribution can be minimized at an increased net income. It is frequently used in a variety of applications, including transportation delivery routing, urban school bus route planning, postal

delivery, rural school bus route planning, gasoline delivery trucks, urban waste collection, and snow plough [8]. A customer is an entity with a specific demand that necessitates the presence of a vehicle, a unit capable of moving between customers and the depot, and a unit that initially contains the customers' requests. The fleet is defined as the total group of vehicles. Moving a vehicle between the depot and the customers comes with a certain cost. A route is a sequence of visited customers by a certain vehicle, starting and ending at a depot. The goal of the Vehicle Routing Problem is to serve all customers, minimizing the total cost of the routes of all vehicles [28].

The Vehicle Routing Problem (VRP) is a general class of problems in which a fleet of vehicles with limited capacity based at one or several depots has to be routed serving a certain number of customers to minimize the number of routes and total travelling time as well as distance of all vehicles.[29] also describe as finding the minimum distance or cost of the combined routes of a number of vehicles m that must service a number of customers n.

This system is mathematically characterized as a weighted graph $G = (V, A, d)$, where $V = \{v0, v1, ., vn\}$, represents the vertices and $A = \{(vi, vj): i \neq j\}$ represents the arcs, there is a central depot where each vehicle begins its run, and each of the other vertices symbolizes one of the n clients. The quantity d_{ij}, which is measured using Euclidean computations, represents the distances connected with each arc. Each client has a non-negative demand q_i, and each vehicle has a capacity limitation (Q) [30].

As was pointed out in the [31], Consider for instance a fleet of capacitated vehicle $k \in K$ at a depot designated as node i = 1 and a set of customers to be visited, each having a request$j$ $(j = 2, \ldots\ldots, n)$. The tours of the vehicles at the depot on the set of arcs, say A $=\{(i, j): i, j = 1, \ldots\ldots, n | i \neq j\}$ such that the total distance covered in minimized while satisfying the requests at each node. Hence, the general objective in a VRP is formulated as

$$\sum_{k \in K} \sum_{(i,j) \in A} C_{ij} X_{ijk} \qquad (2.1)$$

Where C_{ij} is the cost of traveling from j to point i. The distance traveled, the time spent on each arc, the amount of fuel consumed, and other factors could all contribute to this cost. X_{ijk} is a binary variable that takes the value of 1 if vehicle k travels the arc (i, j) and 0. On the other hand,

multiple assumptions are commonly used in order to fully propose a VRP, resulting in varied constraints. Furthermore, the incorporation of these more sophisticated constraints is a major contributory aspect that has resulted in a variety of VRP variants. On the other hand, in spite of the mathematical formulation of VRP, the constraints listed below are introduced and mathematically formulated.

2.2.1. Net flow of vehicle

For net flow of vehicle, all vehicle routes are expected to start and end at the depot at the end of operation. This is usually expressed as

$$\sum_{i=1}^{n} X_{ijk} = \sum_{j=1}^{n} = 1 \quad \forall k \in K \tag{2.2}$$

K is the total number of vehicles at the depot.

2.2.2. Number of Service per customer

A possible explanation for this is that the demand needs to be served only once on each node. Therefore, the vehicle must get on and off at the node. This is expressed by the following formula.

$$\sum_{i,j=1}^{n} X_{ijk} = 1 \ \forall k \in K \tag{2.3}$$

2.2.3. Node-vehicle relationship

Another common constraint is that a vehicle can access and serve each node on a set of routes only once.

Another common constraint is that each node on a set of routes can only be visited and served once. This is represented by a binary variable called Y_{ij} if any vehicle uses arc (i, j) and 0 if otherwise. The following is how this variable is related to X_{ijk}

$$\sum_{k \in K} X_{ijk} = Y_{ij} \quad \forall i, j \tag{2.4}$$

2.2.4. Capacity

However, due to the fleet's overall characteristics, the capacity constraint must be applied, i.e., the total capacity of the vehicles must not be exceeded. As a result, if a vehicle's capacity is C and demand at nodes j = 1, …, n is d_j , the capacity constraint is given as

$$\sum_{i=2}^{n}\sum_{j=1}^{n} d_j X_{ijk} \leq C \quad \forall k \in K \qquad (2.5)$$

2.2.5. Sub tour elimination

Tours that do not begin and end at the depot are known as subtours. Solutions with cycles employing nodes 2, 3, ..., n are met in the presence of subtours. Because all moves must begin and end at the depot in order to achieve the desired result, it is critical to find a mechanism to eliminate subtour-like cycles. A subtour removing constraint is defined by [32] as

$$\sum_{(i,j) \in N\, XN\, i \neq j} y_{ij} \leq |N| - 1 \; \forall N \in \{2,3, \ldots \ldots, n\} \qquad (2.6)$$

N is any suitable subset of the nodes 2, 3, ..., n, and |N| is the number of nodes in N. y_{ij} has the same meaning as before.

2.3. Basic versions of VRP

As mentioned above different constraints were major contributory factor to the existence of a number of VRP variants. Depending on the factors that change in a time infrastructures facilities and awareness are changing from time to time. There are many types of the Vehicle Routing Problem that require a modification of the definitions given in the previous section. This section gives an overview of the most common simplifications of, and extensions to the VRP. Note that these types do not necessarily exclude each other, combinations of two or more of these types can be made to form more complex types of the VRP.

1. Classical VRP

According to [33], in classical VRP, the clients are known ahead of time. Furthermore, the driving time between customers and the service periods at each customer are calculated. According to Laporte (1992), the classic VRP is as follows: Let $\mathrm{G} = (\mathrm{V}, \mathrm{A})$ be a graph where $\mathrm{V} = \{1 \ldots \mathrm{n}\}$ vertices representing cities and the depot at vertex 1, and A being the set of arcs. A nonnegative distance matrix $C = (C_{ij})$. C_{ij} can be regarded as a travel cost or a trip time in specific situations. When C is symmetrical, a set E of undirected edges can typically be used to replace A. Furthermore, assume that there are m available cars in the depot, were $mL < m < mU$. m is said to be fixed, When $ml = 1\ and\ mU = n - 1$. When m isn't fixed, it's common to attach a

fixed cost f to the use of a vehicle. The VRP entails creating a collection of low-cost vehicle routes that meet the following criteria:

i. Each city in V1 is visited by exactly one vehicle;
ii. All vehicle routes begin and end at the depot;
iii. Some side constraints are met.

2. Capacitated Vehicle Routing Problem (CVRP)

As previously stated, capacity is a constraint that must be applied. A vehicle is only authorized to visit and serve each customer on a set of routes once under a typical CVRP. The vehicle begins and ends its journey at the central depot in order to minimize overall travel costs (distance or time) and not exceed the vehicle's total capacity. This is the fundamental problem of vehicle routing, as VRP may also be viewed as a traveling salesman problem when there is no capacity constraint and only one vehicle (TSP).

3. Vehicle Routing Problem with Time Windows (VRPTW)

The VRPTW is a general characteristic of vehicle routing problems, where customer service can be started within the time window, the earliest time and the latest time that customers will be allowed to receive their products [19]. In this problem, the dimension of time is introduced, and one has to consider both travel time and service time at each customer location. A set of time windows for each customer could be also considered (VRP with Multiple Time Windows). And, these time windows could be flexible depending on some extra costs (VRP with Soft Time Windows)[13].

On VRP-TW problems, time limitations have been investigated, which are relevant in various real-world applications (e.g., express courier carriers, postal services, newspaper delivery, e-commerce, and so on). A Time window is defined as the time interval inside which a vehicle can arrive to a destination to satisfy a request. Two types of time window constraints can be defined

- Hard time windows, which are defined as a strict constraint, in which there is no possibility for a vehicle to arrive to destination after the upper time limit. It is usually also impossible to arrive to destination before the lower time limit, but in some cases this possibility is

considered, allowing the possibility of stopping the vehicle at destination until the lower time limit is reached.

- ❖ Soft time windows, which are defined in the objective function, and represented by an increasing cost penalty if the vehicle arrives to destination outside the time window interval. This representation of the time windows can be applied to many real applications, in which the request can be satisfied, in some conditions, even if time constraint is not strictly respected [34].

4. Vehicle Routing Problem with Pickup and Delivery (VRPPD)

Each customer is assigned two quantities, one for delivery to the customer and the other for pick-up and return to the depot. In addition to the constraint that the total pickup and total delivery on a route cannot exceed the vehicle capacity, it also must ensure that this capacity is not exceeded at any point on the route. One variant of the pickup-and-delivery problem occurs when the pickup demand is not returned to the depot but should be delivered to another customer, as in the transport of people. In some cases, the vehicles must pick up and deliver items to the same customers in one visit (Simultaneous Pickup-and-delivery VRP), as when picking up and dropping off new and returned bottles.

The other essential option is the 1 − M− 1 (“one-to-many-to-one”), which means that all delivery demands start at the depot and all pickup demands end at the depot. All delivery needs can be considered a single commodity, and all pickup demands can be considered a separate commodity. In the literature, this type is referred to as Delivery and Collection. [13].

The Vehicle Routing Problem with Backhauls (VRPB) is a very similar version to the VRPPD, but with the added constraint that all goods must be delivered before any goods may be picked up, because the vehicle is filled "Last-In, First-Out" (LIFO) [28]. The problem can be separated into two separate CVRPs: one for delivery (linehaul) customers and another for pickup (backhaul) customers, with certain vehicles allocated for linehaul and others for backhaul customers [33].

5. **Split Delivery Vehicle Routing Problem (SDVRP)**

The same customer can be served by different vehicles if it will reduce the overall cost. This relaxation of the basic problem is important in those cases where a customer order can be as large

as the capacity of the vehicle [13]. This variant also allows the customer to demand more of a good than the capacity of one vehicle, a situation that is not unrealistic in real life. Finding a solution to an instance of this variant of the VRP is much more complex, and could theoretically make it a continuous optimization problem instead of a discrete optimization problem[28].

6. **Stochastic Vehicle Routing Problem (SVRP)**

The Stochastic Vehicle Routing Problem (SVRP) covers all the variants of the VRP in which one or more properties of the VRP are random. For example, the customer can be present only with a certain probability (think of an ice-cream car looking for children to serve). It can also happen that customers have a certain random demand (for example depending on whether or not the children are hungry). A random factor could also be incorporated in the service time (i.e., is it raining while goods are unloaded, slowing down the process?). Even more dynamic is the variant where the distance matrix is not static and has random factors influencing it. The latter version is interesting, as it has a nowadays more and more important practical application: traffic congestion situations [28].

7. **Asymmetric Capacitated Vehicle Routing Problem (ACVRP)**

The ACVRP is a variant of the CVRP in which the cost between two vertices is not always symmetric, i.e., C_{ij}., need not be equal to C_{ji}. Although, in comparison to the CVRP, this variant is more likely to be encountered in practice (due to the prevalence of one-way streets in most urban areas) [35].

8. **Multi-depot Vehicle Routing Problem (MDVRP)**

Let G represent a collection of depots. The MDVRP is a generalization of the CVRP that takes into account more than one depot. In addition, the vehicle must begin and stop at the same location. In most cases, the number of cars per depot is provided as input data [35].

2.4. Capacitated Vehicle Routing Problem (CVRP)

As previously stated, the basic version of the VRP is the Capacitated VRP (CVRP). The CVRP considers a homogeneous vehicle fleet that transports goods from a single depot to client locations and has a fixed capacity (in terms of weight or number of items) [11].

In the Homogeneous CVRP (or Uniform Fleet CVRP) each vehicle in the fleet has the same capacity Q. The only difference in the formal definition is that a route is considered feasible if the total demand of all customers on a route R does not exceed the vehicle capacity Q: $(\sum_{j=1}^{k} d_j \leq Q)$ where d_j is the demand of customer v_j. Of course, to ensure that vehicles are always big enough, the demand of a customer is never greater than the capacity of a vehicle: $d_j \leq Q$ $(1 \leq j \leq n)$. Also, the total demand of all customers cannot be greater than the total capacity of all vehicles: $(\sum_{j=1}^{n} dj \ \leq \ m * Q\ )$ [28].

[31] emphasizes that their research on CVRP backs up Laporte's (1992) claim that CVRP is defined for a set of homogeneous vehicles $k \ \in K$, an associated service area defined on a graph $G = (N, A)$ such that $= D \ \cup C$, where D is the single depot represented at node 0, C is the set of n customers, $i \ \in C,\ A$ is the set of arcs linking the nodes, i.e., $(i, j) \ \in A$. The cost (distance or time) associated with traveling from node i to j is given as d_{ij} such that $d_{ij} = \ d_{ij}$ (the case of symmetric CVRP). If a vehicle k uses the arc(i, j), the decision variable is $X_{ijk} = 1$, otherwise $X_{ijk} = 0$. The mathematical IP formulation of the problem is

$$\sum_{i\in N} \sum_{j\in N} \sum_{k\in K} d_{ij} X_{ijk} \qquad (2.7)$$

Such that

$$\sum_{i\in N} \sum_{k\in K} X_{ijk} = 1 \ \forall i \ \in C \qquad (2.8)$$

$$\sum_{i\in N} \sum_{k\in K} X_{ijk} = 1 \ \forall j \ \in C \qquad (2.9)$$

$$\sum_{i\in N} \sum_{j\in N} W_i X_{ijk} < \ 1 \ \forall k \ \in K \qquad (2.10)$$

$$\sum_{i\in N} X_{ijk} - \sum_{i\in N} X_{ijk} = 0 \ \forall j \ \in C, k \ \in K \qquad (2.11)$$

$$\sum_{i\in C} X_{0jk} \leq 1 \ \ \forall k \in K \qquad (2.12)$$

$$\sum_{i\in C} X_{i0k} \leq 1 \ \ \forall k \in K \qquad (2.13)$$

$$\sum_{i\in C} \sum_{j\in S} X_{0jk} \leq |S| - 1 \ \ \forall S \subseteq C, |S| \ \geq 2, k \ \in \ K \qquad (2.14)$$

$$X_{ijk} \ \in \ \{0,1\} \ \forall i, j \ \in N, k \ \in K \qquad (2.15)$$

The cost associated with the set of routes covered by vehicle $k \epsilon K$ is minimized using the objective

(2.7) Each consumer must be visited once, according to equations (2.8) and (2.9)

(2.10) represents the capacity constraint, where W_i is the demand at node i and Q is the carrying capacity of each vehicle. The (2.11) of constraints balances the input and outflow of vehicles. Each tour begins and ends at the depot, according to equations (2.12) and

(2.13) The subtour breaking constraint is defined by equation (2.14) and the domain of $X_{ijk}$is defined by (2.15).

2.5. Markov Decision Process

Though the definition of Markov Decision Process (MDP) varies from source to source, its essential meaning remains the same. A Markov decision process, or MDP, is a reinforcement learning problem that satisfies the Markov property. The following five entities are used to define A Markov Decision Process, according to a definition offered by [36]. 1) A set of states $s \in S$, state is a set of tokens that represent every possible state for an agent. 2) A set of actions $a \in A$, A is a set of all actions that can be taken. The set of actions that can be executed while in state S is defined by A(s). 3) The transition probability $\delta_a(S, S')$, which expresses the probability of being in state S and taking action 'a'. 4) The reward function $R_a(S, S')$, indicates the reward for being in a state S, taking an action ‘a’ and ending up in a state S’, i.e., how good it is for the agent to be in a given state (or how good it is to perform a given action in a given state). 5) The policy $\pi(s)$, which is a mapping of states to actions. It indicates the action ‘a’ to be taken while in state S. The policy π can be viewed on as the set of rules that are utilized when selecting the next action.

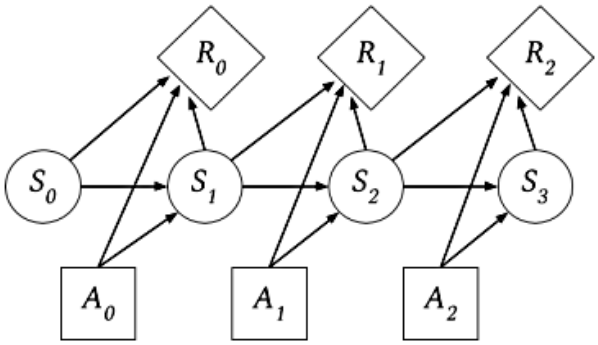


Figure 2.2: Visualization of a Markov Decision Process, where S represents the states, R the rewards and A the actions. The subscripted integers denote the timesteps of the process. Illustration based on [37].

In the case of VRP, the state indicates the location and demand of the customers and the sequence of the customers visited. The actions are the selection of what customer to visit next. Depending on problem specific constraints, the reward could represent the cost or time associated with the vehicle moving between customers.

Finding a policy that achieves a lot of reward, i.e., the summed rewards of all state-action pairs until the process reaches a terminal state, is approximately what solving an MDP task means. The policy that maximizes this reward is symbolized as π^*, and it is regarded the optimal solution to the MDP. There is always at least one policy that is better than or comparable to all others. This is an optimal policy [38].

2.6. Application of VRP

With the integration of the logistics industry, information technology, and the globalization trend, the logistics distribution system as a whole is assuming a greater and greater importance in the system. Because the transportation system is one of the most significant subsystems in the distribution system, accounting for roughly half of all logistics costs, it is the first to cut logistics distribution costs. Among them, whether the transportation route directly affects the distribution speed, cost and benefit, especially the determination of distribution line is a complicated system engineering. Select appropriate vehicle routing, can speed up the response to customer demand, improve service quality, enhance customer satisfaction to the logistics link, reduce service provider operating costs [24].

Because of its critical significance in the development of distribution systems and logistics in various sectors, the VRP has piqued the interest of numerous scholars in recent decades. School bus routing, street cleaning, municipal solid waste collection, dial-a-ride systems, routing of salespeople, heating oil distribution, milk delivery, mail pickup and delivery, maintenance unit routing, transportation of disabled people, currency delivery to ATM machines, prisoner transportation between jails and courthouses, beverage delivery to bars and restaurants, and so on are just some of the real-world applications of VRP. It has a lot of practical applications. They can

be either environmental (lower emissions from reduced gasoline consumption) or financial (reduced transportation costs, lower consumer prices, and optimal business resource usage) [39].

Waste Collection VRP: Waste collection in metropolitan areas is one of the most significant municipal services, and it must be planned and done with utmost caution and efficiency due to special environmental and recycling regulations. E-waste is one of the most important aspects of the Waste Collection VRP (WC-VRP), which involves the collection of waste electrical and electronic equipment and relies on efficient container loading and route optimization [40].

Transportation of hazardous materials: Due to the raising demand for this type of transportation, the problem gained special interest in recent years. In such expeditions, a potential threat to public (by means of leakage, explosion, poisoning or other accidents with serious consequences) is considered a major issue. Hence, the key factor is the minimization of the probability of an accident and the minimization of its potential consequences [40].

Bike sharing systems: An interesting usage of the VRP framework in a large-scale inventory rebalancing problem is in the context of a bike sharing system. Such systems are rapidly developing all over the world. In 2015 the estimated number of bike networks in major cities exceeded 1000 and in another few hundred BSSs were under construction or seriously planned [40].

2.7. Machine Learning

The words, Artificial Intelligence and Machine Learning are not new. They have been researched, utilized, applied and re-invented by computer scientists, engineers, researchers, students and industry professionals for more than 60 years. The mathematical foundation of machine learning lies in algebra, statistics, and probability. Serious development of Machine Learning and Artificial Intelligence began in 1950’s and 1960’s with the contributions of researchers like Alan Turing, John McCarthy, Arthur Samuels, Alan Newell and Frank Rosenblatt. Samuel proposed the first working machine learning model on Optimizing Checkers Program. Rosenblatt created Perceptron, a popular machine learning algorithm based on biological neurons which laid the foundation of Artificial Neural Network [41].

