CHAPTER 4

MIXED INTEGER NON-LINEAR PROGRAMMING FOR MOBILE MEDICAL UNITS

4.1 INTRODUCTION

This chapter presents and discusses the process that was utilized to convert the mathematical model into programming codes and finally numerical results. A metaheuristic algorithm was used to carry out this approach, and each step of the procedure was presented and described in this chapter.This chapter demonstrates the process, including the coding software and other hardware requirements, as well as subsections showcasing the employed algorithm.

In this chapter, the problems faced by patients in rural or underdeveloped areas to receive proper healthcare and making healthcare facility accessible to them. By transforming the produced ideas, identified gaps in the literature, and presumptions regarding the study subject into a mathematically stated model, this chapter advances the research to a new level. This model converts the entire issue into mathematical equations, which were then solved to get the desired outcomes. The suggested model was also developed as a MINLPMMUS (Mixed Integer Non-Linear Programming of Mobile Medical Units) model. This chapter is organised as follows: To get a thorough understanding of integer programming, linear programming, MINLPMMUS, and MILP, Section (3.2) provides a overview of Mixed Integer Non-Linear Programming of Mobile Medical Units (MINLPMMUs). The model description is found in section (3.3), which also includes a thorough explanation of how the problem is presented, problem assumptions (which define the necessary presumptions required to assess the model), then we have defined the sets and parameters, required equations (which represents calculation of specific parameters), followed by decision variables, defining objective functions, and necessary constraints for the model.

4.2 MIXED INTEGER NON-LINEAR PROGRAMMING FOR MOBILE MEDICAL UNITS (MINLPMMUS)

Understanding and defining MINLPMMUs models is crucial as the suggested objective and constraints was created for a MINLPMMUs model. In order to achieve this, various terminology must be defined. A function can be minimised or maximised using linear programming (LP), however to use this optimization approach, the goal function and model constraints must meet certain linearity requirements. As we know that the possibility of rounding-off values can give impossible or suboptimal solutions, linear programming was not suitable as it may generate results that contain fractional numbers, that are occasionally undesirable and unrealistic in real-time applications(Sam et al., 2018). In order to address this problem, integer linear programming (ILP) was developed. ILP demands the same linearity criteria as LP while producing integer results for the unknown variables. However, as a result of academics' efforts to address real-world issues and include several objectives into the generated models, the complexity of problems has risen. The mixed integer linear programming (MILP) optimisation approach is supposed to be implemented in this situation because certain solutions must generate integer results while others have to provide continuous fractional values. Three requirements must be satisfied in order to use MILP. Prior to everything else, objective functions have to be defined as linear functions. Finally, linear functions need to have either a minimising or a maximising aim. Constraints should be linear. Researchers frequently make significant efforts at tackling these issues as a result of the increased need to provide solutions to real-world challenges. Therefore, nonlinear objective function, constraint, or both may be included in the constructed models. In order to resolve this issue, optimization models that is the Mixed Integer Non-Linear Programming for Mobile Medical Units (MINLPMMUs) method is employed. Therefore, MINLPMMUs is a method of optimization that makes use of mathematical equations that are used to maximize or minimize a required nonlinear programming function under the same type of constraints. MINLPMMUS also allows the incorporation of integer (discrete) and fractional (continuous) variables. Since it allows for the integration of several actual situations and decision factors, MINLPMMUS is utilised in various real-world scenarios, such as, transport, supply chain management, logistics, manufacturing and garbage collection. Due to extra efforts made to solve a real-life situation, the suggested MMUVRP model has a large lot of complexity, which makes the model more realistic. However, the model also contains several decision variables with various values. For instance, whereas time taken to travel a specific route is fractional, variables like an expert visiting a specific hamlet are binary. Patient satisfaction levels are an example of a variable with real values ($\in [0,1]$). As a result, using MINLPMMUs for optimizations makes more sense and produces more accurate results.

4.3 MODEL DESCRIPTION

The challenge for researchers and introducing the mathematical formulated model in depth is in this section. Finally, the formulated model is provided together with a full explanation of requirement of all sets, parameters, variables, constraints, and the objective function. The model assumptions are also stated.

4.3.1 PROBLEM PRESENTATION

The need for MMU services has recently increased significantly due to increasing cost of the healthcare providers in clinics and hospitals as compared to MMU services. Additionally, the recent event, the covid-19 outbreak required preventative measures, such as social isolation. Without a doubt, hospital services cannot be replaced by MMU services. Patients may, however, frequently require low- to medium-level medical skills for extended periods of time. The elderly, those with long-term illnesses, and people healing from accidents or surgery are examples of these patients. Researchers have recently researched the MMUVRP and incorporated many VRP types and aims into their models, including cost minimization, expert sending consideration, travel consideration, and patient choice consideration. Additionally, writers make various assumptions about various circumstances, such as a single depot, several depots, many cars and carers, and uncertainty regarding journey times and services. As time goes on, MMUVRP develops in complexity to meet the ongoing demands of solving real-world issues. In contrast with other organizations, such waste management, where innovation and technique were used to make sure better effective activities take place, none of these earlier attempts took into account the utilization of techniques necessity for advantages of MMUs and patients(Ramos et al., 2018). Additionally, literature on VRP has no evidence that has been studied that service quality and customer satisfaction levels can be clearly and directly measured. Therefore, the focus of this research is on how MMUVRP and technology are used to quantify the quality of service. Regarding the efficient use of technology, the formulated mathematical model presents a cutting-edge method that determines if patients have normal or serious conditions using heart rate monitors. As a result, the created model offers a multi-objective MMUVRP that seeks to accomplish the following objectives: (1) minimising travel time while taking into account the patient's health (normal or critical) and the route type, (2) while taking into account the patient's health and the kind of route, the vehicle should be moving as quickly as possible between nodes, (3) reducing expenditures incurred as a result of a caregiver's workload deviating from average, (4) By calculating the gap (difference) between the projected and actual level of service, one can reduce the penalty costs resulting from subpar service. As far as our concerned knowledge, the model adds two unique (unconsidered before) components to MMUVRP, namely the use of technology (sensors) and gauging service quality by gauging and reducing the difference between expectation and perception. Additionally, by combining each of the three aspects of sustainability—minimizing travel time (which reduces fuel consumption), using electric vehicles, and minimising deviation from the average workload—which improves working conditions—this research supports its intentions to present a completer and more realistic model. The suggested model fits into the following categories. A carer goes in a vehicle to provide care for a predetermined group of patients who are spread out over a network with a single depot, several cars (carers), and a certain number of nodes (patients). These people may have either a normal condition or a serious condition.