Machine learning is a branch of artificial intelligence that tries to use intelligent software to enable computers to do skilled tasks. The backbone of intelligent software that is utilized to generate machine intelligence is statistical learning methods. Because machine learning algorithms require data in order to learn, the field must be linked to database. Similarly, words like Knowledge Discovery from Data, data mining, and pattern recognition are common [42].

[43] Machine Learning (ML) is not a new concept. ML is closely related to Artificial Intelligence (AI). AI becomes feasible via ML. Through ML, computer systems learn to perform tasks such as classification, clustering, predictions, pattern recognition, etc. To archive the learning process, systems are trained using various algorithms and statistical models to analyse sample data. The sample data are usually characterized by measurable characteristics called features and an ML algorithm attempts to find a correlation between the features and some output values called labels. Then, the information obtained during the training phase is used to identify patterns or make decisions based on new data. ML is ideal for problems such as regression, classification, clustering, and association rules determination. Depending on the learning style, ML algorithms can be grouped into four categories:

Supervised Learning: Supervised learning uses algorithms like Linear Regression and Random Forest to solve issues involving regression, such as weather predicting, assessing life experience, and population growth prediction. Additionally, supervised learning uses algorithms like Support Vector Machines, Nearest Neighbour, Random Forest, and others to solve classification issues including digit recognition, audio recognition, diagnostics, and identity fraud detection. In supervised learning, there are two stages. The phases of training and testing. The data sets utilized in the training phase must have labels that are known. The algorithms try to anticipate the output values of the testing data by learning the link between input values and labels. [43].

[44] According to the definition, supervised learning is a machine learning technique that is applied to systems where the right response is delivered by a competent expert and is designed for huge amounts of input data (training sets). A system is taught with data that has been labelled in supervised learning. Each data point is divided into one or more categories by the labels. The system then learns how this training data is organized and uses that knowledge to forecast which categories new output data should be classified into. The outputs of a finished supervised learning process should be near enough to be useful for all given input sets.

Unsupervised Learning: Unsupervised learning deals with problems involving dimensionality reduction used for big data visualisation, feature elicitation, or the discovery of hidden structures. In addition, supervised learning is utilized to solve problems like recommendation systems, consumer segmentation, and targeted marketing. In contrast to supervised learning, no labels are available in this type. This category of algorithms aims to find patterns in testing data, cluster it, or forecast future values [43].

Semi-supervised Learning: This is a hybrid of the two previous categories. Unlabelled and labelled data are both used. It works similarly to unsupervised learning, but with the added benefit of a part of labelled data [43].

Reinforcement Learning: The algorithms in this learning style attempt to forecast the output of a problem using a set of tuning parameters. The calculated output is then used as an input parameter to produce further outputs until the ideal output is determined. This learning style is used by Artificial Neural Networks (ANN) and Deep Learning, which will be discussed later. Reinforcement learning is mostly employed in AI games, skill acquisition, robot navigation, and real-time decision-making [43].

Reinforcement learning, according to [44], addresses assignments in which some data contains labelled training sets but other data does not. Instead of training datasets that indicate the right output for a given input, reinforcement learning assumes that training datasets only tell whether an action is correct or not. If an action is incorrect, the difficulty of determining the correct action persists.

There are two important aspects to consider when employing machine learning techniques: how computationally intensive and how fast an approach is. The most appropriate machine learning algorithm is chosen based on the application type. If real-time analysis is required, the algorithm chosen should be able to keep up with the changes in the data [43].

2.8. Reinforcement Learning

The Studies over the past two decades have provided important information on Reinforcement Learning (RL). RL is one of the heuristic methods used to solve MDPs within a field of machine learning. In RL, the decision-making entity is called the agent. The agent interacts with the MDP that within RL is known as the environment. The general RL problem is formalized as a discrete

time stochastic control process where an agent interacts with its environment in the following way: the agent starts, in a given state within its environment $S_0 \in S$, by gathering an initial observation $\omega_0 \in \Omega$. At each time step t, the agent has to take an action $a_t \in A$. As illustrated in Figure *2.3*, it follows three consequences: (i) the agent obtains a reward $r_t \in R$, (ii) the state transitions to $s_{t+1} \in S$, and (iii) the agent obtains an observation $\omega_{t+1} \in \Omega$ [38].

The interaction consists of the agent selecting an action a_t in state s_t according to the policy π, then it observes the feedback from the environment. The feedback consists of the next state S_{t+1} and the associated reward r_t of the transition from s_t to S_{t+1}. [45] One episode consists of this interaction, visualized in Figure 2.3, repeated until a terminal state is reached.

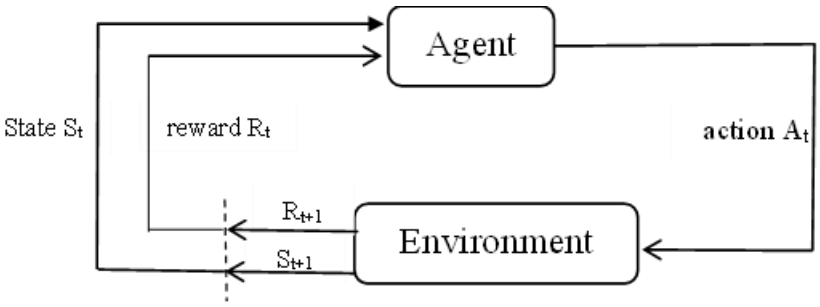


Figure 2.3: Visualization of the reinforcement learning framework. The subscripts t and t+1 denotes the timesteps of the process. Illustration based on [38].

The goal of an agent is to take actions that maximizes the total rewards. This sum of rewards is in RL is known as the return which indicates how good and bad is an event. Basically, the agent tries to randomly exploring the environment which is done by having the agent selecting random actions in order to obtain π^*. Gradually, based on the accumulated previous experience, the agent can learn how to select better and better actions by observing the outcomes of the exploring [46].

There are two categories of methods regarding how to store the previous experience in order to obtain π^*, these are known as value-based and policy-based methods. In the value based RL, we first calculate the value function for each state as stated in (2.16 and then we use these values to evaluate the policy [47].

$$V^{\pi}(S) = E\left[\sum_{t}^{r} r_t\right] \forall_s \in S \quad (2.16)$$

The goal is to approximate the optimal value function $V^*(s)$, i.e., the function that has the highest value for all states, described in (2.17).

$$V^*(S) = max_{\pi} V^{\pi}(s) \quad \forall_s \in S \quad (2.17)$$

By using $V^*(s)$, an estimate of the optimal policy π^* can be derived as described in (2.18.

$$\pi^* = arg\ max_{\pi} V^{\pi}(s) \quad \forall_s \in S \quad (2.18)$$

Policy based method search for an optimal policy directly with no value function; hence the goal is to find the optimal policy directly. It is an optimization problem where a parameterized policy π^* is updated to maximize the return. By defining the policy as a set of parameters θ, a differentiable function J_θ can be defined as formulated in (2.19) [46].

$$J_\theta = E_{\pi\theta}[r(\tau)] \quad (2.19)$$

By then exploring the environment using the current policy π_θ and observing rewards, the gradient with respect to θ that maximizes J_θ, can be calculated. By iteratively updating the parameters θ in the direction of the gradient, i.e., updating the policy directly, policies close to π^* can be obtained.

Both value-based and policy-based methods do not include assumptions about the transition function (the model of the environment), and hence they are called model-free RL [47]. Moreover, they have their own separate advantages. In value-based approach, a small change in value can cause an action to be or not be selected, i.e., they have good performance with regards to sample efficiency. But this is not the case in Policy-based methods [48]. This is why methods called actor-critic methods have emerged by combing the two methods. In this technique, both value and policy are learned.

2.9. Actor-critic methods

Actor-critic approaches, as highlighted by [38], include a separate memory structure to clearly reflect the policy separate from the value function. The policy structure is referred to as the actor because it is used to select actions, and the estimated value function is referred to as the critic because it evaluates the actor's actions. Learning is always on-policy: the critic must learn about

and analyse whatever policy the actor is now pursuing, and compare it to the anticipated expected return. The advantage is the result of this. As shown in Figure 2.4, this scalar signal is the critic's only output and drives all learning in both actor and critic.

Actor-Critic architectures constitute a hybrid approach between value-based and policy-gradient methods by computing both the policy (the actor) and a value function (the critic). where the critic calculates the value function while the actor updates the policy using the values calculated by the critic. The actor (policy) receives a state from the environment and chooses an action to perform. At the same time, the critic (value function) receives the state and reward resulting from the previous interaction. The critic uses the error calculated from this information to update itself and the actor [47].

It is necessary to understand that the advantage metric is utilized through both the actor and the critic in their learning processes. For the actor, the advantage function provides a measure of comparison for each action to the expected return at the state. At the same time, the advantage is used by the critic to consider how well it estimated the expected return. Note that in early levels of the learning process the critic estimates are simply random, i.e., more or less useless. But simply after a few updates, the critic begins to learn the value function. This offers a long-term overall performance increase of the total model, outweighing the slow begin [45].

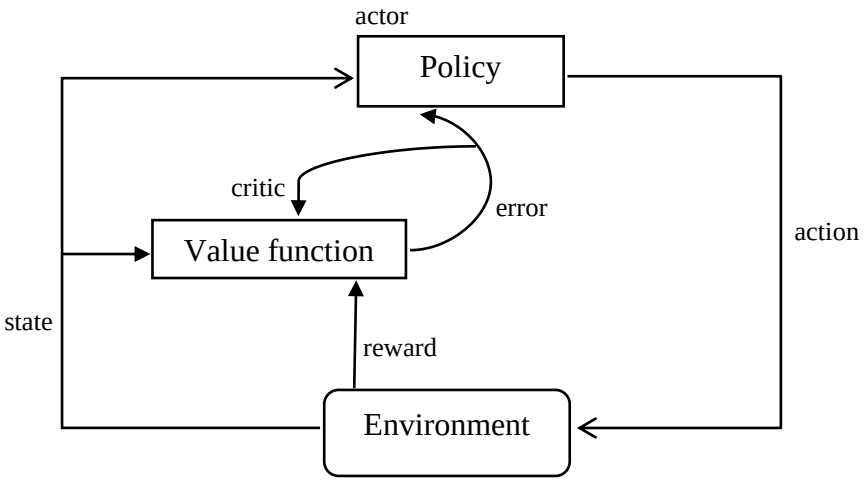


Figure 2.4: Overview of the actor critic algorithm, from [38]

Actor critic methods intend to combine value functions with a parametrized policy. But how can we store the parameters in the actor and critic components? If the action- and state-spaces are very small, a simple table would suffice. However, this is seldom the case, seeing that one of the primary troubles concerning combinatorial optimization problems are that the units of possible actions and state are enormous. During later years Artificial Neural Networks (ANNs) have, due to their inherent structure, been emerging as a popular tool used to store these parameters [36]. This has resulted in a new paradigm inside RL, known as Deep Reinforcement Learning (Deep RL).

2.10. Deep Reinforcement Learning

This section provides the fundamental components and concepts of ANNs to gain perception on this study. Note that only related concepts deemed highly relevant in order to understand this report are presented.

2.10.1. Artificial Neural Networks

An artificial neural network (ANN) is a network of interconnected units with some of the qualities of neurons, which are the primary components of nervous systems. The units (the circles in Figure *2.5*) are typically semi-linear, which means they compute a weighted sum of their input signals before applying a nonlinear function called the activation function to the result to produce the unit's output, or activation. The neurons can be joined in both a sequential and parallel fashion, as illustrated in Figure *2.5*, implying that there are no loops in the network, that is, no paths within the network by which a unit's output can impact its input.A weight roughly corresponds to connection between the neurons themselves [38]. ANN works on three layers: input layer, output layer and “hidden layers”: layers that are neither input nor output layers. This type of network has weighted interconnections and learns by adjusting the weights of interconnections in order to perform parallel distributed processing. The Perceptron learning algorithm, Back-propagation algorithm, Hopfield Networks, Radial Basis Function Network (RBFN) are some popular algorithms. ANNs with multiple hidden layers are known as Deep Neural Networks (DNN) [41].

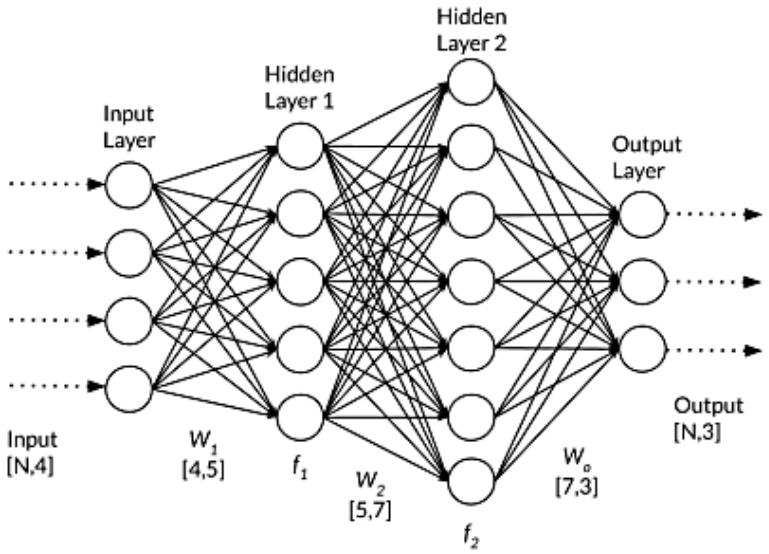


Figure 2.5: Visualization of an Artificial Neural Network. W represents the sets of weights between each layer. The integers within the brackets denotes the size of W. Hidden Layers are represented by f. The subscripts denote the integer or variable of the corresponding layer. Illustration based on [38].

The network weights are adjustable and can be adjusted to achieve the best solution. In the context of RL, the optimal solution is the weight that maximizes the expected return defined as a parameter of either of the two functions described in Section 2.8. During training, the input is repeatedly supplied to the network, observing the rewards received and adjusting the parameters by propagating the reward signal in the opposite direction over the network. This process can be performed in a variety of ways, and the methodology used in this study is defined in Section 2.9.2 as well. Often, these updates are done in batches, i.e., The average error of multiple inputs (batch) affects the adjustment. When the learning process is complete, the knowledge needed to solve the problem is stored in the weight configuration that generated the smallest error [12].

2.10.2. Stochastic Gradient Descent

One major issue in ANN to learn by updating its parameters, there must be a way on how to update them. Therefore, Stochastic Gradient Descent (SGD) method is utilised, it performs a parameter update for each training in the direction of the gradient in order to minimize some objective function [49].While a variety of definitions of the term objective function have been suggested, but in general they are utilized to calculate some performance metric describing the error of the network outputs or they can aim to maximize expected reward. In the case of RL, when the optimal

value is unknown this becomes complicated since an error can be hard to define. [38] As explained in section 2.9.1, it is clear that the objective functions are used for the actor and critic networks respectively presented.

There are three types of gradient descent, each with a different amount of data used to compute the objective function's gradient. We make a trade-off between the precision of the parameter update and the time it takes to complete an update, depending on the amount of data. As a result, SGD updates the entire dataset one at a time. As a result, it is usually faster, resulting in speedier calculations and the ability to learn online. SGD makes frequent, high-variance updates, causing the objective function to vary a lot. The magnitude of the steps we take to reach a (local) minimum is determined by our learning rate. To put it another way, we follow the slope of the surface produced by the objective function downhill until we reach a valley [49].

2.10.3. Actor-critic methods using Deep RL

This research revolves around an actor-critical method of storing policy and value function parameters separately using two neural networks. The actor network stores the parameterized policy J_θ, i.e., the expected return of following the current policy π_θ, denoted in (2.20). Given the Policy Gradient Theorem [36], the derivative of J with respect to θ, can be expressed as in (2.21 in the actor-critic setting.

$$J_\theta = E_{\pi\theta}[r(\tau)] \tag{2.20}$$

$$\nabla \mathrm{E}_{\pi\theta}[r(T)] = \mathrm{E}_{\pi\theta}\left[\left(r(T) - V(S_0{:}\,\emptyset)\right)\left(\sum_t^T \nabla_\theta \log \pi_{\theta(a_t \mid s_t)}\right)\right] \tag{2.21}$$

By analyzing the contents of (2.21 the second term in the product of the right-hand side denotes the gradient of the log probabilities of the policy π_θ. To provide some intuition, this term essentially provides a value of how certain the actor is in its selection process. If the policy is uncertain, the sum of the log probabilities generated by taking a set of actions will be a large negative value. This will yield a gradient with large magnitude. However, if the model learns that some actions are better than others in different states, it will increase the probabilities of selecting these actions and the sum will approach zero. This will produce smaller gradients, thus not updating the policy as much.

The first term of the product denotes the advantage, i.e., the accumulated reward of a given trajectory subtracted by the output of the critic. As a reminder, the critic outputs consist of the estimate of the value function given the starting state S_0. This term will naturally decrease as the routes are being optimized. The critic estimates are used to reduce the variance of the actor gradient updates, proven to yield a faster learning process and good convergence properties for the model [9]. The intuition behind the product in (2.21) is therefore that it provides a model performance measurement (2nd term) that also includes the stochasticity of the model (1st term).

$$\theta \leftarrow \theta + \alpha_1 V_\theta J \quad (2.22)$$

After using (2.21 to calculate the gradients of the policy π_θ, the actor parameters are updated using SGD in the direction of the gradient. This operation is formulated in (2.22, where α_1 & denotes the actor learning rate [45].

$$\emptyset \leftarrow \emptyset + \alpha_2 \nabla_\emptyset (r(T) - V(S_0{:}\emptyset))^2 \quad (2.23)$$

The parameters of the critic network $\emptyset$ are also updated using SGD. The objective function is defined as the mean squared error of its prediction compared to the actual rewards, i.e., the square of the advantage, described in (2.23, (2.23. This in order to improve the estimates produced by the critic. In (2.23, α_2 denotes the critic learning rate [45].

2.11. Explanations of the proposed VRP model

This section will provide the reader with a thorough description of the method applied to the VRP proposed by Nazari et.al in their article "Reinforcement Learning for Solving the Vehicle Routing Problem" [5]. This since it constitutes the foundation on which this study is based on. Later, in section 3.2, are the implementation changes made to adapt this method to work on the pharmaceutical delivery, presented. Throughout section 2.10 [5] is utilized as a first-hand source.

2.11.1. Description of VRP inputs

The goal of the problem is to find the shortest route that visits each customer once in order to deliver pharmaceutical products, starting and finishing from the same node, called a depot, while using a fleet of vehicles. The states or customer is partially represented by a set of inputs $X = \langle x^i, i = 1, \dots, M \rangle$, Where M is the number of customers that is going to be served. Each input x_i is

tuple consisting of two components: customer i's location and demand, called static and dynamic inputs, $x^i = (s^i, d^i)$. Static inputs are defined as two-dimensional Euclidean coordinates which is the latitude and longitude property of customer location, and dynamic inputs are defined as integers. When delivering packages, the dynamic input of visiting customers is reduced. Therefore, the inputs x^i is more formally defined as a sequence of tuples $\{x_t^i = (s_t^i, d_t^i), t = 0, \ldots, T\}$. The location of customer remains constant during the episode, since location is static input. The set of all inputs X at time t indicated as X_t.

The main task of actor is to determine which orders should be serviced by each route (truck) and in what sequence the orders should be visited. An order can be a delivery to a customer. The actions are indicated as y_t. The actor begins by selecting the primary customer to visit $Y_0 \in X_0$ and after that along these lines chooses actions. Actions are stored in a s sequence $Y = \{Y_t\ t = 0, \ldots, T\}$, where T indicates the time step when the demand from all customers is zero, such that the termination condition is met. provide a complete representation of the state. Rewards are maximized, so this VRP implementation defines rewards as the negative Euclidean distance between customers.

2.11.2. Action Selection Process

The action selection process is done by the actor started with selecting the first customer to visit and then subsequently selects an action. It's a stochastic process that involves selecting samples from a probability distribution of available actions, such as customers with demand remaining. This distribution is calculated using the actor's parametrized policy and the current state. The goal of the actor is to optimize its parameters in a way that minimizes the objective function of the Policy Gradient method, described in section 2.5.3. i.e., to produce the shortest sequence Y, given X_0. It's worth noting that, because the vehicle's weight is limited, it'll be compelled to return to the depot if it's empty. This is accomplished by setting the probabilities of all actions, alongside selecting the depot, to zero manually.

2.11.3. Training the Proposed Model

The networks are trained using the REINFORCE method, which is a policy gradient technique. This is the basic policy gradient algorithm upon which practically all advanced policy gradient algorithms are built. Policy gradient methods are reinforcement learning approaches that use

gradient descent to optimize parametrized policies in terms of expected return (long-term cumulative reward). To employ this strategy, we must first parameterize the stochastic policy π with parameters θ. Actor and critic networks are the two sorts of networks provided by the algorithms. a critic network that evaluates the reward for any problem instance from a given state and an actor network that predicts a probability distribution over the next action at any given decision step. In the pseudocode below, θ e indicates the parameters of the actor network and $\emptyset$ indicates the critic parameters.

Algorithm 1: REINFORCE Algorithm

1: initialize the actor network with random weights θ and critic network with random weights $\emptyset$

2: for iteration $= 1, 2, \ldots$ do

3: reset gradients:$d\theta \leftarrow 0, d\emptyset \leftarrow 0$

4: sample N instances according to Φ_M

5: for n = 1, ⋯, N do

6: initialize step counter $t \leftarrow 0$

7: repeat

8: choose $Y_t^n + 1$ according to the distribution $P(Y_t^n + 1 \,|Y_t^n, X_t^n)$

9: observe new state $X_t^n + 1$

10: $t \leftarrow t + 1$

11: until termination condition is satisfied

12: compute reward $R_n = R(Y^n, X_0^n)$

13: end for

14: $d\theta \leftarrow \frac{1}{N}\sum_{n=1}^{N}(R^n - V(X_0^n;\, \emptyset))\nabla_\theta \log P(Y^n|X_0^n)$

15: $d\emptyset \leftarrow \frac{1}{N}\sum_{n=1}^{N}\nabla_\emptyset(R^n - V\,(X_0^n;\emptyset))^2$

16: update θ using $d\theta$ and $\emptyset$ using $d\emptyset$.

17: end for

Pseudocode of the REINFORCE algorithm [7].

Intuitively, REINFORCE algorithm begins with an initial guess for the value of policy's weights that maximizes the expected return. Then, as the algorithm consists of sampling N instances of X,

it iterates over the set of instances, each X^n is then processed by the actor network until it considers that it is eventually reached the maximum expected return. Based on the reward function the resulting sequences Y^n are then evaluated, also for each Y^n total Euclidean distance is computed. Then, the critic network calculates the expected return by using the starting instances, X_0^n. According to step 14 and 15 from above Pseudo code the updates of θ and $\emptyset$ are computed by using the outputs from the actor, critic and reward function. Finally, the parameters of actor and critic network are updated.

After the training is completed, the actor network needs to have optimized its parameters and may be applied as a near to optimal policy while planning future routes for the vehicle. i.e., given new instances of X, a simple forward pass through the actor network will generate a close to optimal sequence Y, describing in which order to visit the customers. The reader is referred to [7] for details such as used hyper parameters, network sizes and optimizers used.

Table 2.1: Different ML algorithms pros vs cons

Algorithm	Parameters	Basic idea	Pros	Cons
K-means	Number of clusters	Distances between points	Suitable for a dataset having an even number of cluster size, the flat geometry and not too many clusters	(i) Clustering results may differ for a different initial value of K (being the only hyperparameter in K-means)
DBSCAN	Neighborhood size	Distances between nearest points	Good for uneven cluster size with non-flat geometry, where there are not too many clusters;	(i) Results low-quality (LQ) clustering as it struggles at separating nearby clusters and (ii) quadratic computational complexity
OPTICS (an extension of DBSCAN)	Minimum cluster membership	Distances between points	Suitable for an uneven cluster size with non-flat geometry and variable cluster density;	(i) Results LQ clustering as it struggles at separating nearby clusters and (ii) quadratic computational complexity

2.12. Related works

In this section, the research area related to our works that have been done on solving VRP was realized and also the gap relevant to our work was discussed. Since the first VRP presented by Dantzig and Ramser in 1959 [14], many algorithms have been proposed for solving either the classical VRP or its variants. Exact algorithms were proposed as well as heuristics [33].

Gladkov et al. [18] Proposes that an integrated approach for solving transport-type problems. So, they propose to use modified bio inspired methods of search for optimal decisions, based on complex criteria adapted and give an approximate sequence of the algorithm for solving the VRP. This approach allows them to construct algorithms for solving transport-type problems with local optima in polynomial time. But this work does not exceed the limits of the polynomial dependence and is quadratic in nature. In addition, the data size is relatively small and constant when compared with other researches and the study was not state any recommendation and future work.

The study has been done by Xu et al. [21] discover Enhanced Ant Colony Optimization by reviewing previous papers about Ant Colony Optimization. In order to gain better solutions, K - means, crossover, and 2-Opt where applied to enhance ACO. The study modifies the version of the Best Cost Route Crossover. The algorithm was implemented using MATLAB (version: R2010b) language and open-source dataset. They perform statistical analysis using a paired t-test to investigate whether there are statistically significant differences between ACO, K-ACO, E-ACO, DVRP-GADAPSO, and VNS according to the solution quality. E-ACO is statistically significant from it with 0.983 and mean difference of -5.45. The analysis indicates that the E-ACO performs as well as other algorithms which are the most effective approaches recently proposed in the study. The data set the study used were open-source data. But this work also does not use any data preparation method.

Martinson et al. [8] Showed that how to optimize route in logistics distribution based on particle swarm optimization. They used data set from (Songyi Wang, 2017) and (Shahrzad Amini, 2010) which considers a fixed distribution center distributing goods to 100 customers, a fixed vehicle cost of 200Yuan and fuel cost of 3Yuan/km per mileage in the transportation process and for the Parameters. They use MATLAB R2014a to implement the PSO. According to the results, parameters such as the population size of the algorithm and number of iterations had effect on the

optimization result. In the experiment, it was showed that when the population size and the number of iterations increased, optimal and operational results can be obtained faster and better. But like other evolutionary optimization algorithms, it is easy to fall into the local optimum. In this work there is no any data preprocessing method and no comparison with other model.

The other study done by Catay et al. [50] aims to develop a set of vehicle routes originating and terminating at the depot. In the study the ACO approach was applied to the well-known Vehicle Routing Problem with Pickups and Deliveries (VRPPD). The proposed algorithm is coded using C++. The study data set consists of 63 instances with the number of customers varying from 25 to 150. The performance of the algorithm for VRPSPD is tested using two well-known benchmark problem sets from the literature. For each VRP instance generated a VRPSPD problem by splitting the original demand between demand and pickup loads. Another instance was obtained by exchanging these demand and pickup loads of every other customer. The second benchmark problem was improvements on the best-so-far. The experimental analysis clearly show that promising results compared to the results published in the literature. Furthermore, improvements on some of the best-so-far solutions are obtained as well. The research was not stated the sampling techniques of the data and the source of the data. On top of this, the research work was not stated the techniques of data preparation. According to the result of the study the approach was used to solve such problem scores best result. The algorithm has a good effect in solving the VRP problem, but it is easy to fall into the local optimum problem if the calculation time is long.

Singhtaun et al. [51] has developed a delivery planning program to determine vehicle usage and to provide an optimal delivery route for a fleet of vehicles in the case study company. After program trial, the program facilitates the planners to make decisions on delivery planning and the delivery cost is decreased by 8.15%. The new mathematical model applied in the program can represent the real problems. The branch and bound algorithm in Open Solver can solve the model, which is a MIP, in an acceptable computing time. Whereas, [20] formulated a green-MDVRP problem. The carbon emission of the logistic network is added as a cost function to the routing costs for accounting the environmental impact of the supply chain. Two soft computing search procedures are developed to solve the discrete optimization problem. An ACO based heuristic and a hybrid heuristic combining ACO and VNS are used to solve the problem near optimally. The algorithms are tested on randomly generated problem instances. The hybridization provides

significant improvement in the solutions. Based on the computational study, the results are found to be consistent over the test data. The computational results in this work provide guidelines for environmentally conscious and responsible route selection decisions. This work does not mention the source of the data and also sampling techniques of the population size.

Moryadee et al. [19] were used two algorithms, which were the Genetic Algorithm (GA) and Tabu Search (TS), as the solution methods. For the Genetic Algorithm, the initial population size, crossover operator, and mutation operator were set at a specific value to generate the best solution as was mentioned in the literature. The Tabu Search (TS) was used in OptQuest. Both methods had a stopping condition as a certain number of trials. Finally, the results of both methods were compared in terms of computation time and solution. Moreover, the best solution was compared with the current solution of the company in order to show the improvement. Tabu search algorithm is used to solve the VRP problem, although it can ensure the exploration of different effective search methods, but it is not easy to implement in practice because it involves complex neighborhood transformation and solving strategies.

Nazari et al. [5] propose a model that combines an attention mechanism with a recurrent neural network (RNN) decoder. The static element embeddings are fed into the RNN decoder at each time step, and the RNN output and dynamic element embeddings are fed into an attention mechanism, which produces a distribution across the possible inputs that can be picked at the next decision point. The suggested framework appeals because it employs a self-driven learning approach that only requires reward calculation based on created outputs, as long as the reward is observed and the feasibility of a generated sequence is verified. As a result, the desired meta-algorithm can be learned.

Table 2.2: Summary of related work

Author	Title	Method	Algorithm	Result	Gap
Gladkov, S. N. Scheglov, and N. v.	The application of bioinspired methods for solving	modified bio inspired methods of search for	bionic algorithms	the time complexity of the developed bionic algorithms does	Unfit for highly complex transportation system.

Gladkova [18]	vehicle routing problems	optimal decisions		not exceed the limits of the polynomial dependence and is quadratic in nature	
H. Xu, P. Pu, and F. Duan, [21]	Dynamic Vehicle Routing Problems with Enhanced Ant Colony Optimization	K - means, crossover, and 2-Opt where applied to enhance ACO	Ant Colony Optimization	The analysis indicates that the E-ACO performs as well as other algorithms which are the most effective approaches recently proposed in the study	More uncertain time to convergence
A. Martinson and X. Qiang, [8]	Route Optimization in logistics distribution based on Particle Swarm Optimization	use MATLAB R2014a to implement the PSO	Particle Swarm Optimization (PSO)	As population size and number of iterations increased, optimal and operational results can be obtained faster and better.	No comparison with other model. Exponentially increasing calculation
B. Çatay, [50]	Ant Colony Optimization and Its Application to	ACO approach was applied to the well-	Ant Colony Optimization (ACO)	According to the result of the study the approach was	Difficult theoretical analysis.

	the Vehicle Routing Problem with Pickups and Deliveries	known Vehicle Routing Problem with Pickups and Deliveries (VRPPD		used to solve such problem scores best result	
C. Singhtaun and S. Tapradub, [51]	Modeling and Solving Heterogeneous Fleet Vehicle Routing Problems in Draft Beer Delivery	ACO based heuristic and a hybrid heuristic	ACO based heuristic and a hybrid heuristic combining ACO and VNS are used	results in this work provide guidelines for environmentally conscious and responsible route selection decisions	Probability distribution can change for each iteration.
C. Moryadee and W. A. and M. R. Shaharudin [19],	Congestion and Pollution , Vehicle Routing Problem of a Logistics Provider in Thailand	Genetic Algorithm (GA) and Tabu Search (TS	the results of both methods were compared in terms of computation time and solution. Moreover, the best solution was compared with the current solution of	Tabu search algorithm is used to solve the VRP problem, although it can ensure the exploration of different effective search methods	Not easy to implement in practice because it involves complex neighborhood transformation and solving strategies. Genetic algorithms do not scale well with complexity.

			the company in order to show the improvement.		
M. Nazari, A. Oroojlooy, L. v Snyder, and M. Tak, [5]	Deep Reinforcement Learning for Solving the Vehicle Routing Problem	RNN	end-to-end framework for solving the Vehicle Routing Problem (VRP) using reinforcement learning	On capacitated VRP, their approach outperforms classical heuristics and Google's OR-Tools on medium-sized instances in solution quality with comparable computation time	Hardware dependency. Difficulty of showing the problem to the network.

CHAPTER THREE

3. RESEARCH METHODLOGY

This chapter describes common research methods for creating datasets, achieving research goals, and exploring techniques for answering research questions. This chapter describes and justifies the methods used to conduct research on building ML models to solve vehicle routing problems in pharmaceuticals delivery. The study followed Design Science Research Design and applied a variety of methodologies starting with problem identification, defining the objectives of the solution, design and development, testing and evaluation. This is the procedure used to evaluate the two approaches and models using EPSS's imported custom datasets. This chapter also covers a description of the software tools used to apply different algorithms to data preprocessing and classification.

3.1. Methodology

The experimental research methodology was used in order to design this research work. In general, EPSS distribution geographical area data were first collected from EPSS Adama to feature vectors organized by each various size VRP instances. Then, RL models were trained with clustered and un-clustered VRP instances. Finally, the results were statistically evaluated to analyze the effects of the two results on the performances of the two approaches.

3.2. Literature Review

A detailed literature review is performed to understand the vehicle routing problem and its solution. This literature review is proposed to achieve the research goals related to vehicle routing problem, identify the appropriate algorithms, and identify the appropriate issues in the case study. Continuous literature reviews are conducted to gather the information needed at various stages of the proposed study.

The sources of this literature review are books, journal articles, conference articles, web articles, and research articles published in various online databases (Google Scholar, PubMed, arxiv, researchgate). Gaps in previous solutions will be identified during the revision of this article in

order to use as inputs to proposed solution. This is important content in which all possible references and journals related to the study are researched and analyzed.

3.3. Materials and Tools

3.3.1. Software Tools

This section gives an overview of the tools used in the implementation of solution. It is included in the study so that others can reproduce the study using the same. The implementation of the experiment used in the research is written in a programming language called Python. It also uses some supplemental libraries, such as: Pandas, Numpy, ScikitLearn, Keras. The most commonly used libraries are described below.

Anaconda: Used to implement models, free and open-source Image processing, data science, machine learning and related applications aimed at adding and simplifying package management and deployment. It contains various IDEs such as Qt, Console, Jupyter Notebook, Visual Studio, and Spyder in order to write the coding part. Here in this study a Jupyter notebook is used for implementing the coding part. It is easy to use and can be run in a web browser.

Numpy: which stands for Numerical Python, is a collection of multidimensional array objects and functions for manipulating them. NumPy allows you to do math and logical operations on arrays. NumPy is the foundation library for scientific computing in Python because it has data structures and high-performance functions that the Python basic package lacks [52].

Scikit-learn: is a general-purpose Python machine learning toolbox that is extensively used and trusted. It includes tools for model selection and evaluation, data transformation, data loading, and model persistence, as well as a wide range of common supervised and unsupervised machine learning algorithms. Classification, grouping, prediction, and other typical tasks can all be accomplished using these models [53].

Torch: The torch package contains data structures for multi-dimensional tensors and defines mathematical operations over these tensors. Additionally, it provides many utilities for efficient serializing of Tensors and arbitrary types, and other useful utilities [54].

Matplotlib: This package is a Python library that is currently popular for producing plot and 2D data visualization. This is the most suitable library for this purpose, as data analysis requires a visualization tool [52].

Pandas: This is a Python package that provides expressive data structures that work with both relational and labeled data. This is an open source Python library that contains data structures and data manipulation tools designed to make cleaning and analyzing data in Python fast and easy [53].

3.3.2. Hardware Tools

To implement the machine learning model with the selected software tools this study uses the Lenovo V50t Tower Desktop with the following specifications: CPU Intel(R) Core (TM) i5- 10400 CPU @ 2.90GHz processor, 8GB RAM.

3.4. Methods of Data Understanding

3.4.1. Source of Data

The data was obtained from the Ethiopian Pharmaceutical Supply Services's (EPSS) health facility catchment data, which was integrated using Google Maps with the help of JSI [27]. There are 420 health facilities in the EPSS Adama catchment region, both direct and indirect. The direct distribution of pharmaceutical items to a health facility to meet the health facility's operational requirements is known as direct health facility (DHF). Indirect health facility (IDHF) denotes the absence of direct pharmaceutical delivery.

The Vehicle Routing Problem (VRP) is a complex combinatorial optimization problem that falls into the NP-Hard Problem category, which is a problem that requires difficult computation and a significant amount of time as the problem data increases in size [1]. Therefore, from the total number of health facilities only 204 DHF were chosen. The travel distance between the facilities was calculated using a dataset that included the geographic coordinates of each health facility. The coordinates of the EPSS Adama Hub were also given, allowing for the computation of travel distances from facilities to the EPSS Adama Hub (depot). Using a separate dataset with mappings of all health facilities to their coordinate locations, health facilities without coordinate locations were considered missing values, and the missing values were replaced using the imputation technique. As a consequence, a dataset with 205 rows, 1 depot (EPSS Adama Hub), and 204 health

facilities was created. Instead of utilizing their names, the algorithm was later identified by the index of each position 0-205, which fitted the implementation well.

3.4.2. General Assumptions made during route optimization

The goal is to determine a set of minimum cost vehicle routes that start and end at the depot, assuming a fleet of homogeneous vehicles with given internal and external dimensions of length, width, and height.

- A fixed-capacity fleet of homogeneous vehicles.
- Customers' geographic locations (health facilities) are known.
- Information about the customer's is known and provided.
- During distribution, each customer's orders are delivered by exactly one vehicle, although each vehicle can service several customers.

3.5. Data Preprocessing

This step includes collecting sample data and deciding which data, including format and size, are needed. These attempts can be guided by background knowledge. Verification of the data's usefulness is part of this stage. Once the business problem and the goal have been established, the following stage is to determine what types of data are available and which data is the most appropriate and relevant for implementing the model that would achieve the goal. Due to its often-large size and likely origin from various, heterogeneous sources, real-world knowledge is prone to being noisy, missing, and inconsistent. As a result, data preparation is essential to acquire the best results from machine learning algorithms on data [28]. To clean the disposed or the original dataset the dataset must be passing through the following data pre-processing methods and techniques.

The goal of this technique was to develop an approach that would be capable of handling future, unknown inputs. i.e., not only perform well on a small number of training data instances, but also generalize well to data from 204 health facilities in the region. Using the domain knowledge, the data was used to expose the algorithm to more generic feasible scenarios. The column vectors represent the location of health facilities at which pharmaceutical products are delivered. Thus, producing datasets of size [n, 205]. Note that the first row corresponds to the EPSS Hub, which is the depot, i.e., where pharmaceutical products are picked.