This research proposes a unique technique in VRP by assessing the standard of service using performance-related models in order to create an improved and realistic model. The research of (Khorshidi & Hejazi, 2011) was used to gauge the effectiveness of the MMU services offered, together with internal benchmarks established by authorities in the area, where patients' requirements and expectations are assessed. To determine the level of fulfilment of the specified internal measure, the normalised connection between quality-related dimensions and the internal measure will be computed and applied.

4.3.2 MODEL ASSUMPTIONS

1. Multiple places of destination (individuals to be treated) and a single depot (the starting point).

2. It is assumed that capacity of all the MMUs is same.

3. It is assumed that the patients to be treated have medical situation that can be treated by first aid at mobile medical vans.

4. It is assumed that all the patients are provided with one specialist meeting their requirements

5. The time taken by a healthcare supervisor to treat the patients at any village centre is same.

4.3.3 PROBLEM DEFINITION AND FORMULATION

4.3.3.1 PROBLEM DEFINITION

Now, utilizing the vehicle flow concept, we offer a mathematical formulation for the capacitated VRP. The generalized covering multi-traveling salesman problem is explained on a directed graph G = (V, A) where the set of medical centers with a single depot D, a set of villages P, and a set of location centers (where patients are to be treated) Q. Set of arcs are defined for the links between depot and location centers and it is given by $A = \{(q, r)|q, r \in Q \cup \{D\}\}$. Each arc (q, r) is linked with a distance dqr. In addition, auxiliary sets are defined between villages and location centers as $B = \{(p, q)|p, q \in Q \cup P\}$and it is having the distance d_{pq}. It is assumed in the problem that the coverage of the village by a location center is dependent on the distance between them. Location center q can serve village p only if the distance d_{pq} between them is less than or equal to a pre-specified coverage distance C. As an alternative solution, a location center q can "cover" a village P if it is within the coverage radius Q of the location center. On a similar note, if a village p is located within a coverage radius R of a location center q then it can be 'covered' by the facility. Also, in the problem, a set of specialist S has been considered who can fulfil the demand of location centers Q. Each village p has a demand of J_{ps} units for a specialist s that is believed to be fulfilled only if at least one location center is within the coverage distance of the village and is visited by a mobile medical unit (MMU) M. It is assumed that a set of MMU M is there having depot as its starting and end point. Hence, the objective of the problem of to determine the optimized route of the MMU with a specialist S such that the demands at each location center are maximized with a constraint that an MMU can travel a particular distance in a day. The pictorial representation of the problem is presented in Figure 4.1. The following notation is used to formulate a strategic programming model for the problem:

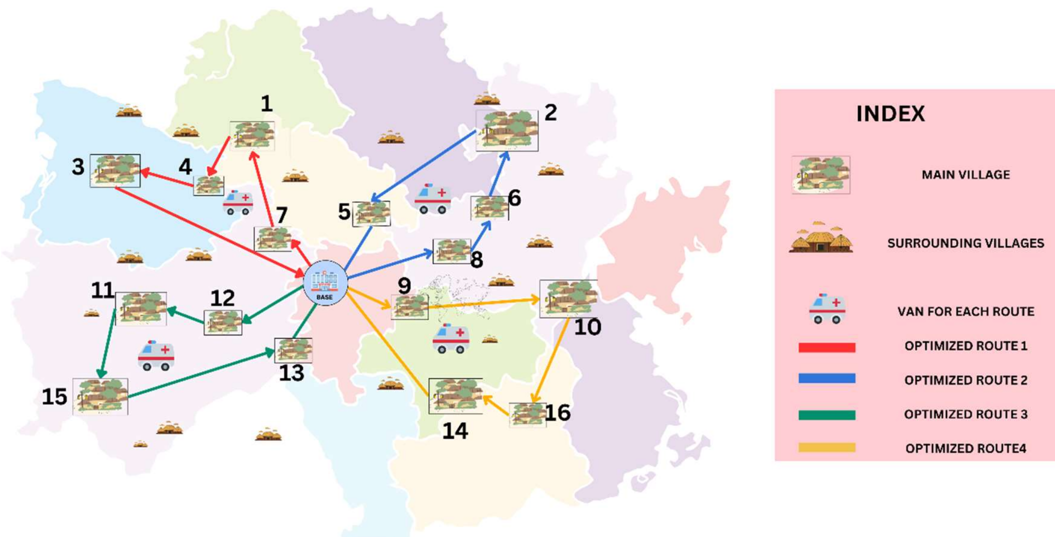

Figure 4.1 Vehicle Routing of Mobile Medical Units with specialist

4.3.3.2 PROBLEM FORMULATION

Sets and Indices

V	Set of all vertices
D	Depot
P = {1, 2,….,m}	Set of villages indexed by *p*

Q = {1, 2,….,n}   Set of location centers by q,r

S = {1, 2,…., s}   Set of specialists indexed by s

M = {1, 2,…, o}   Set of mobile medical units

Parameters

J_{ps}	Demand of the patients in a village at a particular location center to be served by a specialist
d_{pq}	distance between the villages and the location center
d_{qr}	distance between the location center *q* and *r*
D	maximum distance that a mobile medical unit can travel/reach

$$\alpha_{pq} = \begin{cases} 1, & if\ d_{pq} \leq R \\ 0, & Otherwise \end{cases}$$

The problem is modelled using the following variables as:

$$Z_{pqs} = \begin{cases} 1, & if \text{ village } p \text{ is covered by location center } q\ by\ a\ particular\ specialist\ s \\ 0, & Otherwise \end{cases}$$

$$Y_{qms} = \begin{cases} 1, & if \text{ location center } q \text{ is visited by a specialist s mobile medical unit } m \\ 0, & Otherwise \end{cases}$$

$$X_{qrms} = \begin{cases} 1, & if \text{ mobile medical unit m travels from facility q to r by specialist s} \\ 0, & Otherwise \end{cases}$$

$$v_{ms} = \begin{cases} 1, & \textit{if} \text{ specialist s is assigned to mobile unit m} \\ 0, & \textit{Otherwise} \end{cases}$$

U_q *and* U_s = variable to define sub-tour elimination condition

Model

Objective 1 Maximize $$\sum_{p\in P} \sum_{q\in Q} \sum_{s\in S} z_{pqs} J_{ps} \quad (1)$$

Subject to

$$\sum_{r\in Q} x_{\text{qrms}} = \text{y}_{\text{qms}} \quad \forall \text{q} \in \text{Q} \cup \{\text{D}\},\ \forall \text{m} \in \text{M}, s \in S \quad (2)$$

$$\sum_{r\in Q} x_{\text{rqms}} = \text{y}_{\text{qms}} \quad \forall \text{q} \in \text{Q} \cup \{\text{D}\},\ \forall \text{m} \in \text{M}, s \in S \quad (3)$$

$$x_{\text{qrms}} = v_{ms},\ \forall \text{q} \in \text{Q}, \forall \text{m} \in \text{M}, s \in S \quad (4)$$

$$\text{y}_{\text{qms}} \leq v_{ms} \quad \forall \text{q} \in \text{Q}, \forall \text{m} \in \text{M}, s \in S \quad (5)$$

$$\sum_m v_{ms} \leq 1, \quad \forall s \in S \quad (6)$$

$$\sum_s v_{ms} \leq 1, \quad \forall m \in M \quad (7)$$

$$Z_{pqs} \leq \sum_{m\in M} \alpha_{pq} y_{qms} \quad \forall p \in D, \forall q \in Q, s \in S \quad (8)$$

$$\sum_m \sum_s y_{qms} \leq 1, \forall q \in \text{Q} \quad (9)$$

$$\sum_{q\in Q} \sum_{s\in S} Z_{pqs} \leq 1 \quad \forall p \in P \quad (10)$$

$$\sum_{q\in Q\cup\{D\}} \sum_{r\in Q\cup\{D\}} d_{qr} X_{qrms} \leq TD \quad \forall \text{m} \in \text{M}, s \in S \quad (11)$$

$$\text{U}_\text{q} - \text{U}_\text{r} + \text{n}\ \text{X}_{\text{qrms}} \leq (\text{n-1}) \quad \forall q,r \in \text{Q}, \forall \text{m} \in \text{M},\ s \in S \quad (12)$$

$$\text{Z}_{\text{pq}} \in \{0, 1\} \quad \forall p \in \text{P}, \forall q \in \text{Q} \quad (13)$$

$$\text{Y}_{\text{qm}} \in \{0, 1\} \quad \forall q \in \text{Q}, m \in M \quad (14)$$

$$X_{qrm} \in \{0, 1\} \quad \forall q, r \in Q, \forall m \in M \tag{15}$$

$$U_q, U_r \geq 0 \quad \forall q, r \in Q \cup \{D\} \tag{16}$$

Where

In objective function (1), the goal is to maximize patient demand in all villages that can be covered from location centers by all available specialists and mobile medical units.

Constraints (2) and (3) define moving in-arc and out-arc to and from a location center q by mobile medical unit m by specialist s, respectively, thus ensuring one location center is served by only one mobile medical unit and a specialist required at that location center.

Constraint (4) explains that every specialist should be assigned a mobile medical unit

Constraint (5) explains that the number of locations to be visited by a specialist should be less thane qual to the number of specialists assigned in the mobile medical unit

Constraints (6) and (7) say that one specialist is assigned to only one mobile medical unit and vice-versa.

Constraint (8) explains that a village can be served by a particular specialist and mobile medical unit only and only if the location center is visited by at least one specialist and mobile medical unit.

Constraint (9) stated that the location center is visited only by a particular specialist of a mobile medical unit.

Constraint (10) ensures that patients in a village are treated only by a particular specialist of a mobile medical unit, in case of a village lying within the coverage distance of two different location centers.

Constraint (11) ensures that the mobile medical unit covers a distance that is less than or equal to the maximum distance allowed.

Constraint (12) is sub tour elimination constraint.

And the remaining constraints (13) to (16) explain the nature of different variables

This model, also known as a distance-constrained covering multi-traveling salesman problem, can be thought of as a variation of the generalized covering traveling salesman problem. While adhering to the constraint on the maximum driving distance for each salesman, we are trying to increase the amount of demand that can be satisfied by m-salesmen visiting the facilities. However, for any given problem, there is a chance to have additional optimal solutions utilizing the same optimal value (maximum covered demand) but various salesmen's routes or cumulative journey distances.

Objective 2 One of the major objectives of MMUs is to cover maximum demand within the given range of time. So, to accomplish the same, the below sets, indices, constraints to added in addition to the above formulations

$T = [t_{pq}]$ (18)

Set (18) represents the matrices associated with V (set of arcs) of Travel Time

$ST = (st_{1s}, st_{2s}, \ldots.., st_{ps})$ (19)

Indice (19) represents the service time (at centre p) by a specialist s

$t_{pq} = t_{qp}$ (20)

Equation (20) is for symmetry.