3.5.2. Data Cleaning

During collecting or entering data, transforming or extracting data, exploring or analyzing data and submitting the draft report for peer review the data may have some errors, outliers and some missing. Therefore; data cleaning was needed for fixing an error occurred [28].

3.5.2.1. Remove Duplication

According to [55], 80% of data scientists' work is data preparation. These reasons would be sufficient to do whatever it takes to get clean dataset. Dataset may include data objects that are duplicates, or almost duplicates of one another. There is no need for duplicate values in dataset. Additionally, duplicate data can return false results that lead to incorrect business decisions. Also, these values affect the accuracy and efficiency of algorithm. To mitigate that risk, deduplication is a vital step in the data-cleaning process. According to [56], data deduplication refers to the process of deleting redundant data from a dataset in order to create a more coherent dataset. Duplicate data must be removed, either inside a dataset or as part of a data model, to ensure accurate and speedy analysis findings.

Deduplication can have a significant impact on the accuracy of a algorithm. In this study, the researcher used two Pandas module approaches to locate and eliminate duplicate data from the dataset. Duplicate () and drop duplicates () are the two methods used to process 4.39 percent of duplicates data.

The information used in this study includes attributes such as health Facility name, longitude, and latitude. The overall number of health facilities is 204, with 9 redundancy data points. Finally, the total count is 194. The overall health facility count in the above details is the original data received after sampling a total dataset. Duplicate data was found in 9 of the 205 members, including the depot, as shown in the diagram below.

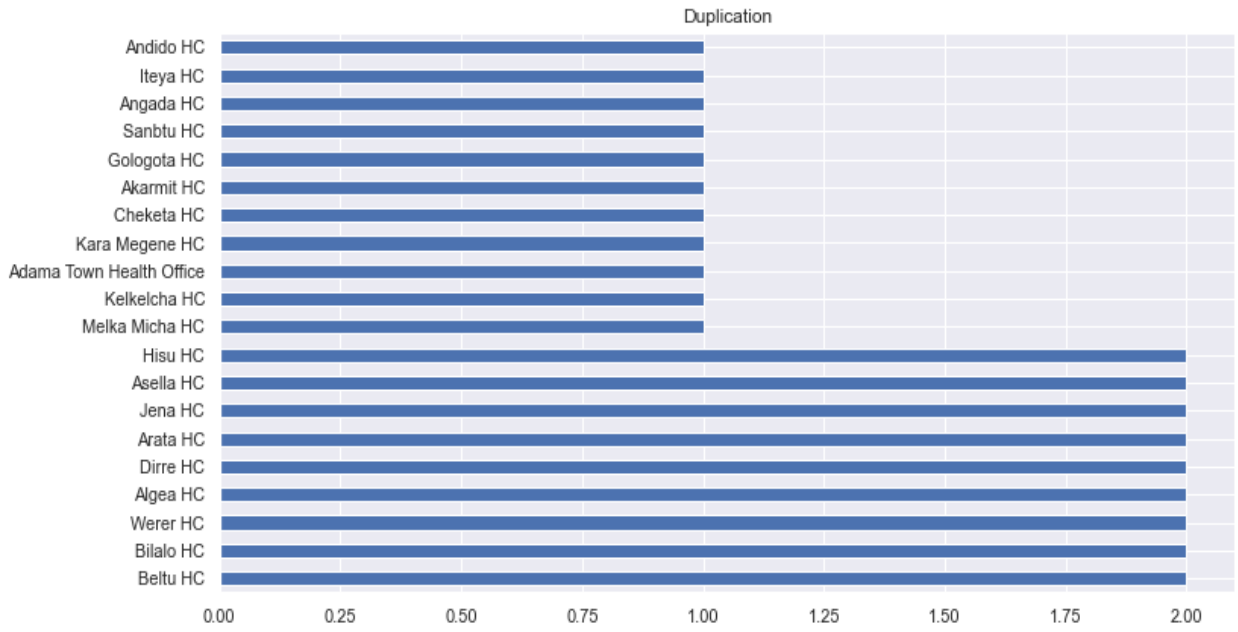


Figure 3.1: Data Duplication Result

3.5.2.2. Missing Value Handling

Missing values refer to the values for one or more attributes in a data was absent or jumped during data collection. This presence of missing values can affect the performance of a classifier constructed using that dataset as a training sample. Scientifically rates of less than 1% missing data are generally considered minor and 1-5% manageable. However, 5-15% needs to refined strategies and more than 15% may cruelly impact any kind of interpretation. [29].

Here in this study, the researcher tries to discuss different methods and techniques to handle missing data and the best methods were selected. The following methods were some of them in case of handling missing values:

Ignore the tuple: This approach is recommended when the class label is missing, but it is ineffective until the tuple has numerous attributes with missing values. It's even worse when the percentage of missing values per attribute fluctuates a lot [57].

Fill in the missing value manually: This approach was time-consuming and may not be feasible given a large data set with many missing values [57].

Replace with a summary: The most commonly used imputation technique. Summarization here is the mean, mode, or median for a respective column [56].

Use a universal constant to fill in the missing value: Replace all missing attribute values by the same constant such as a label “Unknown” [57].

Random replace: replace the missing values with a randomly picked value from the respective column. This technique would be appropriate where the missing values row count is insignificant [57].

Imputation: This technique preserved all cases by replacing missing data with a probable value based on other available information. A simple procedure for imputation was to replace the missing value with the mean or median.

For this study, the researcher chooses imputation approaches that have been used the **mice package** [56] for imputation (Multivariate Imputation by Chained Equations) for handling the missing values. The reason why the researcher selects these methods was which were a mice package gives functions that can impute continuous, binary, and ordered/unordered categorical data, and imputing each incomplete variable with a separate model that, means in one different value were filled in feature column missed space rather than fill the same values in all missed space. Creating multiple imputations as compared to a single imputation (such as mean) takes care of uncertainty in missing values. Therefore, the MICE imputed data on a variable by variable based on specifying an imputation model per variable [52].

The EPSS dataset has 3 variables, two of which give geographic information on health facilities. There are approximately 9 missing values in the data set. As a result, in order to manage missing values, the researcher prefers for imputation methodologies that have been used in the mice package. The imputation step's goal is to fill in missing values numerous times with information from the observed data. After all missing values have been imputed, the data set can be clustered using typical approaches on complete data.

Missing values may exist in a dataset. These are rows of data in which one or more values or columns are missing. The values could be blank or have a particular character or value attached to them. In a loaded dataset, missing values are marked as a NaN (not a number) value, as seen in Table 3.1.

Table 3.1: Sample Data with Missing Value

	Name	longitude	latitude
111	Batu No 1 HC	NaN	38.719245
131	Ali HC	NaN	39.926551
150	Areda Tere HC	NaN	40.759300
159	Dirre HC	8.948740	NaN
168	Hachaltu Gudinan HC	NaN	39.133700
179	Meteteh Bila	NaN	39.709851
181	Mukye Haro HC	NaN	39.266900
186	Welergi HC	8.136421	NaN
190	Lakecha HC	7.382296	NaN
192	Wosha HC	NaN	39.480773

```
In [75]: percent_missing = df.isnull().sum() * 100 / len(df)
         missing_value_df = pd.DataFrame({'column_name': df.columns,
                                          'percent_missing': percent_missing})
         missing_value_df
```

Figure 3.2: Sample Code to Calculate Percent of Missing Value

Table 3.2: Percentage of Missing Value in Dataset

	column_name	percent_missing
Name	Name	0.000000
longitude	longitude	3.589744
latitude	latitude	1.538462

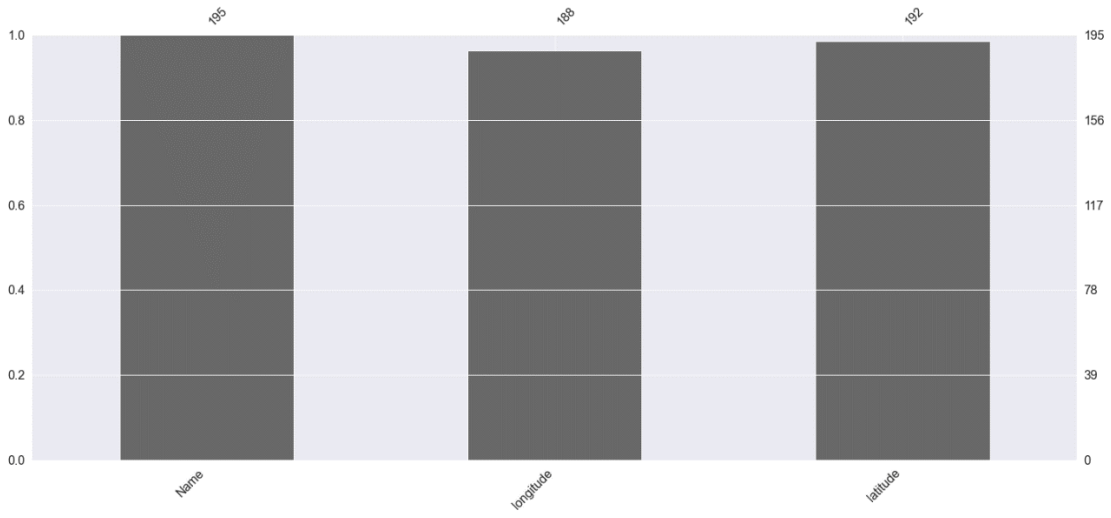


Figure 3.3: Missing Value

The percentage of missing value was estimated as indicated in Figure 3.2 with respect to an outcome result with the variables, as provided in Table 3.2 (Longitude and latitude). The figure in Table 3.2 represents the percentage of missing data. As a result, the longitude feature detected 3.58 percent of missing values, whereas the latitude feature found roughly 1.5 percent. As stated previously, data from various research claimed that a missing rate of 5% or less is insignificant.

Most machine learning methods require numeric input values, and each row and column in a dataset must have a value. As a result, missing values can cause problems for machine learning algorithms. As a result, identifying missing values in a dataset and replacing them with a numeric value is typical. This is known as data imputing, or imputation of missing data. As a result, the iterative imputation approach was utilized to impute missing values in this study.

Iterative imputation is a technique in which each feature is modeled as a function of the others, such as when predicting missing values in a regression problem. Each feature is imputed one after the other, allowing previously imputed values to be utilized as part of a model to predict future features. It is iterative because the process is performed numerous times, allowing for ever-better estimations of missing values as missing values across all features are calculated [58].

The Iterative Imputer methods were implemented in Python using the scikit-learn module. Iterative Imputer is a multivariate imputer that estimates each attribute by subtracting it from the others. In a round-robin method, each characteristic with missing values is modeled as a function of other

features. The Iterative Imputer algorithm is similar to the MICE technique, except instead of several imputations, it returns a single imputation. The impyute library was used to create MICE implementations [59].

```
In [54]: # impute missing value using MICE
imputer = IterativeImputer(imputation_order='ascending',max_iter=10,random_state=42,n_nearest_features=5)
imputed_dataset = imputer.fit_transform(df)
```

Figure 3.4: Imputation sample code

Table 3.3: Result Dataset after Imputation Applied

	Name	longitude	latitude
0	Iteya HC	8.126111	39.226111
1	Huruta Health Center	8.145833	39.347500
2	Bote Health Center	8.293333	38.945556
3	Goro HC	6.992778	40.484444
4	Sedika HC	7.632500	39.816111
...	...	...	...
190	Wonber HC	8.647406	38.880109
191	Gedamsa HC	8.350000	39.180000
192	Burka HC	8.146806	38.930059
193	Gello Rephi HC	7.994109	38.654996
194	Haleku Gulenta HC	7.869264	38.646327

195 rows × 3 columns

In order to check for imputed data, the study enumerate each column and report the number of rows with missing values for the column. Also, the numerical checks provide information about the list of all columns in the dataset, the number and percentage of missing values. It can be useful for assessing whether the data is imputed or not.

Table 3.4: Percentage of Missing value after imputation

	column_name	percent_missing
Name	Name	0.0
longitude	longitude	0.0
latitude	latitude	0.0

As noted in chapter three section 3.5.2 the original dataset received from EPSS contains a missing value. However, the presence of missing data can influence our results, especially when a dataset or even a single variable, has a high percentage of values missing. Thus, it is always a good idea to check a dataset for missing data. In order to have cleaned and complete data the study uses the iterative imputation model using MICE method. This also helps to get accurate and precise output from proposed ML model. Therefore, the researcher understand that 2 variables have missing values with 5.1% and in case of values, 9 values were missed from the total of values.

3.6. Clustering Health Facility

Clustering is the process of dividing a population or set of data points into groups so that data points in the same group are more similar to each other and differ from data points in other groups. This is essentially a collection of objects based on their similarities and differences [52]. K-Means clustering, agglomerative clustering, and DBSCAN are three unsupervised learning algorithms for discovering clusters with similar properties from datasets. Due to the numerous possibilities available, selecting the best clustering algorithm for the dataset might be tricky. The cluster's qualities, the dataset's characteristics, the number of outliers, and the number of data items are all critical elements that influence this selection. To select the optimum clustering approach, specific clustering methods are listed.

Partitional clustering: Divide the data object into unique groups. That is, an object cannot be a member of multiple clusters, and each cluster requires at least one object. The number of clusters indicated by the variable k must be specified by the user for these strategies. Many partition clustering techniques are iterative in nature, distributing a subset of data points to k-clusters. K-means and k-medoids are two popular partition clustering techniques [60].

Hierarchical clustering: Cluster assignments are determined by creating a hierarchy in hierarchical clustering. This can be done from the bottom up (Agglomerative clustering) or from the top down (Divisive clustering). A dendrogram is a tree-based point hierarchy produced by these approaches [61].

Density-based clustering: The density of data points in a location is used to calculate cluster assignments. Where there are high concentrations of data points separated by low-density regions, clusters are assigned. However, the approach is slightly sensitive to the starting point, resulting in ambiguity at cluster boundaries, however these difficulties are minor when dealing with large amounts of data [62]. One of the more important algorithms in this category is Density-Based Spatial Clustering of Applications with Noise (DBSCAN). Ester et al. (1996) proposed DBSCAN to find arbitrary shaped clusters. It doesn't need to know how many clusters there are at the start because it recognizes clusters based on density. Instead, distance-based parameters that serve as changeable thresholds are used [61].

3.6.1. K-means clustering

The number of customers in the target supply chain is assumed to be finite; hence, the number of clusters is also finite, and then the number of non-empty clusters k for n customers is a Stirling partition number that is given by:

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k}\binom{k}{i}(-1)^{k-i}i^{n} \qquad (3.1)$$

Where k is number of cluster and n is number of nodes.

In order to solve VRP, using full dataset is not possible especially for a large number of nodes where the number of different combinations will be all the summations for starting from a single cluster to the maximum number of clusters. So, finding a global optimal partitioning solution is an NP-hard problem that is not practical to solve especially in dynamic supply chains.

In order to solve this problem, partitioning-based iterative clustering is used to cluster the customers before performing the routing phase. K-means clustering is the most popular clustering method and is used to develop separated clusters, so that each customer is assigned to exactly one cluster. K-means clustering is an iterative relocation algorithm that minimizes or maximizes the

value of a selected criterion or criteria until convergence. The most important advantage for the K-means clustering method is that it consumes less memory resources compared to other clustering methods such as hierarchical clustering analysis [63], [64]. So, in this work K-means algorithm is elaborated to group customers into core clusters.

For a customer location (ax, ay) and a cluster's centroid (cx, cy), the Euclidean distance between them is given by:

$$d(a, C) = \sqrt{(a_x - c_x)^2 + (a_y - c_y)^2} \qquad (3.2)$$

Figure *3.5* illustrates the algorithm for this procedure.

Algorithms: K-Means algorithms

Begin

specify the number k of clustering to assign.

randomly initialize k centroids.

repeat

expectation: Assign each point to its closet centroid.

maximization: Compute the new centroid (mean) of each cluster.

until the centroid position does not change.

End

Figure 3.5: Algorithm for the K-means clustering

3.6.2. Evaluation Methods

Generally, there are two metrics to evaluate the k-means algorithm such as elbow technique and silhouette technique.

Elbow method:

The number of clusters is one of the parameters in the K-Means clustering method (k). The elbow method, which involves plotting the sum of squared distances versus k values and selecting the inflection point, is a popular approach for determining the ideal value of k. (point of diminishing returns). The SSE is the sum of each point's squared Euclidean distances from its closest centroid. Because this is a measure of inaccuracy, the goal of k-means is to reduce it as much as possible.

The elbow point is the sweet position when the SSE curve begins to bend. This point's x-value is believed to be a good compromise between error and number of clusters [65].

Silhouette analysis: which is based on the silhouette coefficient. The equation for calculating the silhouette coefficient for a particular data point:

$$S(o) = \frac{b(o)-a(o)}{\max\{a(o),b(o)\}} \qquad (3.3)\ [66]$$

Where,

- s(o) is the silhouette coefficient of the data point o
- a(o) is the average distance between o and all the other data points in the cluster to which o belongs
- b(o) is the minimum average distance from o to all clusters to which o does not belong

The value of the silhouette coefficient is between [-1, 1]. A score of 1 denotes the best meaning that the data point o is very compact within the cluster to which it belongs and far away from the other clusters. The worst value is -1. Values near 0 denote overlapping clusters [66].

However, this study uses elbow technique for quick response and intuition of number of k or number of customers for k-means algorithm.

3.6.3. Result of Clustering

Generally, different clustering algorithms are described in the above concepts. Here in these study K-means iterative clustering is applied on given dataset and optimal value of k or number of customers can be calculated based on elbow method. The Elbow method is a visual method to test the consistency of the best number of clusters by comparing the difference of the sum of square error (SSE) of each cluster, the most extreme difference forming the angle of the elbow shows the best cluster number [60]. The below Figure 3.8 describes elbow point or number of clusters used to on K-means algorithms.

```
In [64]: # Elbow Curve
         K_clusters = range(1,10)
         kmeans = [KMeans(n_clusters=i) for i in K_clusters]
         Y_axis = df[['lon']]
         X_axis = df[['lat']]
         score = [kmeans[i].fit(Y_axis).score(Y_axis) for i in range(len(kmeans))]
         # Visualize
         plt.plot(K_clusters, score)
         plt.xlabel('Number of Clusters')
         plt.ylabel('Score')
         plt.title('Elbow Curve')
         plt.show()
```

Figure 3.6: Sample Code for Elbow Method

```
In [65]: # Use Kneedle library to determine elbow point
         kneedle = KneeLocator(K_clusters, score, S=1.0, curve="concave", direction="increasing")
         n_clusters = round(kneedle.knee, 3)

         print(round(kneedle.knee, 3))
```

Figure 3.7: Sample Code to Determine Elbow Point or Number of Cluster

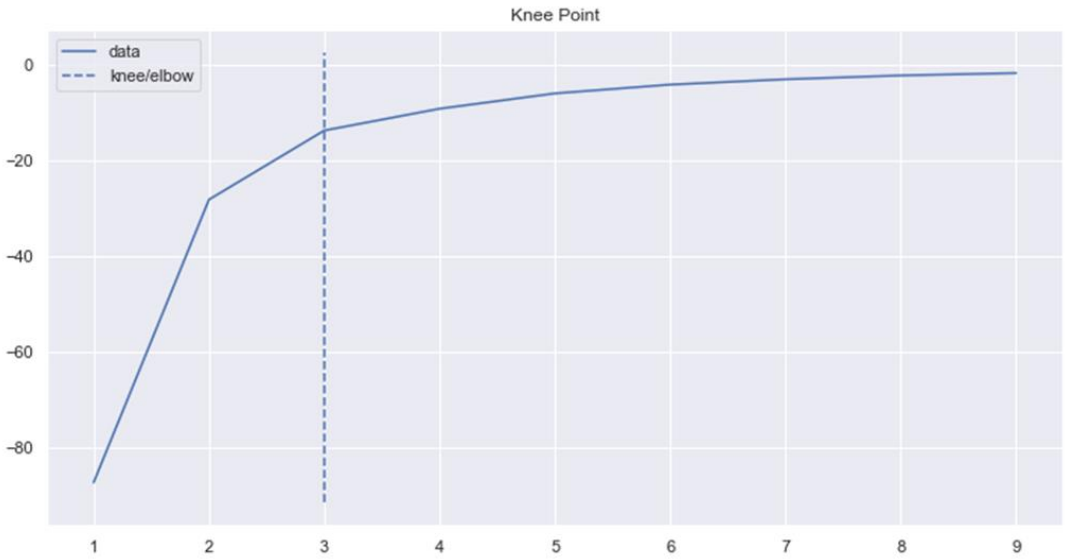


Figure 3.8: Elbow point / Number of Cluster Visualization

The 205 datasets had gone through a cleaning process to remove junk data, producing 194 datasets. By examining the point position on the "elbow" arm, the cluster's number was obtained. Figure 3.8 depicts the Elbow Graph, with the elbow shape fixed at point 3. The best cluster has a number of three in this visualization.

```
In [240]: # cluster using KMeans
kmeans = KMeans(n_clusters, init ='k-means++')
kmeans.fit(X[X.columns[1:3]]) # Compute k-means clustering.
X['cluster_label'] = kmeans.fit_predict(X[X.columns[1:3]])
centers = kmeans.cluster_centers_ # Coordinates of cluster centers.
labels = kmeans.predict(X[X.columns[1:3]]) # Labels of each point
X.head(10)
```

Figure 3.9: Sample Code to Cluster Dataset Using K-Means

Table 3.5: Result of Clustering Method

Name	longitude	latitude	cluster_label
Wetera Gola HC	8.130843	39.608744	1
Batu No 2 HC	7.913790	38.710041	0
Sire robi HC	8.480948	39.164789	0
Huruta Health Center	8.145833	39.347500	0
Gasera HC	7.373628	40.198261	1
Gumguma HC	7.516576	39.078755	0
Metehara HC	8.906567	39.926451	1
Arerti HC	8.928900	39.425700	1
Bulbula HC	7.724200	38.641900	0
Denkaka HC	8.693934	39.050407	0
Denebi Gudo HC	8.273008	39.691160	1
Adele HC	7.795320	39.898766	1
Sire HC	8.277173	39.498431	1
Koka HC	8.443381	39.029739	0
Bishoftu HC	8.747661	38.953700	0
Asella HC	7.959423	39.125100	0
Abosa HC	8.022436	38.721877	0
Dheke Bora HC	8.702000	39.215600	0
Welergi HC	8.136421	39.578386	1
Chancho HC	8.257400	40.129900	1

20 rows × 4 columns

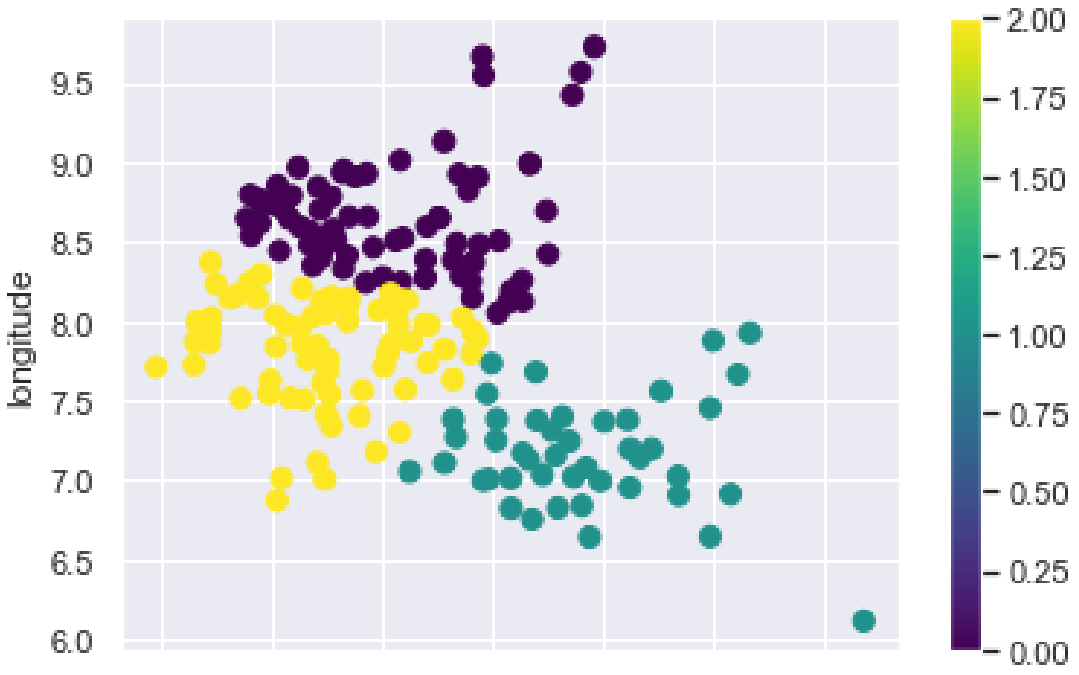


Figure 3.10: Visualize Result of K-means clustering

As described in section 3.4.1., the reason why the study focused clustering on K-Means algorithm is that, it is the best clustering algorithm for coordinate location. Because, it can handle large volumes of data, works on different data types and based on above stated characteristics of the clusters. EPSS health facility catchment dataset clusters can be seen in Figure 3.10, whereas the Figure 3.11 contain dataset after clustering based on elbow analysis as seen in Figure 3.8.

Each cluster will be served by a single truck and will have at least one customer. The clustering that results have the following properties:

i. At least one customer should be present in a cluster.
ii. Each customer is assigned to only one cluster;
iii. Clusters are designed such that all customer in the cluster can be handled by a single truck.

To provide an overview of the approach, the study presents an example with 6 customers in Figure 3.11. A potential outcome of our k-means clustering approach is the determination of three clusters with at least one customer each, where each cluster is served by a single vehicle.

Note that one vehicle can serve multiple customers and the sequence of customers served by a vehicle is determined in a subsequent stage by solving a vehicle routing Problem (VRP) for each vehicle. The final outcome of our approach is presented in Figure 3.12 where we solve the respective VRP problem for each vehicle.

To summarize, our approach comprises the following steps:

Step 1: solve the VRP problem to find a set of optimal cost routes for a fleet of vehicles considering first the determination of clusters with at least one customer each. (Outcome of Figure 3.11);

Step 2: for each vehicle visiting one or more customer and solve a VRP to determine the optimal order of serving the customers (outcome of Figure 3.12).

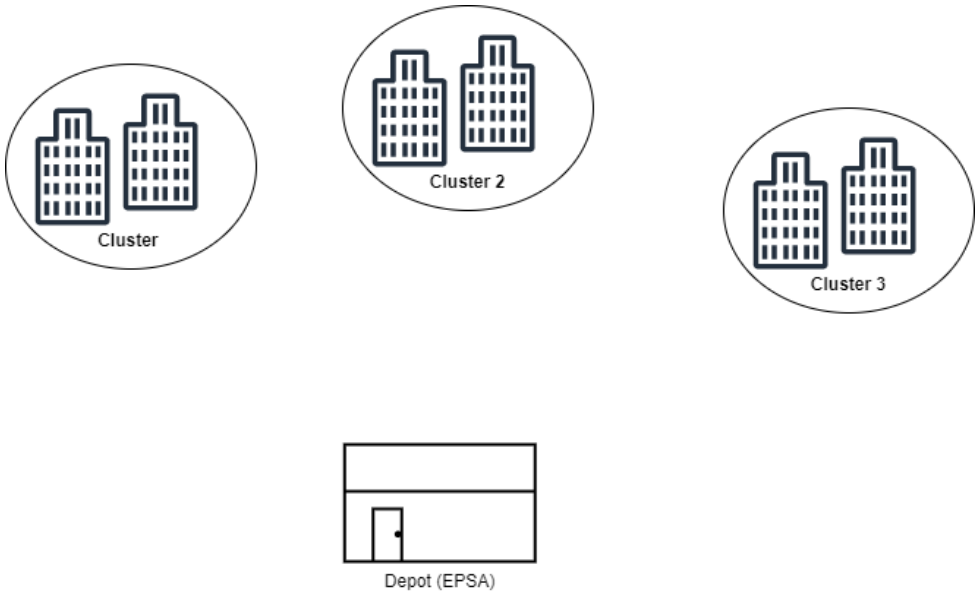


Figure 3.11: Outcome of cluster

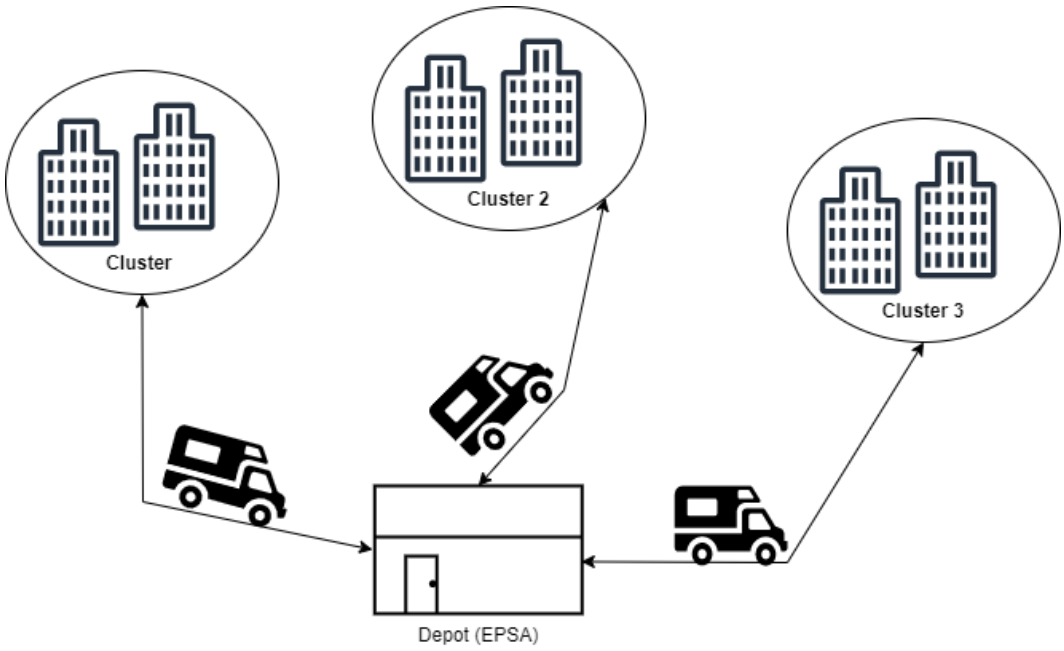


Figure 3.12: Outcome of route optimization

CHAPTER FOUR

4. IMPLEMENTATION, RESULT AND DISCUSSION

This chapter presents the results of proposed solution to represent the evaluated metrics. The purpose of the study was to implement a model that could create optimized drug distribution roots based on cost consumption metrics. After discussing the results, the last section of this chapter presents and describes the most important results from several perspectives. This is followed by an important evaluation of the method. The implemented actor critic model was trained using the RL algorithm called REINFORCE on a dataset containing EPSS scenarios from 204 health facilities and one hub. Then the implementation and results shown in Chapter 4 should be interpreted.

4.1. Implementation

In the implementation, the VRP problems are solved using the NumPy, torch, matplotlib Python packages that implements the RL algorithm. All tests are conducted in a general-purpose computer with a 2.3 GHz Intel Core 5 processor and an 8GB RAM.

In summary, Figure 4.2 shows two approaches to solving the vehicle routing problem (VRP). These are cluster-first route second approaches and direct route optimization using the ML model. The study uses both clustered and un-clustered data in order to gain insights into the accuracy and efficiency of the approach in minimizing the cost of the route.

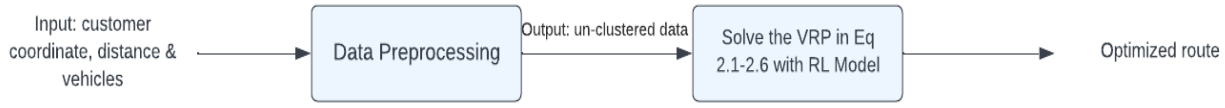


Figure 4.1: Implementation overview of existing approach

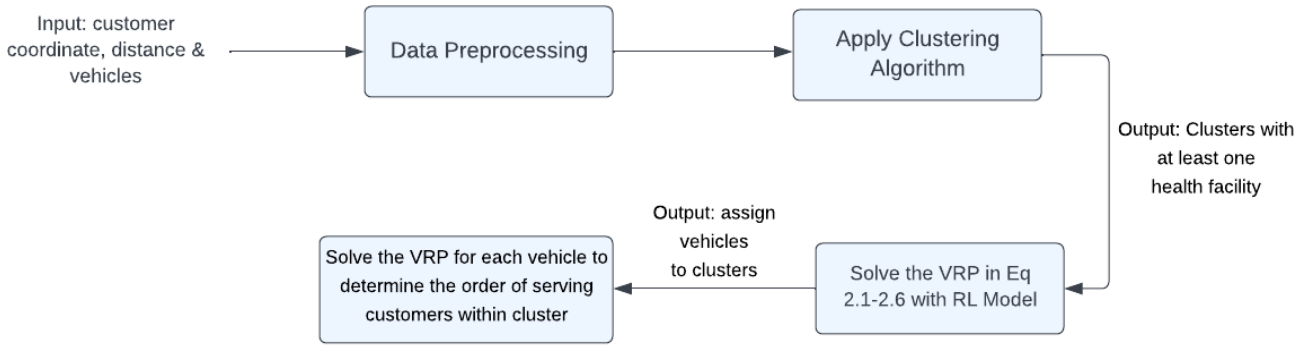


Figure 4.2: Implementation overview of the proposed approach

Initially, information regarding the coordinate location of customers and the number of available vehicles was used to implement the RL model. In the first approach, all dataset is given to the RL model in order to optimize the route. In the second approach, the customers are first clustered based on the distances between them, followed by the RL model to solve VRP.

The specific work presented by Nazari et.al [5] was chosen since it coincided with the researched problem well. Their usage of an RL optimization model was deemed as an interesting approach due to its inherent upside in terms of speed, generalization, and scalability. The heuristic aspects of the models were considered to fit the real-world nature of the researched problem. Before implementing the technique, data is preprocessed to create different sizes of real-world VRP instances that fit the algorithm. Prepared VRP instances are employed in the existing approach, just as they are in the algorithm. The proposed method, on the other hand, optimizes routes using clustered VRP instances. Our training methods are adapted, and we leave the details in the Appendix.

After the VRP model was implemented, the process of adapting that code to fit the pharmaceuticals delivery problem began. This process consisted of implementing problem specific constraints and functionalities into the model. Simultaneously, the more general frameworks such as loading and saving data as well as plotting functionalities were implemented. The main differences in implementation between this study and [5], i.e. the adaptations made, are presented in the following sections, 4.4.1 - 4.5.4.

4.1.1. Input structures

The static input, previously represented as a set of coordinates, was changed to a normalized version of the adjacency matrix. Coordinates representations were not feasible, considering that roads connecting facility very seldom are straight lines. The static inputs were not sampled each training iteration, instead the same matrix was used. The dynamic inputs were represented by the datasets, as previously described in section 3.2.1.4.

4.1.2. Reward function

As different roads have different speed limits, time were not considered a plausible metric to utilize in the reward function. Instead, the rewards were calculated using the distance between facilities listed in the static input. As described in section 2.2 the route produced by the model was interpreted as a set of routes. Therefore, distance corresponding to the vehicle driving from the depot (EPSS Hub) to any facility were considered, since the single routes consists of multiple facilities. It should be noted that originally the model was implemented only to consider total cost as a metric used in the reward function.

4.1.3. Update function

Since the implementation in this study considered delivering pharmaceuticals product to multiple facilities, rather than picking up from health facilities, the vehicle load after delivering to all facilities in the route was initialized to zero. The update function was implemented to, at each time step, increment the space left or load of the vehicle and decrement the number of products while delivering to facility.

4.1.4. Masking probabilities

A problem specific constraint was stated in agreement with the principal. A maximum number of stops per route equals to the number of facilities in the route and a list of facilities with direct transport policies were to be implemented, i.e., no stops were allowed after visiting these facilities. These constraints were implemented in order to generate realistic model produced routes. To prevent the model from violating any constraints, a functionality to force the vehicle to the depot (EPSS Hub) was used. This functionality consisted of masking the probabilities of all actions,

besides choosing the depot, to zero. The functionality was utilized if the following conditions were met:

- The vehicle was located at health facility with a policy of direct transfer to the depot

This was used to prevent the vehicle from selecting the depot with its first action. Also, this prevented the vehicle from repeatedly selecting the depot, since the update function would reset the load when visiting the depot.

4.2. Experimental setup

In this study, a personal computer is used for all experiments. Table 4.1 shows the hardware and software specifications for the machines used in all the experiments.

Table 4.1: Experiment setups

Manufacturer	LENOVO
Model	V50t Desktop
Processor	Intel® Core™ i5-10400U CPU @ 2.9GHz × 6
Memory (RAM)	8 GB
Operating System	Windows 10

In addition to the hardware and software specifications, running the experiment requires the preparation of a VRP instance from the dataset. The experiment uses VRP instances of various sizes to validate the performance of the proposed method. In addition, this study compares the path length of the VRP solution obtained by the proposed approach with the path length obtained by the results of the existing approach. For random instances of 20, 50, 100 nodes. During the pre-processing phase, the investigation takes 195 VRP instances, including the depot, and uses some of them for each size of the problem. 195 was chosen because researchers need different sets of problem configurations. It can be big or small. These VRP instances are extracted from the population with an equal probability distribution. This approach optimizes the clustering results by starting with the clustering VRP node and reaching the outage criteria for visiting all health facilities. This study uses a masking scheme to prevent the node from being accessed multiple times. Table 4.2 presents the data properties before and after pre-process.