$SD = const = \alpha$ (21)

Equation (21) represents the speed which is considered constant for any MMU.

$$TD = \sum_{q \in Q \cup \{D\}} \sum_{r \in Q \cup \{D\}} d_{qr} X_{qrms} \qquad \forall m \in M, s \in S \qquad (22)$$

Equation (22) represents the total distance travelled by a MMU

$$TT = TD/\alpha \qquad (23)$$

Equation (23) represents the total time for travelling distance TD

Hence, the **objective 2** of covering maximum demand in the given time period is as represented by equation (24):

$$\sum_{p \in P} \sum_{q \in Q} \sum_{s \in S} \frac{(d_{qr} X_{qrms})}{\alpha} + st_{ps} \qquad (24)$$

To represent and justify our model, we first validated the results using the random data generated and then with the information provided by Symbiosis University Hospital & Research Center.

4.4 META-HEURISTIC APPROACH (MULTI-OBJECTIVE PROBLEM)

4.4.1 NON-DOMINATED SORTING ANT COLONY OPTIMIZATION (NS-ACO)

As was noted in earlier chapters, it is impossible to produce answers for the suggested model using accurate approaches because of the difficulty of the current issue. As a result, a metaheuristic technique which incorporates and is based on the Ant Colony Optimisation algorithm (ACO) and non-dominated sorting (NDS) approach is employed in this research as an approximation optimisation methodology. The suggested Non-dominated Sorting Ant Colony Optimisation (NS-ACO) method was developed in response to the requirement for a metaheuristic algorithm that addresses both the multi-objective functions and the dynamism of the proposed model. When compared to exact optimisation techniques that result in globally optimal solutions, such an algorithm will produce solutions that are close to optimal[Rader, 2013]. Numerous studies in the literature have demonstrated that the NS-ACO algorithm is appropriate for dealing multi-objective complex NP-hard optimisation problems, including VRP [Bagherinejad & Dehghani, 2016, Gupta & Garg 2017, Kalhor et al. 2011]. The nature of the presented MMUVRP model, as discussed in earlier chapters, necessitates a decision to be made regarding which node to explore following past node. This decision is made based on the health of the patient, which is constantly communicated by the used sensors. The suggested approach must also take into account the existence of multi-objective functions; therefore, this is not the only problem that has to be resolved. As a result, ACO is used to address the problem of variable routing of vehicles that occurs in two circumstances. The first is when the intended routing path must be changed based on the patients' medical status (normal or critical), and the other instance is related to the demand of the patients to choose whether or not to use a charging station. To show the Pareto front solutions, the NDS approach is utilized to identify and order the best solutions produced by the ACO algorithm. A personal laptop with Windows 10 operating system, a 3.00 GHz CPU, an Intel i5 processor, and 8.00 GB of RAM was used to write the suggested NS-ACO algorithm.

4.4.2 ANT COLONY OPTIMIZATION (ACO) ALGORITHM

One of the optimization methods that adheres to the swarm intelligence optimization approach is ACO. The observed behavior of insects like bees and ants served as the basis for the development of swarm intelligence optimization techniques. Scientists have created algorithms that use simulations of the social organization and behaviors of insects to tackle complicated real-world issues as a result of the unexpected social coordination displayed by these insects. As a result, ACO is a meta-heuristic optimisation technique that offers solutions for complicated problems based on and helped by ant behaviour, in terms of their cooperative behaviour and travel out of their nest (colony) to their sources of food via several distinctive pathways. As a result, ACO is a meta-heuristic optimisation technique that offers solutions for complicated problems based on and helped by ant behaviour, in terms of their cooperative behaviour and travel out of their nest (colony) to their sources of food via several distinctive pathways. Ants in the actual world have a special way of moving from their nest to their source of food and back, but they are unable to communicate with one another and organise their movements. Each ant departs the nest to travel one of many paths to the food source. Each ant chooses at random which path to take, and they do this at every intersection until they reach food. According to logic, each route has a distinct travel distance from other routes, and each ant travels a different distance from other ants. Ants leave behind a chemical compound known as pheromones as they are moving, which they utilise to communicate with one another and choose the best route. As previously mentioned, ants choose their routes at random, but when they detect pheromones, they will take the route that contains them. As this process continues, shorter (faster) routes will contain the most pheromones because ants will use them more frequently than other ants taking routes that are longer. Due to the high concentration of pheromones in particular pathways, the likelihood that further ants will follow the same route grows as the total number of ants travelling it increases. This particular type of insects uses pheromone-based communication in order to not only find food but also overcome challenges whenever an obstacle appears in their path. Additionally, pheromones disappear as the likelihood that ants will choose the longer route declines; as a result, the concentration of pheromones along this route falls and any already present pheromones vanish. Nevertheless, the arbitrary route selections continue, albeit with less likelihood. Ants are able to live because to their haphazard selection of paths, which also aids them in locating and traversing alternate routes when barriers arise. The cornerstone of ACO is its capacity to address complicated real-world situations, particularly those in which each node in the routing plan has a different relevance from other nodes and needs to be given a greater likelihood of being visited by the vehicle (in the case of VRP). The PhD dissertation of Dorigo (1992) provided the first use of ACO in VRP, and throughout time the algorithm has undergone many and continual improvements, according to Bell and McMullen [Liu et al., 2019 Zheng et al. 2020]. Figure 4.2 depicts the ACO algorithm flowchart from Khanna et al. (2015)'s research.