Table 4.2 data properties

Before and after pre-process	Total number of nodes
Before	205
After	195

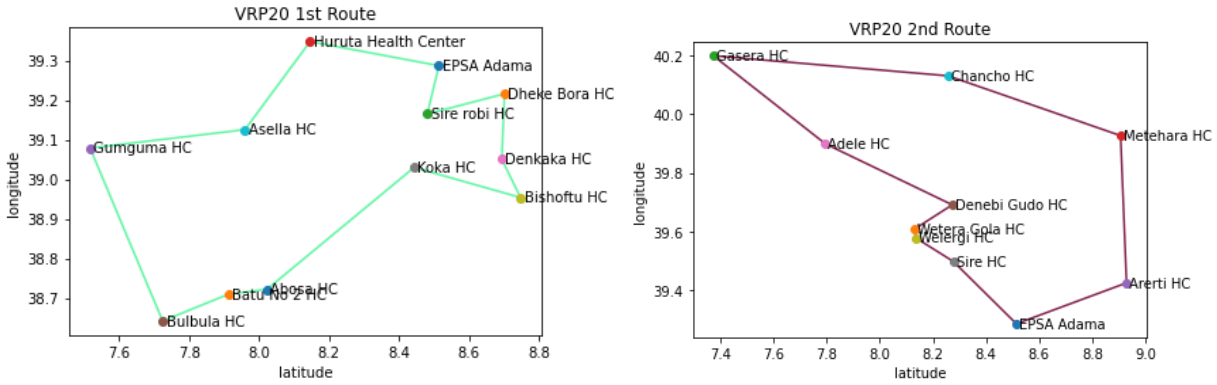
4.3. Experimental Scenarios

Various experiments were conducted to answer research questions from researchers. In general, these experiments fall into two categories based on the two available approaches. The first group of experiments is intended to show the evaluation of an algorithm using clustered VRP instances of various sizes. Analytical and research approaches are presented using algorithm and RQ2 responses. The second group of experiments performs analysis and experiments based on non-clustered VRP instances of various sizes and how they justify and resolve VRPs. Based on the second experiment, the basics and performance of the approaches used in this study to optimize different VRP instances are compared to the proposed approaches. Therefore, this experiment answers RQ2 and RQ3. The remaining other questions will be answered based on the literature review.

4.4. Results and Discussion

This section provides detailed VRP results, including a comparison of the results of the two approaches and an illustration of the generated solution. Researchers have shown the effectiveness of clustering to further improve the quality of the solution. In the implementation, the VRP is solved by the Python package. As already mentioned, the globally optimal solution for NP-Hard VRP problems can only be computed in small instances [1]. To demonstrate the proposed and existing approach to VRP, this study develops a new set of problem instance instances consisting of three categories with problem instances of different sizes.

These problem instances are generated to resemble real problems in the pharmaceutical supply chain. These categories include VRP20, which has a total of 20 nodes (1 depot and 19 customers/ health facilities), VRP50 (1 depot and 49 health facilities), and VRP100 (1 depot and 99 health facilities). The results are shown below and explained accordingly.



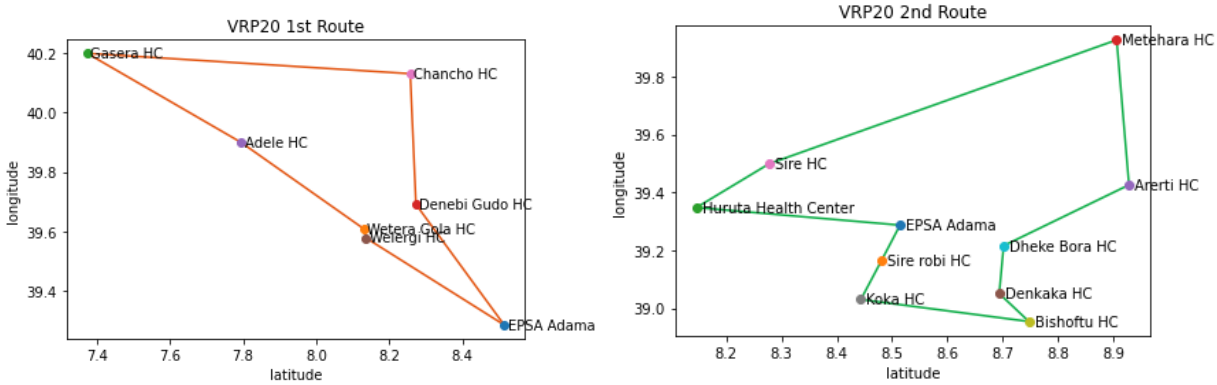
Tour Length = 321.33 (km) Tour Length = 390.4 (km)

Figure 4.3: VRP20 instance results based on the proposed approach

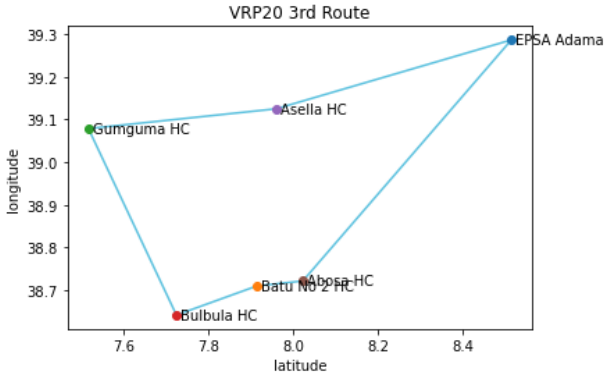
Table 4.3: VRP20 instance solution with proposed approach

Route/Vehicle	Path	Tour Length
1	EPSS Adama -> Huruta Health Center -> Asella HC -> Gumguma HC -> Bulbula HC -> Batu No 2 HC -> Abosa HC -> Koka HC -> Bishoftu HC -> Denkaka HC -> Dheke Bora HC -> Sire robi HC -> EPSS Adama	321.33
2	EPSS Adama -> Arerti HC -> Metehara HC -> Chancho HC -> Gasera HC -> Adele HC -> Denebi Gudo HC -> Wetera Gola HC -> Welergi HC -> Sire HC -> EPSS Adama	603.02

The outcomes of a VRP20 problem instance utilizing the proposed approach are shown in Table 4.3. The experiment's findings reveal that only two routes/vehicles are required to solve the specified VRP instance. Column 1 shows the number of routes/vehicles in further detail. Columns 2 and 3 show the route's path and tour length, respectively.



Tour Length: 315.22 (km) Tour Length: 297.57 (km)



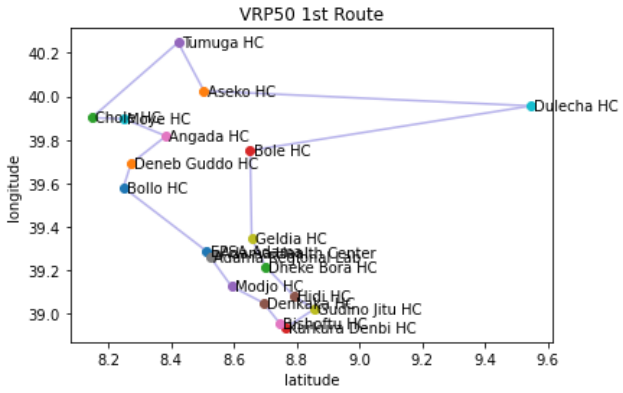
Tour Length: 244.91 (km)

Figure 4.4: Results of a VRP20 instance based on an existing approach

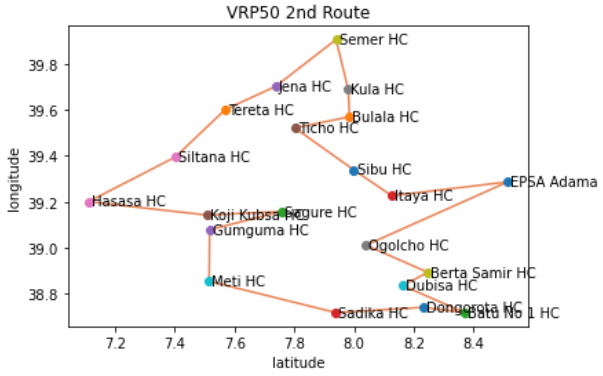
Table 4.4: Results of VRP20 instances using the existing approach

Route/Vehicle	Path	Tour Length
1	EPSS Adama -> Denebi Gudo HC -> Chancho HC -> Gasera HC -> Adele HC -> Wetera Gola HC -> Welergi HC -> EPSS Adama	315.22
2	EPSS Adama -> Huruta Health Center -> Sire HC -> Metehara HC -> Arerti HC -> Dheke Bora HC -> Denkaka HC -> Bishoftu HC -> Koka HC -> Sire robi HC -> EPSS Adama	297.57
3	EPSS Adama -> Abosa HC -> Batu No 2 HC -> Bulbula HC -> Gumguma HC -> Asella HC -> EPSS Adama	244.91

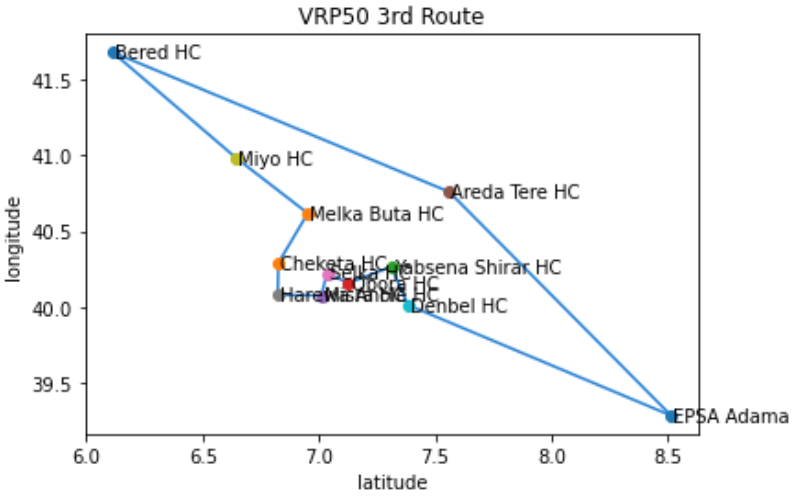
In the VRP20 instances the study present the results when solving VRP using the existing approach (see Table 4.4). The solutions of these instances show 3 route/vehicle to solve VRP.



Tour Length = 477.579 (km)



Tour Length = 523.979 (km)



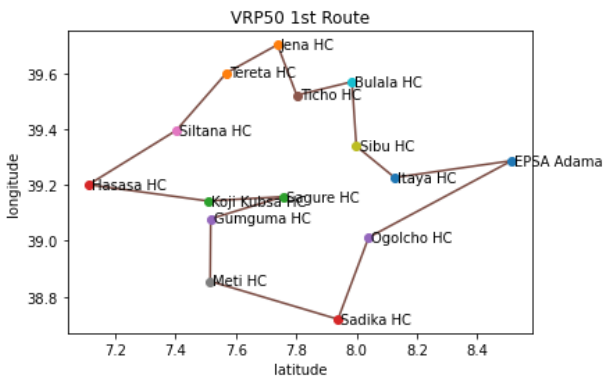

```
Tour Length = 756.85 (km)
```

Figure 4.5: VRP50 instance results based on the proposed approach

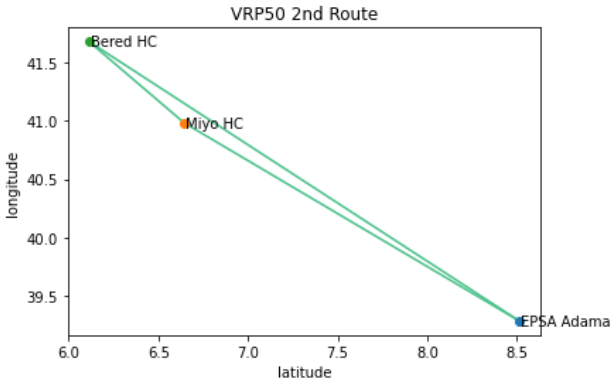
Table 4.5: Results of VRP50 instances using the proposed approach

Route/Vehicle	Path	Tour Length
1	EPSS Adama -> Adama Health Center -> Adama Regional Lab -> Modjo HC -> Denkaka HC -> Bishoftu HC -> Kurkura Denbi HC -> Gudino Jitu HC -> Hidi HC -> Dheke Bora HC -> Geldia HC -> Bole HC -> Dulecha HC -> Aseko HC -> Tumuga HC -> Chole HC -> Moye HC -> Angada HC -> Deneb Guddo HC -> Bollo HC -> EPSS Adama	477.579
2	EPSS Adama -> Itaya HC -> Sibu HC -> Ticho HC -> Bulala HC -> Kula HC -> Semer HC -> Jena HC -> Tereta HC -> Siltana HC -> Hasasa HC -> Koji Kubsa HC -> Sagure HC -> Gumguma HC -> Meti HC -> Sadika HC -> Dongorota HC -> Batu No 1 HC -> Dubisa HC -> Berta Samir HC -> Ogolcho HC -> EPSS Adama	523.979
3	EPSS Adama -> Areda Tere HC -> Bered HC -> Miyo HC -> Melka Buta HC -> Cheketa HC -> Harewa Anole HC -> Misra HC -> Selka HC -> Obora HC -> Yabsena Shirar HC -> Denbel HC -> EPSS Adama	756.85

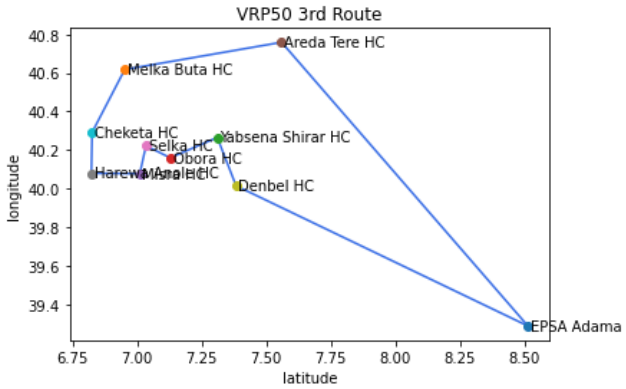
For instances of VRP50, Table 4.5 shows the results of using the proposed approach with 50 customers and one depot. The solutions for these instances show that three vehicles may be suitable for a particular instance.



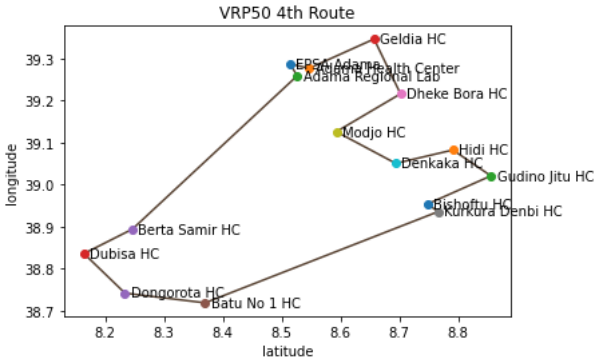
Tour Length: 420.06 (km)



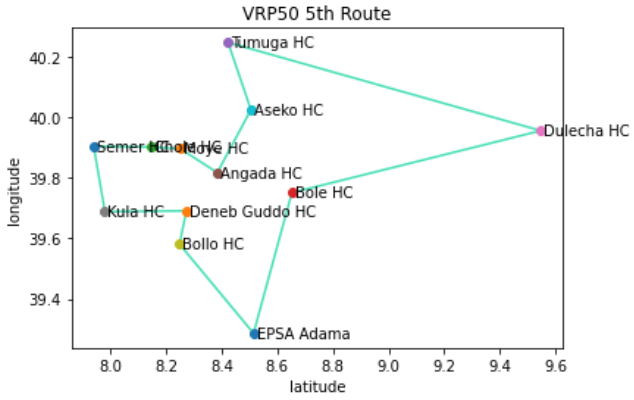
Tour Length: 670.28 (km)



Tour Length: 514.92(km)



Tour Length: 214.649 (km)



Tour Length: 427.27(km)

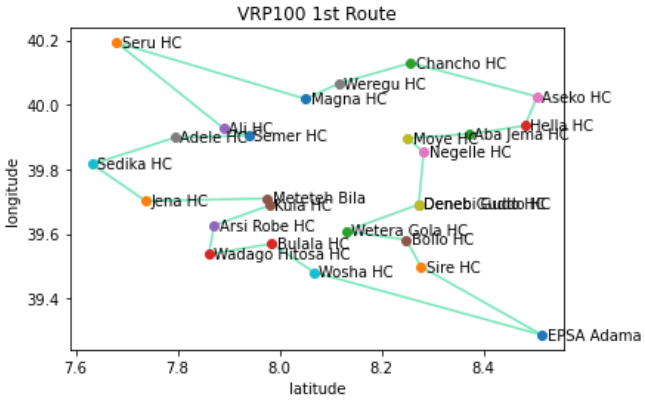
Figure 4.6: VRP50 instance solution based on existing approach

Table 4.6: Results of VPR50 instances using existing approach

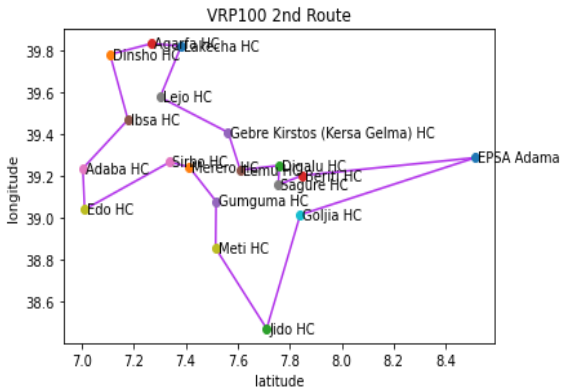
Route/Vehicle	Path	Tour Length
1	EPSS Adama -> Itaya HC -> Sibu HC -> Bulala HC -> Ticho HC -> Jena HC -> Tereta HC -> Siltana HC -> Hasasa HC -> Koji Kubsa HC -> Sagure HC -> Gumguma HC -> Meti HC -> Sadika HC -> Ogolcho HC -> EPSS Adama	420.06
2	EPSS Adama -> Bered HC -> Miyo HC -> EPSS Adama	670.28
3	EPSS Adama -> Denbel HC -> Yabsena Shirar HC -> Obora HC -> Selka HC -> Misra HC -> Harewa Anole HC -> Cheketa HC -> Melka Buta HC -> Areda Tere HC -> EPSS Adama	514.92
4	EPSS Adama -> Adama Regional Lab -> Berta Samir HC -> Dubisa HC -> Dongorota HC -> Batu No 1 HC -> Kurkura Denbi HC -> Bishoftu HC -> Gudino Jitu HC -> Hidi HC -> Denkaka HC -> Modjo HC -> Dheke Bora HC -> Geldia HC -> Adama Health Center -> EPSS Adama	214.649
5	EPSS Adama -> Bollo HC -> Deneb Guddo HC -> Kula HC -> Semer HC -> Chole HC -> Moye HC -> Angada HC -> Aseko HC -> Tumuga HC -> Dulecha HC -> Bole HC -> EPSS Adama	427.27

The results obtained from the existing approach of VRP50 instance are summarized in

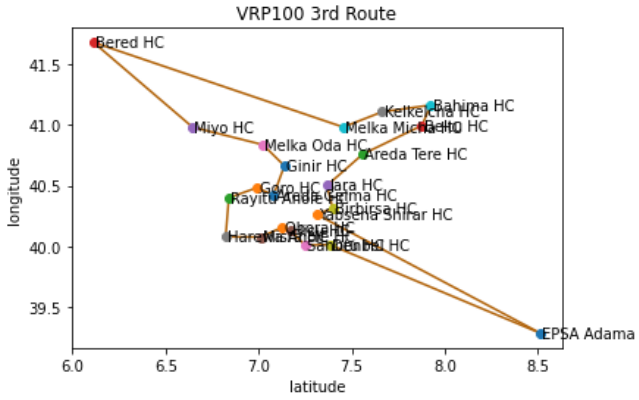
Table *4.6*. According to the result 5 route is formed that 1 vehicle used for each route.

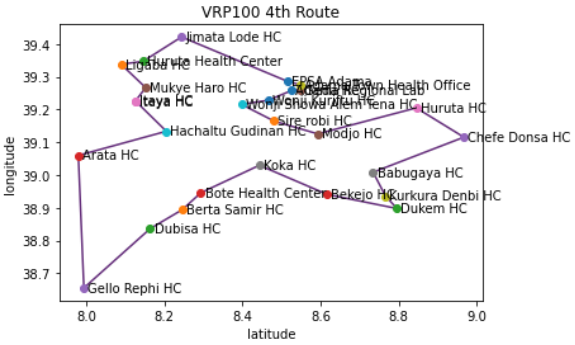


Tour Length: 398.78 (km)



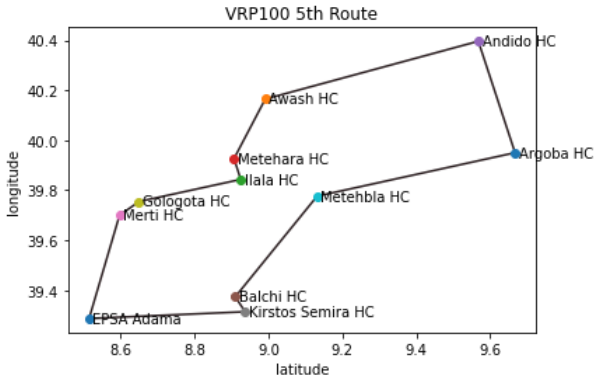
Tour Length: 542.31 (km)





Tour Length: 873.609 (km)

Tour Length: 342.61 (km)



Tour Length: 363.659 (km)

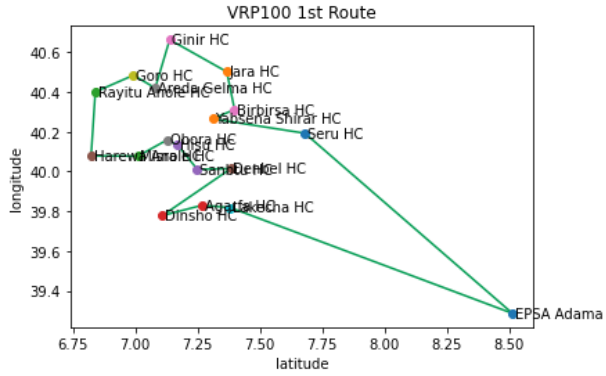
Figure 4.7: VRP100 instance results based on the proposed approach

Table 4.7: Results of VRP100 instances using the proposed approach

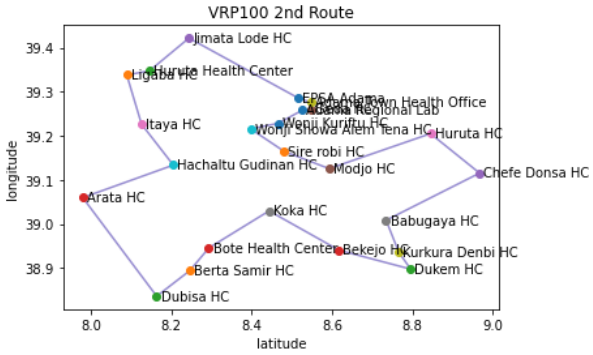
Route/Vehicle	Path	Tour Length
1	EPSS Adama -> Wosha HC -> Bulala HC -> Wadago Hitosa HC -> Arsi Robe HC -> Kula HC -> Meteteh Bila -> Jena HC -> Sedika HC -> Adele HC -> Semer HC -> Ali HC -> Seru HC -> Magna HC -> Weregu HC -> Chancho HC -> Aseko HC -> Hella HC -> Aba Jema HC -> Moye HC -> Negelle HC -> Deneb Guddo HC -> Denebi Gudo HC -> Wetera Gola HC -> Bollo HC -> Sire HC -> EPSS Adama	398.78
2	EPSS Adama -> Beriti HC -> Sagure HC -> Digalu HC -> Lemu HC -> Gebre Kirstos (Kersa Gelma) HC -> Lejo HC -> Lakecha HC -> Agarfa HC -> Dinsho HC -> Ibsa HC -> Adaba HC -> Edo HC -> Sirbo HC -> Merero HC -> Gumguma HC -> Meti HC -> Jido HC -> Goljia HC -> EPSS Adama	542.31
3	EPSS Adama -> Denbel HC -> Sanbtu HC -> Hisu HC -> Obora HC -> Misra HC -> Harewa Anole HC -> Rayitu Anole HC -> Goro HC -> Areda Gelma HC -> Ginir HC -> Melka Oda HC -> Miyo HC -> Bered HC -> Melka Micha HC -> Kelkelcha HC -> Bahima HC -> Beltu HC -> Areda Tere HC -> Jara HC -> Birbirsa HC -> Yabsena Shirar HC -> EPSS Adama	873.609

4	EPSS Adama -> Adama Town Health Office -> Geda HC -> Adama Regional Lab -> Wonji Kuriftu HC -> Wonji Showa Alem Tena HC -> Sire robi HC -> Modjo HC -> Huruta HC -> Chefe Donsa HC -> Babugaya HC -> Kurkura Denbi HC -> Dukem HC -> Bekejo HC -> Koka HC -> Bote Health Center -> Berta Samir HC -> Dubisa HC -> Gello Rephi HC -> Arata HC -> Hachaltu Gudinan HC -> Iteya HC -> Itaya HC -> Mukye Haro HC -> Ligaba HC -> Huruta Health Center -> Jimata Lode HC -> EPSS Adama	342.61
5	EPSS Adama -> Merti HC -> Gologota HC -> Ilala HC -> Metehara HC -> Awash HC -> Andido HC -> Argoba HC -> Metehbla HC -> Balchi HC -> Kirstos Semira HC -> EPSS Adama	363.659

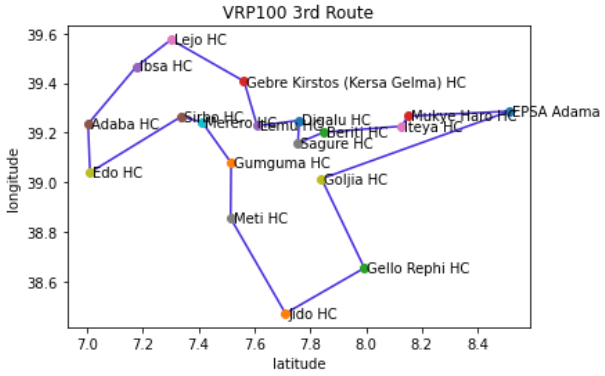
The Table 4.7 illustrates about results of VRP100 instance using proposed approach. It allows to state path of each route, and shows a tour length of each of the routes. In terms of number of vehicle/ routes, the result show 5 that each customer in corresponding route can be served by one vehicle.



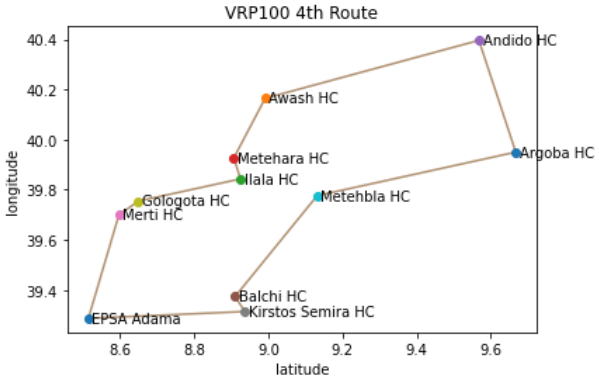
Tour Length: 536.32 (km)



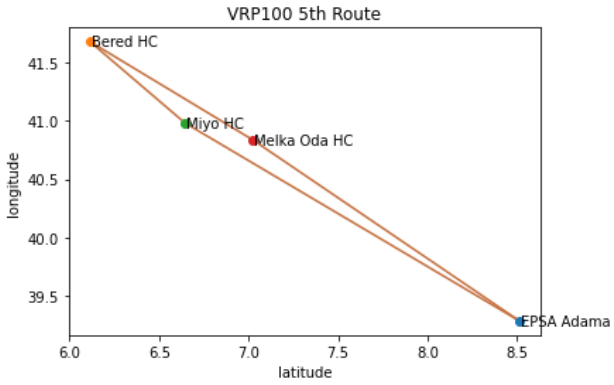
Tour Length: 300.32 (km)



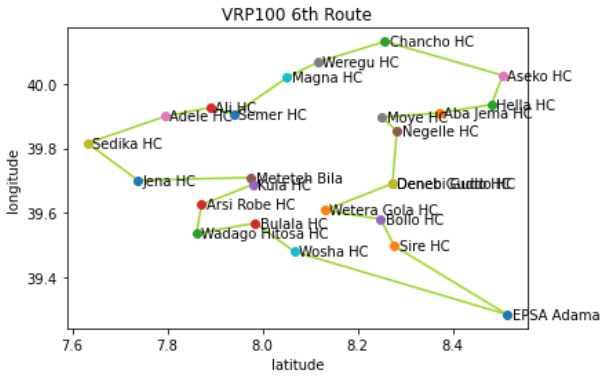
Tour Length: 485.57 (km)



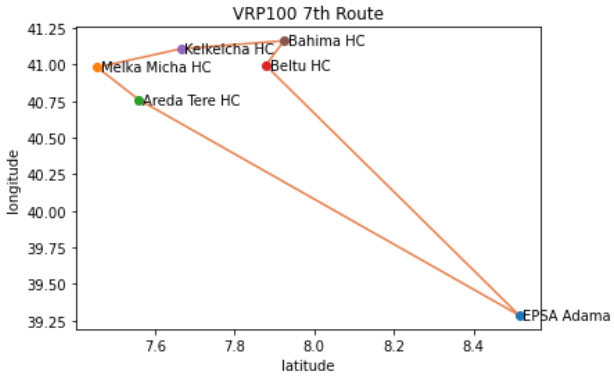
Tour Length: 363.659 (km)



Tour Length: 670.319 (km)



Tour Length: 339.37 (km)



Tour Length: 470.73 (km)

Figure 4.8: VRP100 instance results based on existing approach

Table 4.8: Results of VRP100 instances using existing approach

Route/Vehicle	Path	Tour Length
1	EPSS Adama -> Seru HC -> Yabsena Shirar HC -> Birbirsa HC -> Jara HC -> Ginir HC -> Areda Gelma HC -> Goro HC -> Rayitu Anole HC -> Harewa Anole HC -> Misra HC -> Obora HC -> Hisu HC -> Sanbtu HC -> Denbel HC -> Dinsho HC -> Agarfa HC -> Lakecha HC -> EPSS Adama	536.32
2	EPSS Adama -> Jimata Lode HC -> Huruta Health Center -> Ligaba HC -> Itaya HC -> Hachaltu Gudinan HC -> Arata HC -> Dubisa HC -> Berta Samir HC -> Bote Health Center -> Koka HC -> Bekejo HC -> Dukem HC -> Kurkura Denbi HC -> Babugaya HC -> Chefe Donsa HC -> Huruta HC -> Modjo HC -> Sire robi HC -> Wonji Showa Alem Tena HC -> Wonji Kuriftu HC -> Adama Regional Lab -> Geda HC -> Adama Town Health Office -> EPSS Adama	300.32
3	EPSS Adama -> Goljia HC -> Gello Rephi HC -> Jido HC -> Meti HC -> Gumguma HC -> Merero HC -> Sirbo HC -> Edo HC -> Adaba HC -> Ibsa HC -> Lejo HC -> Gebre Kirstos (Kersa Gelma) HC -> Lemu HC -> Digalu HC -> Sagure HC -> Beriti HC -> Iteya HC -> Mukye Haro HC -> EPSS Adama	485.57

4	EPSS Adama -> Merti HC -> Gologota HC -> Ilala HC -> Metehara HC -> Awash HC -> Andido HC -> Argoba HC -> Metehbla HC -> Balchi HC -> Kirstos Semira HC -> EPSS Adama	363.659
5	EPSS Adama -> Miyo HC -> Bered HC -> Melka Oda HC -> EPSS Adama	670.319
6	EPSS Adama -> Sire HC -> Bollo HC -> Wetera Gola HC -> Denebi Gudo HC -> Deneb Guddo HC -> Negelle HC -> Moye HC -> Aba Jema HC -> Hella HC -> Aseko HC -> Chancho HC -> Weregu HC -> Magna HC -> Semer HC -> Ali HC -> Adele HC -> Sedika HC -> Jena HC -> Meteteh Bila -> Kula HC -> Arsi Robe HC -> Wadago Hitosa HC -> Bulala HC -> Wosha HC -> EPSS Adama	339.37
7	EPSS Adama -> Areda Tere HC -> Melka Micha HC -> Kelkelcha HC -> Bahima HC -> Beltu HC -> EPSS Adama	470.73

The solutions obtained for a sample VRP100 instance are shown in Table 4.8. This study used an existing method for generating these results; the best path is also shown in the table. As shown in the table, 7 vehicles are used to go along 7 routes produced and covers the tour length as show in the table. Each vehicle begins and end its journey at the depot.

Table 4.9: Comparison of proposed approach solution and existing approach with different sizes of VRP instances

Problem	Approach	Instance	Route/Vehicle	Tour Length
VRP	Proposed	20	2	711.73
	Existing	20	3	857.7
	Proposed	50	3	1758.41
	Existing	50	5	2247.18
	Proposed	100	5	2520.97
	Existing	100	7	3166.29

In this initial investigation, the study reports the solution when solving VPR using the proposed and the existing approach when imposing a VRP20, VRP50 and VRP100 instance. Then, compare the solution of proposed approach and the solution of the existing approach. The purpose is to explore the optimality gaps of the proposed approach solutions with respect to the existing approach solution. The latter shows whether clustering leads to a better solution by allowing it to explore the solution space more efficiently.

The first input to the decoder is an embedding of the depot position, as expressed in assumption the vehicle is at the depot at the start. The vehicle picks which of the customer nodes or the depot to

visit in the next step at each decoding phase. An embedding of the depot location and distance after visiting customer node i. The route/ sequence is updated after visiting customer node i. It is a formal definition of the VRP's state transition function used. After sampling a set of nodes to visit, then compute the total vehicle distance and utilize it as the reward signal.

As discussed, to ensure an unbiased comparison, this study use the same experimental setup when solving the VRP using both proposed and the existing approach. The summary results from all size of VRP instances are reported in Table 4.9 where the present the optimality gap(s) when solving the proposed approach and existing approach with respect to their optimal solution. In more detail, column 2 presents the approach that used in each problem instances and column 3 presents the size of the instance, which is number of health facilities to be served by the depot. Column 4 presents the number of vehicle/ routes, which is the vehicle serves each individual health facility exist in the route. Finally, column 5 presents the total tour length for each VRP instances. The researcher notes that all solutions of the instances by proposed approach reported in column 5 are optimal while compared to the existing approach.

RQ2. What is impact of clustering on route optimization?

As show on the Table 4.9, when VRP instance is 20 solved using proposed approach the number of vehicles become 2 which results a tour length of 711.73. On the other hand, using the existing approach, the number of vehicles become 3 and the trip length is 857.7. From the result, when VRP instance is 20, proposed approach results with a smaller number of vehicle as well as short tour length. It is important to emphasize that, based on the 50-node problem instances using proposed approach where 3 vehicle/route are found, the tour length results with 1758.41and vehicle/route difference for the existing approach and the tour length of 5 vehicle/route is 2247.18.

The most interesting aspect of this result is the total distance has been shortened by about 18.6% for VRP20, 24.40% for VRP50 and 22.69% for VRP100 with this proposed method. This implies that the solutions of the proposed approach also use smaller number of vehicles to solve problem while compared to the existing approach, which suggests that the effective utilization of vehicle. In generally, the overall result of the study shows that clustering approach can affect the cost of route optimization in pharmaceutical delivery.

CHAPTER FIVE

5. CONCLUSIONS AND FUTURE WORKS

5.1. Conclusion

In the distribution of pharmaceuticals, there are different customers who need to accommodate different product containers. This study introduces an approach for solving Vehicle Routing Problem (VRP) instances using a clustering algorithm that groups customers and solves VRPs with cluster in mind. In the first stage, the proposed approach cluster customers and assigns vehicles to the cluster. In the second phase, the study determines the tour of each vehicle by considering the customer groups that belong to the assigned cluster of that vehicle.

Research experiments have shown that the proposed approach returns a better solution than the solution for solving non-clustered VRP instances in finding short tour lengths. For instances of larger problems, it is not even possible to compute the solution when solving a non-clustered (existing approach) VRP within the proposed experimental setup. In particular, it shows on average the total distance has been shortened by about 21.89% with the proposed method.

5.2. Recommendation

Taking into account the unique characteristics of pharmaceutical distribution, the paper proposes a method for optimizing the Ethiopian pharmaceutical distribution route, with the goal of minimizing overall distribution costs (including storage, vehicle operating costs, and transportation costs) and the maximum transport distance each time as constraints. The approach used is an RL-based cluster-first route second approach and implemented using a Python module.

The results of the experiments show that the suggested method may effectively address pharmaceutical distribution route optimization problems involving a large number of vehicles and provide a scheme for health facilities with the lowest distribution cost. In compared to traditional distribution systems that rely on intuition and experience, the proposed methodology can minimize overall distribution distance, lower overall expenditures, and improve the quality of Medicare services.

Furthermore, because using the whole dataset to solve VRP is not possible, especially for a big number of nodes with a large number of alternative combinations, applying the clustering approach to the full dataset helps to solve the problem in a real-world scenario. With the increasing growth of health facilities, the advantage of the optimal distribution route is expected to stand out.

5.3. Future works

Future study could extend this approach and apply it to a variety of VRPs, such as the Vehicle Routing Problem with Time Windows (VRPTW), Vehicle Routing Problem with Pickup and Delivery, Split Delivery Vehicle Routing Problem, and Vehicle Routing Problem with Profits (VRPP). Many clusters with varied hyperparameters (e.g., select the customer per cluster and type of product delivered) and optimal settings can be developed to extend the proposed approach. Furthermore, this study does not take into account road conditions, traffic jams, or weather, which will be the focus of future research. Model-based approaches, such as the Neural network approach, can be used to provide adaptive clustering by learning the attributes of a certain type of problem instance.

Reference

[1] F. Mari, W. F. Mahmudy, and P. B. Santoso, “an Improved Simulated Annealing for the Capacitated Vehicle Routing Problem (Cvrp),” Kursor, vol. 9, no. 3, p. 117, 2019, doi: 10.28961/kursor.v9i3.178.

[2] J. M. De Magalhães and J. P. De Sousa, “Dynamic VRP in pharmaceutical distribution-a case study,” Cent. Eur. J. Oper. Res., vol. 14, no. 2, pp. 177–192, 2006, doi: 10.1007/s10100-006-0167-4.

[3] H. Shaban, C. Maurer, and R. J. Willborn, “Impact of Drug Shortages on Patient Safety and Pharmacy Operation Costs.,” Fed. Pract., vol. 35, no. 1, pp. 24–31, 2018, [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/30766319%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6248141

[4] Y. Wang, S. Sun, and W. Li, “Hierarchical Reinforcement Learning for Vehicle Routing Problems with Time Windows,” Proc. Can. Conf. Artif. Intell., 2021, doi: 10.21428/594757db.f0516e23.

[5] M. Nazari, A. Oroojlooy, M. Takáč, and L. V. Snyder, “Reinforcement learning for solving the vehicle routing problem,” Adv. Neural Inf. Process. Syst., vol. 2018-Decem, no. Nips, pp. 9839–9849, 2018.

[6] C. M. Capital and N. Orleans, “E Merging T Rends in C Ommercial R Eal E State F Inance,” vol. 386, no. February, 1999.