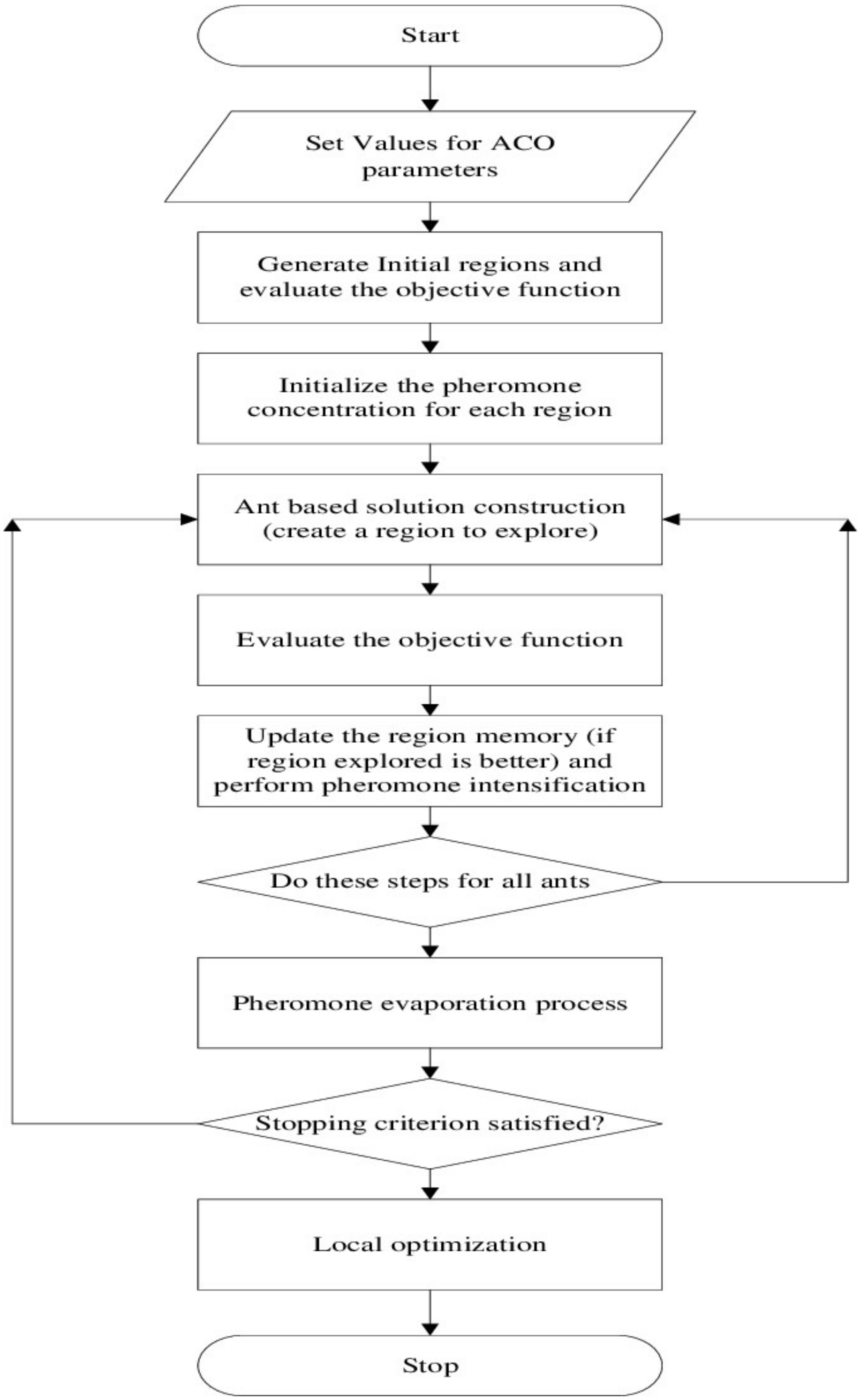

Figure 4.2: ACO algorithm flowchart (Khanna et al., 2015)

4.4.3 SOLUTION APPROACH USING VARIABLE NEIGHBORHOOD SEARCH

The Variable Neighbourhood Search (VNS) method, developed by Mladenovi'c and

Hansen, is a metaheuristic for resolving combinatorial optimisation issues. Its central concept is the systematic modification of a local search's neighbourhood. The two primary components of the VNS algorithm are a shaking phase (Diversification) to break out of local optima and a local exploration phase (Intensification) to look for an improvement in the immediate vicinity of the present solution using specified neighbourhood structures. It makes advantage of the next discoveries as:

- A global minimum is a local minimum with regard to all neighbourhoods that are conceivable.
- A local minimum with respect to one neighbourhood is not always an optimum minimum with respect to another.
- A variation of the VNS known as the Variable Neighbourhood Descent (VND) performs the neighbourhood structures in a deterministic manner.

The solution in the paper utilizes the step functioning of Variable Neighborhood Descent (VND), which is a variant of VNS to have a better solution s possible and to search beyond the reach of VNS.

4.4.4 MACHINE LEARNING APPROACH

In this section, we explain our strategy for dealing with the multi-objective optimization of the distance-constrained covering multi-traveling salesman problem and outline its level of complexity as a whole. We provide the Timeline technique for the time frames and halt intervals and also the cost function with which we approach the multi-objective problem as well as our two-stage strategy. The two-stage technique divides the solution space into two distinct issues, hence reducing the complexity of the optimization mechanisms as a whole. Although we cannot be certain, we assume that our technique does not have a negative effect on optimality, particularly since the used meta-heuristics are already suboptimal.

Machine Learning has been categorized into three different categories. One of them is Reinforcement Learning (RL) which is based on goal-directed learning. When the machine must make decisions based on an experimental assumption and learns from its actions and prior experiences, RL is used. In this paper, the multi-traveling salesman problem belongs to the nondeterministic polynomial hard (NP-hard) problem. RL helps in searching for the route that is having maximum reward over time.

RL is the decision-making of science to learn the optimal behavior to obtain maximum outcome rewards. It is the optimal behavior learned through interaction with the environment and observing how to respond to it.

Some important elements of the RL System are The learner, the environment, the results, or the outcomes that the agent will get upon taking action. The RL algorithm can be classified as model-free and model-based. An explicit environment is not created by the model-free method. and can be treated as a trial-and-error method in which the algorithm runs with the help of the environment and the actions taken and gives the most optimal policy out of it.

The action critic algorithm, which essentially combines two approaches—value-based and policy-based—is the most efficient RL algorithm. This helps to enable the efficient use of data with consistent convergence through the employment of both algorithms.

Example of RL: Autonomous driving: - In an uncertain environment, an autonomous vehicle must execute numerous tasks including perception, decision-making, and planning. Vehicle route planning and motion prediction are some of the specific tasks where RL is applied. To make judgments at various temporal and geographical scales, vehicle route planning involves several low-level and high-level policies.

Think about a situation where the environment is kept in a certain state but changes based on the actions made. In RL, the environment is kept in a certain state but changes based on the actions made. We have an agent that interacts with this environment; it sequentially chooses actions and receives input on how well or poorly the new state is after each action is made. RL aims at finding a method for the agent to choose actions based on the present state that, on average, results in favourable states.

4.5 DATASET

We have first got the data randomly generated and then tested the model. The accuracy of the model turned out to be 99% and 100% for different algorithm. This proved the potency of the model and so we also tested the model with one month data set from real world scenario.

We thank SUHRC for providing data for one month for our result implication. So, in the paper, the current implementation divides the villages into ones. Each zones covers 4-5 villages which are in a 5 km area proximity. Here:

Table 4.1 Notations

V1..V4	Villages covered
Data is of the distance from the base village (we will call it zone)	
S	Specialist
P	Number of patients attended (as per the specialist 1..6)
Z	Zone

We have collected data date wise. We have covered 20 villages in a day through total 4 vans and 1 van is covering 5 villages (can be more also). The sample data would look like this:

Table 4.2 Sample Data

Village	X	Y	Distance from Depot	V1	V2	V3	V4	Van sent
1	20	30	5	4	3	5	2	1

S1	S2	S3	S4	S5	S6
Eye Specialist	Gynae	General Physician	Pediatrician	ENT	Pulmonologist

Dispatch time	In time	Out Time	P1	P2	P3	P4	P5	P6	Total	Zone
10:00	10:15	17:00	85.00	25.00	85.00	35.00	25.00	52.00	307.00	Z1

In the sample data X, Y is the coordinate of village 1. The distance of the village is 5 KM from the base station. The villages around it are V1..V4. Van v1 is sent to village 1. Each van would have specialists (s1…s6), not all the villages need to have all the specialists (this sample is just the example). In implementation they will have binary values 0 or 1. 1 is for True and 0 for false. Then we have patients attended P1…P4 (P1 is the number of patients attended by each specialist). Then we total the number of patients who attended and then we allot it a zone number.

4.5.1 IMPORTING LIBRARIES

```
In [15]: pip install deap

Collecting deap
  Downloading deap-1.3.3-cp39-cp39-win_amd64.whl (114 kB)
     ------------------------------------ 114.3/114.3 kB 317.3 kB/s eta 0:00:00
Requirement already satisfied: numpy in c:\users\predator\anaconda3\lib\site-packages (from deap) (1.21.5)
Installing collected packages: deap
Successfully installed deap-1.3.3
Note: you may need to restart the kernel to use updated packages.
```

```
In [43]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
```

```
In [44]: import random
         from deap import base, creator, tools, algorithms, benchmarks
         from deap.benchmarks.tools import diversity, convergence, hypervolume
         import matplotlib.pyplot as plt
```

```
In [2]: from ortools.constraint_solver import routing_enums_pb2
        from ortools.constraint_solver import pywrapcp
```

Importing dataset

Firstly we import the dataset using pd.read_csv(). Then we call the data using .head() function. This function shows the first 5 columns of the dataset.

4.5.2 DATA CLEANING

For data cleaning, we first see if the data is duplicated or if there are any **null** or **NA** values to be replaced or removed. We use the **df.isnull()** function for finding the null values and since there are no null values, we proceed to data exploration.

```
dataset = par                                   ")
x=dataset
```

```
In [52]: x.isnull().sum()

Out[52]: village                                   0
         lat                                       0
         lng                                       0
         Distance (symbiosis medical college)      0
         van                                       0
         Patient of weakness and general issues    0
         fever                                     0
         Cardio Vascular Sytem                     0
         Respiratory org system                    0
         ENT                                       0
         eye                                       0
         git                                       0
         Genito Urinary System                     0
         Injury / Burns / Trauma                   0
         Musculo Skeleton / Bones, Joints          0
         skin                                      0
         dental                                    0
         mental                                    0
         Neurological                              0
         Obstetric                                 0
         Other                                     0
         dtype: int64
```

4.5.3 DATA EXPLORATION

For the data exploration part, we use **df.info()** to get the summary of the dataset. In addition to column numbers, column labels, column data types, and memory usage, it also includes the number of cells in each column (non-null values).

```
In [6]: dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 21 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   village                               32 non-null     object
 1   lat                                   32 non-null     float64
 2   lng                                   32 non-null     float64
 3   Distance (symbiosis medical college)  32 non-null     int64
 4   van                                   32 non-null     int64
 5   Patient of weakness and general issues 32 non-null     int64
 6   fever                                 32 non-null     int64
 7   Cardio Vascular Sytem                 32 non-null     int64
 8   Respiratory org system                32 non-null     int64
 9   ENT                                   32 non-null     int64
 10  eye                                   32 non-null     int64
 11  git                                   32 non-null     int64
 12  Genito Urinary System                 32 non-null     int64
 13  Injury / Burns / Trauma               32 non-null     int64
 14  Musculo Skeleton / Bones, Joints      32 non-null     int64
 15  skin                                  32 non-null     int64
 16  dental                                32 non-null     int64
 17  mental                                32 non-null     int64
 18  Neurological                          32 non-null     int64
 19  Obstetric                             32 non-null     int64
 20  Other                                 32 non-null     int64
dtypes: float64(2), int64(18), object(1)
memory usage: 5.4+ KB
```