[7] M. Nazari, A. Oroojlooy, L. V Snyder, and M. Tak, “Deep Reinforcement Learning for Solving the Vehicle Routing Problem,” 2018.

[8] A. Martinson and X. Qiang, “Route Optimization in logistics distribution based on Particle Swarm Optimization,” Int. J. Comput. Appl., vol. 178, no. 30, pp. 23–27, 2019, doi: 10.5120/ijca2019919179.

[9] Www.affineanalytics.com, “Solution Approach to Resolve Capacitated Vehicle Routing Problem Using Deep Reinforcement Learning,” 2017, [Online]. Available: www.affineanalytics.com%0D

[10] P. P. Repoussis and C. E. Gounaris, “Special issue on vehicle routing and scheduling: Recent trends and advances,” Optim. Lett., vol. 7, no. 7, pp. 1399–1403, 2013, doi:

10.1007/s11590-012-0603-4.

[11] H. Pollaris, K. Braekers, A. Caris, and G. K. Janssens, “The capacitated vehicle routing problem with loading constraints,” pp. 7–12, 2013.

[12] Y. E. Demirtas, E. Özdemir, and U. Demirtaş, “A particle swarm optimization for the dynamic vehicle routing problem,” 6th Int. Conf. Model. Simulation, Appl. Optim. ICMSAO 2015 - Dedic. to Mem. Late Ibrahim El-Sadek, 2015, doi: 10.1109/ICMSAO.2015.7152224.

[13] J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, and A. A. Juan, “Rich vehicle routing problem: Survey,” ACM Comput. Surv., vol. 47, no. 2, 2014, doi: 10.1145/2666003.

[14] P. Campelo, F. Neves-Moreira, P. Amorim, and B. Almada-Lobo, “Consistent vehicle routing problem with service level agreements: A case study in the pharmaceutical distribution sector,” Eur. J. Oper. Res., vol. 273, no. 1, pp. 131–145, 2019, doi: 10.1016/j.ejor.2018.07.030.

[15] J. Rasku, Toward Automatic Customization of Vehicle Routing Systems Toward Automatic Customization of Vehicle Routing Systems. 2019. [Online]. Available: http://urn.fi/URN:ISBN:978-951-39-7826-6

[16] V. Bangalee and F. Suleman, “Evaluating the effect of a proposed logistics fee cap on pharmaceuticals in South Africa - A pre and post analysis,” BMC Health Serv. Res., vol. 15, no. 1, Nov. 2015, doi: 10.1186/s12913-015-1184-6.

[17] Y. Abate, “The Effect of Supply Chain Management Practices on Organizational Performance with the Mediating Role of Inventory Management : The Case of Ethiopian Pharmaceutical Supply Agency Ar ... The Effect of Supply Chain Management Practices on Organizational Per,” no. April, 2020.

[18] L. A. Gladkov, S. N. Scheglov, and N. V. Gladkova, “The application of bioinspired methods for solving vehicle routing problems,” Procedia Comput. Sci., vol. 120, pp. 39–46, 2017, doi: 10.1016/j.procs.2017.11.208.

[19] C. Moryadee and W. A. and M. R. Shaharudin, “Congestion and Pollution , Vehicle Routing Problem of a Logistics Provider in Thailand,” pp. 203–212, 2019, doi: 10.2174/1874447801913010203.

[20] J. E. Vinay, V. Panicker, and R. Sridharan, “Modelling and Analysis of a Green Vehicle Routing Problem,” Twelfth AIMS Int. Conf. Manag., no. 2013, pp. 1310–1318, 2013.

[21] H. Xu, P. Pu, and F. Duan, “Dynamic Vehicle Routing Problems with Enhanced Ant Colony Optimization,” vol. 2018, 2018.

[22] M. Shahrier and A. Hasnat, “Route optimization issues and initiatives in Bangladesh : The context of regional significance,” Transp. Eng., vol. 4, no. February, p. 100054, 2021, doi: 10.1016/j.treng.2021.100054.

[23] R. Maini and R. Goel, “Vehicle routing problem and its solution methodologies: a survey,” 2017. [Online]. Available: https://www.researchgate.net/publication/344245589

[24] X. Zhang and Y. Yin, “Research on the application of genetic algorithm in logistics location,” ICCSS 2017 - 2017 Int. Conf. Information, Cybern. Comput. Soc. Syst., vol. 5064, pp. 435–438, 2017, doi: 10.1109/ICCSS.2017.8091454.

[25] M. Felea and I. Albăstroiu, “Defining the concept of supply chain management and its relevance to romanian academics and practitioners,” Amfiteatru Econ., vol. 15, no. 33, pp. 74–88, 2013.

[26] C. Program-procurement, “Ethiopian Pharmaceuticals Supply Agency Network Analysis,” no. May, 2020.

[27] MOH Ethiopia, Gavi, and JSI, “An Unfinished Journey: Vaccine Supply Chain Transformation in Ethiopia,” 2019, [Online]. Available: https://www.gavi.org/library/publications/gavi-fact-sheets/gavi-supply-chain-strategy/

[28] F. Takes, “Applying Monte Carlo Techniques to the Capacitated Vehicle Routing Problem Master Thesis Frank Takes (ftakes@liacs.nl),” 2010.

[29] J. E. Bell and P. R. McMullen, “Ant colony optimization techniques for the vehicle routing problem,” Adv. Eng. Informatics, vol. 18, no. 1, pp. 41–48, 2004, doi: 10.1016/j.aei.2004.07.001.

[30] M. Bielli, A. Bielli, and R. Rossi, “Trends in models and algorithms for fleet management,” in Procedia - Social and Behavioral Sciences, 2011, vol. 20, pp. 4–18. doi: 10.1016/j.sbspro.2011.08.004.

[31] A. O. Adewumi and O. J. Adeleke, “A survey of recent advances in vehicle routing problems,” Int. J. Syst. Assur. Eng. Manag., vol. 9, no. 1, pp. 155–172, 2018, doi: 10.1007/s13198-016-0493-4.

[32] R. A. Sarker and C. S. Newton., Optimization modelling : a practical introduction, vol. 1, no. 69. 2008.

[33] L. C. Yeun, W. a N. R. Ismail, K. Omar, and M. Zirour, “Vehicle Routing Problem : Models and Solutions,” J. Qual. Meas. Anal., vol. 4, no. 1, pp. 205–218, 2008.

[34] J. Gonzalez-feliu, T. Dottorato, and J. G. Feliu, “Models and Methods for the City Logistics : The Two-Echelon Capacitated Dottorato in Ingegneria Informatica e Sistemi – XX ciclo Models and methods for the City Logistics The Two-Echelon Capacitated Vehicle Routing Problem,” no. February, 2015.

[35] A. Subramanian, E. Uchoa, and L. Satoru, “Computers & Operations Research A hybrid algorithm for a class of vehicle routing problems,” Comput. Oper. Res., vol. 40, no. 10, pp. 2519–2531, 2013, doi: 10.1016/j.cor.2013.01.013.

[36] D. Alexander, “Modern Deep Reinforcement Learning Algorithms,” 2019.

[37] D. L. Poole and A. K. Mackworth, Artificial Intelligence: Foundations of Computational Agents. Cambridge University Press, 2010.

[38] S. S. Richard and G. B. Andrew, Reinforcement Learning: An Introduction, vol. 16, no. 1. The MIT Press Cambridge, 2018. doi: 10.1109/tnn.2004.842673.

[39] T. Pellonper, “Ant colony optimization and the vehicle routing problem,” 2014.

[40] J. Mandziuk, “New Shades of the Vehicle Routing Problem: Emerging Problem Formulations and Computational Intelligence Solution Methods,” IEEE Trans. Emerg. Top. Comput. Intell., vol. 3, no. 3, pp. 230–244, 2019, doi: 10.1109/TETCI.2018.2886585.

[41] J. Alzubi, A. Nayyar, and A. Kumar, “Machine Learning from Theory to Algorithms: An Overview,” J. Phys. Conf. Ser., vol. 1142, no. 1, 2018, doi: 10.1088/1742-6596/1142/1/012012.

[42] M. M. Z. E. Mohammed, E. Bashier, and M. Bashier, Algorithms and Applications, no. December. 2016. doi: 10.1201/9781315371658.

[43] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, “A Review of Machine Learning and IoT in Smart Transportation,” Futur. Internet, vol. 11, no. 4, p. 94, 2019, doi: 10.3390/fi11040094.

[44] B. Bajic, I. Cosic, and M. Lazarevic, “Machine Learning Techniques for Smart Manufacturing : Applications and Challenges in Industry 4 . 0,” no. October, pp. 19–22, 2018.

[45] V. François-lavet et al., “An Introduction to Deep Reinforcement Learning,” Found.

trends Mach. Learn., vol. II, no. 3–4, pp. 1–140, 2018, doi: 10.1561/2200000071.Vincent.

[46] A. Hammoudeh, “A Concise Introduction to Reinforcement Learning,” no. February, 2018, doi: 10.13140/RG.2.2.31027.53285.

[47] A. Alharin and T. Doan, “Reinforcement Learning Interpretation Methods : A Survey,” vol. 8, 2020.

[48] A. Mondal, “A Survey of Reinforcement Learning Techniques : Strategies , Recent Development , and Future Directions A Survey of Reinforcement Learning Techniques : Strategies , Recent Development , and Future Directions,” no. January 2020, 2021.

[49] S. Ruder, “An overview of gradient descent optimization algorithms,” pp. 1–14, 2016, [Online]. Available: http://arxiv.org/abs/1609.04747

[50] B. Çatay, “Ant Colony Optimization and Its Application to the Vehicle Routing Problem with Pickups and Deliveries,” pp. 219–244, 2009, doi: 10.1007/978-3-642-04039-9_9.

[51] C. Singhtaun and S. Tapradub, “Modeling and Solving Heterogeneous Fleet Vehicle Routing Problems in Draft Beer Delivery,” Int. J. Eng. Adv. Technol., no. 8, pp. 2249–8958, 2019.

[52] F. Nelli, Python data analytics: With Pandas, NumPy, and Matplotlib: Second edition. 2018. doi: 10.1007/978-1-4842-3913-1.

[53] M. Wes, Python for Data Analysis. 2017. doi: 10.1097/00007890-200105270-00005.

[54] “torch — PyTorch 1.10 documentation.” https://pytorch.org/docs/stable/torch.html (accessed Feb. 05, 2022).

[55] Gil Press, “Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says.” https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/?sh=785704046f63 (accessed Feb. 19, 2022).

[56] Venkatesh Ganti, “Data Cleaning,” Springer, Redmond, WA, USA, 2004.

[57] K. A. Gaurav and L. Patel, Machine Learning With R. 2020. doi: 10.4018/978-1-7998-2718-4.ch015.

[58] “Iterative Imputation for Missing Values in Machine Learning.” https://machinelearningmastery.com/iterative-imputation-for-missing-values-in-machine-learning/ (accessed Feb. 17, 2022).

[59] O. Altukhova, “Choice of method imputation missing values for obstetrics clinical data,”

in Procedia Computer Science, 2020, vol. 176, pp. 976–984. doi: 10.1016/j.procs.2020.09.093.

[60] E. Umargono, J. E. Suseno, and V. G. S. K., “K-Means Clustering Optimization using the Elbow Method and Early Centroid Determination Based-on Mean and Median,” in Proceedings of the International Conferences on Information System and Technology, Jul. 2019, pp. 234–240. doi: 10.5220/0009908402340240.

[61] F. Murtagh, “Hierarchical Clustering,” in International Encyclopedia of Statistical Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 633–635. doi: 10.1007/978-3-642-04898-2_288.

[62] K. Yeturu, “Machine learning algorithms, applications, and practices in data science,” Handb. Stat., 2020, doi: 10.1016/bs.host.2020.01.002.

[63] G. Nathiya, S. C. Punitha, and Dr. M. Punithavalli, “An Analytical Study on Behavior of Clusters Using KMeans, EM and K* Means Algorithm.” [Online]. Available: http://sites.google.com/site/ijcsis/

[64] H. Huang and K. Kim, “Unsupervised Clustering Analysis of Gene Expression,” CHANCE, vol. 19, no. 3, pp. 49–51, Jun. 2006, doi: 10.1080/09332480.2006.10722802.

[65] “K-means Clustering in machine learning: | by Mukesh Chaudhary | Medium.” https://medium.com/@cmukesh8688/k-means-clustering-in-machine-learning-252130c85e23 (accessed Feb. 18, 2022).

[66] “Silhouette Analysis in K-means Clustering | by Mukesh Chaudhary | Medium.” https://medium.com/@cmukesh8688/silhouette-analysis-in-k-means-clustering-cefa9a7ad111 (accessed Feb. 18, 2022).

APPENDICES

Appendix A

Implementation Detail

The study utilizes 1-dimensional convolution layers for the embedding, with the in-width equal to the input length, the number of filters equal to D, and the number of in-channels equal to the number of x elements. The researcher discovered that training without an embedding layer always results in a poor result. One probable explanation is that the strategy is substantially more efficient at extracting important characteristics from high-dimensional input representations. Because our embedding involves an affine transformation, the embedded input distances are not always proportional to the original 2-dimensional Euclidean distances.

In the decoder, the study employs one layer of LSTM RNN with a state size of 128. Each customer location is also stored in a vector of size 128, which is shared by all inputs. In the critic network, we first compute a weighted sum of the embedded inputs using the actor-network network's output probabilities, and then it has two hidden layers: one dense layer with ReLU activation and another linear layer with a single output. Xavier initialization is used to set up the variables in both the actor and critic networks.

The researcher employs the REINFORCE Algorithm with a learning rate of 10^{-4} to train both networks, as suggested by Nazari et.al [5] .The batch size N is 128, and the gradients are clipped when their norm exceeds 2. In the LSTM decoder, we apply dropout with a probability of 0.1. We also used the entropy regularization, which has been found to be effective in preventing the algorithm from getting stuck in a local optimal state.

Appendix B

Imputation Sample Code

```
# Check percent of missing value
percent_missing = df.isnull().sum() * 100 / len(df)
```

```
missing_value_df = pd.DataFrame({'column_name': df.columns,

                                 'percent_missing': percent_missing})

missing_value_df

df.info()

Name = df.loc[:,['Name']]

# drop Name colomun

df = df.drop(['Name'],axis=1)

# impute missing value

imputer =
IterativeImputer(imputation_order='ascending',max_iter=10,random_state=42,n_nearest_feature
s=5)

imputed_dataset = imputer.fit_transform(df)

# convert numpy array to dataframe

df = pd.DataFrame(imputed_dataset, columns = ['longitude','latitude'])
```

Appendix C

Clustering Sample Code

```
##################### vrp 20 #############

# Elbow Curve

K_clusters = range(1,10)

kmeans = [KMeans(n_clusters=i) for i in K_clusters]

Y_axis = vrp20[['longitude']]

X_axis = vrp20[['latitude']]

score = [kmeans[i].fit(Y_axis).score(Y_axis) for i in range(len(kmeans))]

########## Visualize ############

plt.figure(figsize=(12, 6))
```

```
plt.plot(K_clusters, score)

plt.xlabel('Number of Clusters')

plt.ylabel('Score')

plt.title('Elbow Curve')

plt.show()

kneedle = KneeLocator(K_clusters, score, S=1.0, curve="concave", direction="increasing")

n_clusters = round(kneedle.knee, 3)

print(round(kneedle.knee, 3))

kneedle.plot_knee(figsize=(12, 6))

############## cluster using KMeans ##################

kmeans = KMeans(n_clusters, init ='k-means++')

kmeans.fit(vrp20[vrp20.columns[1:3]]) # Compute k-means clustering.

vrp20['cluster_label'] = kmeans.fit_predict(vrp20[vrp20.columns[1:3]])

centers = kmeans.cluster_centers_ # Coordinates of cluster centers.

labels = kmeans.predict(vrp20[vrp20.columns[1:3]]) # Labels of each point

print(vrp20)

vrp20.to_csv ('vrp20.csv', index=None, header = True)
```

Appendix D

VRP Instances Sample Data

Table 0.1: Sample VRP20 instances

	Name	longitude	latitude

0	Wetera Gola HC	8.130843	39.608744
1	Batu No 2 HC	7.913790	38.710041
2	Sire robi HC	8.480948	39.164789
3	Huruta Health Center	8.145833	39.347500
4	Gasera HC	7.373628	40.198261
5	Gumguma HC	7.516576	39.078755
6	Metehara HC	8.906567	39.926451
7	Arerti HC	8.928900	39.425700
8	Bulbula HC	7.724200	38.641900
9	Denkaka HC	8.693934	39.050407
10	Denebi Gudo HC	8.273008	39.691160
11	Adele HC	7.795320	39.898766
12	Sire HC	8.277173	39.498431
13	Koka HC	8.443381	39.029739
14	Bishoftu HC	8.747661	38.953700
15	Asella HC	7.959423	39.125100
16	Abosa HC	8.022436	38.721877
17	Dheke Bora HC	8.702000	39.215600
18	Welergi HC	8.136421	39.578386
19	Chancho HC	8.257400	40.129900

Table 0.2: Sample VRP50 instance

	Name	**longitude**	**latitude**
0	Deneb Guddo HC	8.273008	39.691160
1	Melka Buta HC	6.951852	40.617856
2	Chole HC	8.148351	39.901936
3	Yabsena Shirar HC	7.310464	40.265086
4	Bole HC	8.652194	39.751152

5	Tumuga HC	8.422714	40.247890
6	Hidi HC	8.791046	39.082215
7	Tereta HC	7.568629	39.600547
8	Sagure HC	7.755829	39.157670
9	Sadika HC	7.938359	38.719245
10	Angada HC	8.384135	39.817617
11	Gumguma HC	7.516576	39.078755
12	Ticho HC	7.801621	39.521070
13	Siltana HC	7.401935	39.394789
14	Adama Regional Lab	8.526300	39.258300
15	Geldia HC	8.656999	39.345702
16	Obora HC	7.128211	40.159428
17	Dulecha HC	9.546700	39.956200
18	Kula HC	7.980400	39.688400
19	Berta Samir HC	8.246200	38.894300
20	Meti HC	7.514731	38.855300
21	Bollo HC	8.248269	39.581162
22	Misra HC	7.010268	40.077382
23	Aseko HC	8.504981	40.025425
24	Sibu HC	7.999232	39.338667
25	Areda Tere HC	7.559704	40.759300
26	Bulala HC	7.985000	39.569167
27	Batu No 1 HC	8.369779	38.719245
28	Itaya HC	8.126121	39.226373
29	Dheke Bora HC	8.702000	39.215600
30	Jena HC	7.737666	39.701486
31	Selka HC	7.034200	40.222600
32	Koji Kubsa HC	7.506525	39.142589

33	Kurkura Denbi HC	8.766167	38.935880
34	Harewa Anole HC	6.822530	40.079400
35	Miyo HC	6.644377	40.981604
36	Hasasa HC	7.111317	39.200788
37	Modjo HC	8.592855	39.125622
38	Denbel HC	7.383311	40.014471
39	Bered HC	6.113094	41.677604
40	Denkaka HC	8.693934	39.050407
41	Ogolcho HC	8.040026	39.011853
42	Cheketa HC	6.825385	40.291725
43	Bishoftu HC	8.747661	38.953700
44	Adama Health Center	8.546600	39.276200
45	Semer HC	7.939718	39.904682
46	Gudino Jitu HC	8.855683	39.020806
47	Moye HC	8.251057	39.896620
48	Dubisa HC	8.163800	38.836200
49	Dongorota HC	8.233593	38.742245