We also use **df.describe()** for the statistical description of the dataset. For any numerical data, it includes count, mean, standard deviation, minimum, 1st quartile, 2nd quartile, 3rd quartile, and maximum.

```
In [5]: dataset.describe()
Out[5]:
```

	lat	lng	Distance (symbiosis medical college)	van	Patient of weakness and general issues	fever	Cardio Vascular Sytem	Respiratory org system	ENT	eye	git	Genito Urinary System	Injury / Burns / Trauma	Musculo Skeleton / Bones, Joints
count	32.000000	32.000000	32.000000	32.0	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000
mean	18.517866	73.606059	22.656250	1.0	12.031250	1.062500	0.031250	24.562500	0.218750	0.281250	5.531250	0.218750	0.906250	6.375000
std	0.137473	0.117475	7.554573	0.0	14.523583	1.162242	0.176777	28.238543	0.490844	0.581121	6.010659	0.659148	1.279097	8.023152
min	18.248400	73.270600	5.000000	1.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	18.455500	73.561375	17.750000	1.0	1.750000	0.000000	0.000000	5.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000
50%	18.508100	73.598400	23.000000	1.0	7.000000	1.000000	0.000000	14.500000	0.000000	0.000000	4.000000	0.000000	0.000000	3.000000
75%	18.551825	73.676725	28.000000	1.0	14.250000	1.000000	0.000000	31.500000	0.000000	0.000000	8.000000	0.000000	1.000000	10.000000
max	19.114600	73.877300	33.000000	1.0	55.000000	5.000000	1.000000	95.000000	2.000000	2.000000	27.000000	3.000000	5.000000	28.000000

The solution after calling the above dataset is presented below. It presents, explains and discusses the results that were reached after solving the suggested MMUVRP. The numerical information used to solve the model is provided first. The acquired results are then displayed, including the route map, cost formulas, workload, and service quality.

4.5 RESULTS AND DISCUSSION

The computational results that were obtained after the designed model was solved are presented in this section. The model, which was created arbitrarily using the GUROBI solver, was solved using hypothetical data. Real-world scenarios were tried to test and confirm the model's solvability once the hypothetical data was validated. Results were first presented using small instances, then subsequently in this chapter, medium- to large-sized examples were demonstrated.

4.5.4 DATA INTERPRETATION

The data required to test the model formulated is generated randomly first and then numerical data of last one month is obtained from SUHRC, Lavale, Pune. This data is used for testing the model and it will eventually help in increasing the robustness of the model. In table 4.1, represents the numerical data of village centers where MMU will visit, demand of patients in that particular node, specialists visiting the village centre. Last two column represents the solution of our model's objective, viz. objective 1 showing the total demand covered by MMU with specialist and objective 2 representing the demand covered in the given time period.

4.5.5 COMPUTATIONAL RESULTS

The general VNS algorithm introduced earlier is used to compute for one-month instances. A variation of the VNS known as the Variable Neighbourhood Descent (VND) performs the neighbourhood structures in a deterministic manner. Various columns are introduced in the table representing "Sr No.", "#Village centres", "#Patients", "Specialist", and "Maximum distance". It is seen in Table below that GUROBI has given results for all the different cases as the restricted time limit.

Table 4.3 Summary of problem instances

Sr. No.	Village Centres	Patients (max capacity considered)	Specialists	Objective 1	Objective 2
1	32	233	1	221	230
2	22	322	2	302	318
3	34	245	1	220	232
4	24	236	2	220	230
5	18	298	3	290	290
6	13	302	3	300	280

7	15	301	1	286	287
8	21	277	1	222	265
9	28	295	2	289	289
10	7	265	1	256	261

We also used different data visualization and data analyzing tools like Anaconda, R studio and Power BI for better understanding of our data and to get a clear idea of our objectives. To illustrate the mathematical model discussed above in section 4.3, we present case problem of SUHRC with 5 different specialists in different villages around SUHRC. The results of different routes taken and total distance travelled are summarized in different graphs as below:

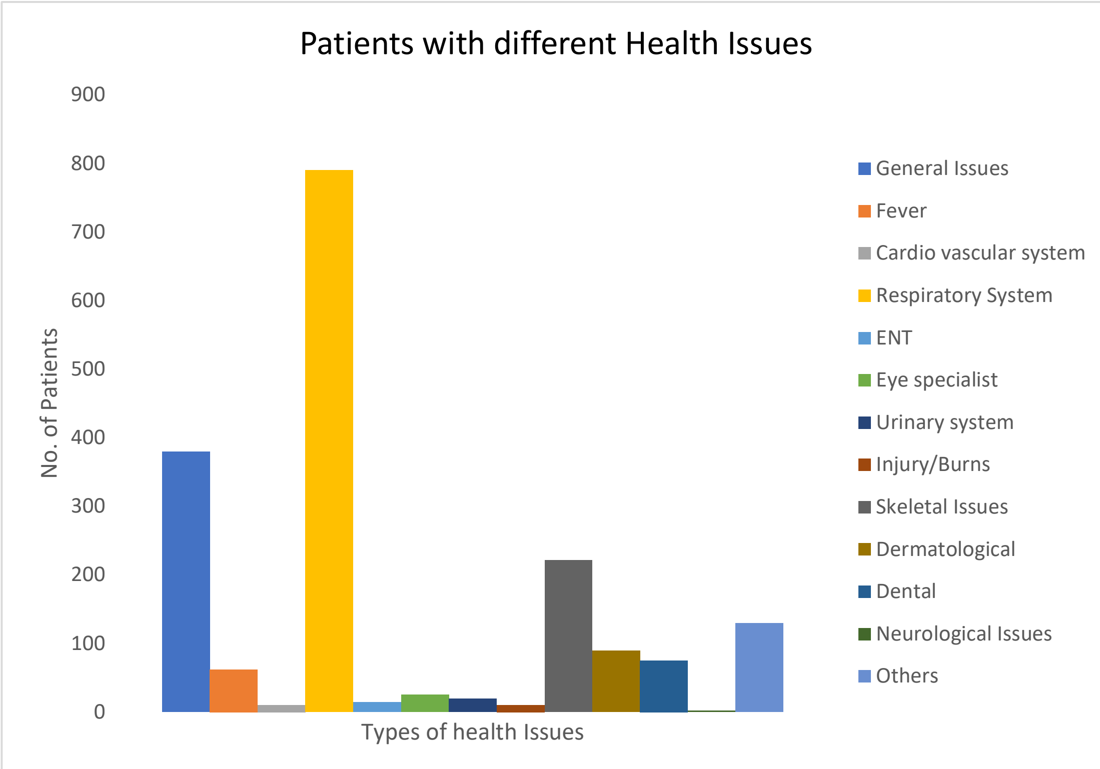

Figure 4.3 Health issues of patients

The Figure 4.3 bar graph shows the no. of patients for each category of health issue. We can see that the no. of patients with respiratory issues is around 786 and for general health issues is 385.

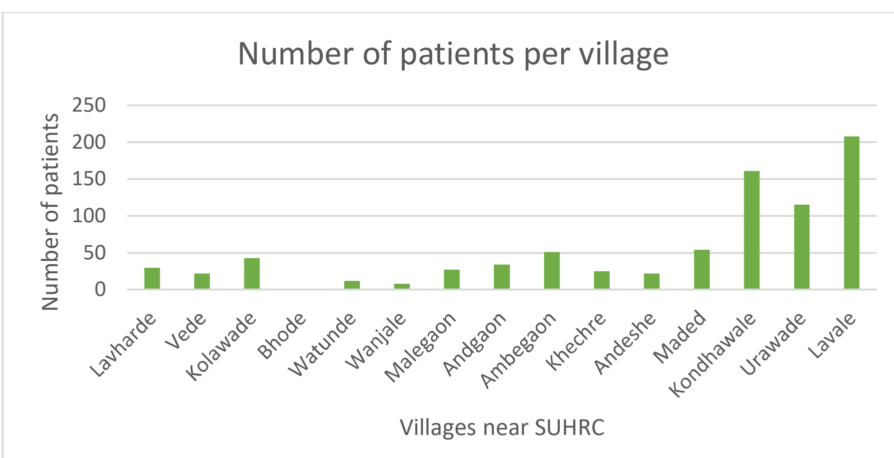

Figure 4.4 Number of patients per village

The Figure 4.4 shows the number of patients in each village. Max no of patients are in the village Lavale(208 patients), followed by Mulkhed (201 patients).

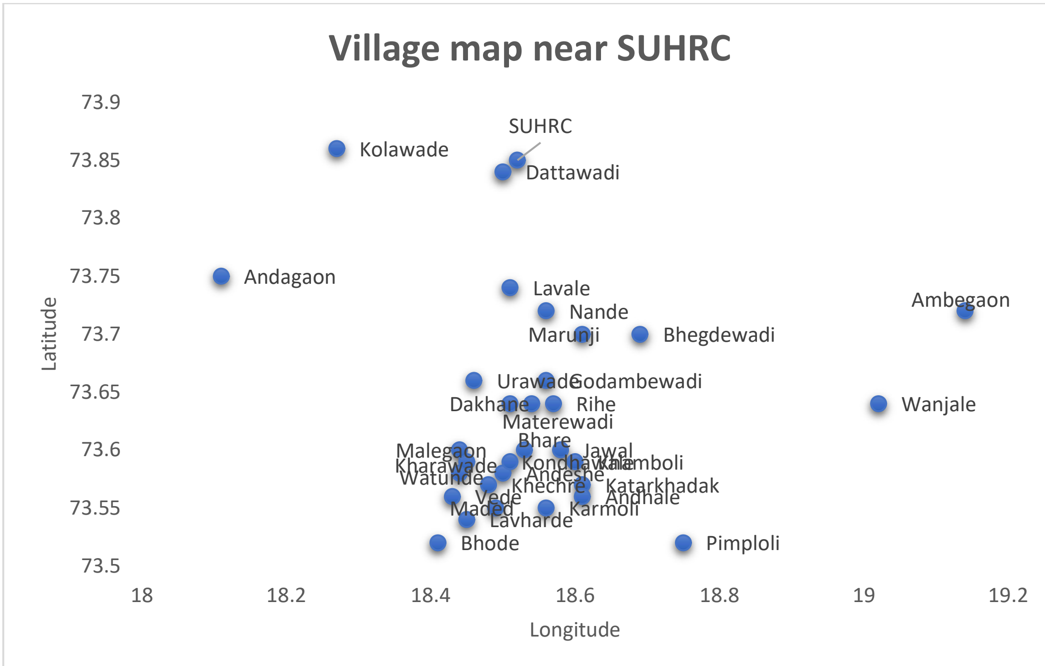

Figure 4.5 Scatter plot for different villages around SUHRC

The Figure 4.5 shows the scatter plotting of all the villages the MHUs travel using the village coordinates.

```
Route for day 0:
 0 ->  9 ->  4 ->  2 ->  5 ->  6 ->  3 ->  1 -> 0
Distance of the route: 73km

Route for day 1:
 0 ->  30 ->  10 ->  7 ->  8 ->  15 -> 0
Distance of the route: 68km

Route for day 2:
 0 ->  18 ->  27 ->  28 -> 0
Distance of the route: 62km

Route for day 3:
 0 ->  32 ->  31 ->  26 ->  23 ->  21 ->  22 ->  25 ->  24 ->  20 ->  29 ->  16 -> 0
Distance of the route: 71km

Route for day 4:
 0 ->  19 ->  14 ->  13 ->  12 ->  11 ->  17 -> 0
Distance of the route: 72km
```

Figure 4.6 Optimal Routes of Villages

Table 4.4 Number indicating villages in Figure 10

Village	Number	Village	Number
Andagaon	1	Bhegdewadi	17
Watunde	2	Dakhane	18
Kharawade	3	Karmoli	19
Kolawade	4	Bhare	20
Bhode	5	Khamboli	21
Vede	6	Katarkhadak	22
Wanjale	7	Rihe	23
Malegaon	8	Jawal	24
Lavharde	9	Andhale	25
Ambegaon	10	Pimploli	26
Khechre	11	Godambewadi	27
Andeshe	12	Marunji	28
Maded	13	Lavale	29
Kondhawade	14	Dattawadi	30
Urawade	15	Materewadi	31

Pirangut	16	Nande	32

The Figure 4.6 shows the optimal route for the number of villages we have been provided with, where 0 is the depot, ie, SUHRC, and 1,2,3, and so on are the villages represented in Table 4.2, where the MMUs travels for the regular checkup. Optimized diagram for different routes in Figure 4.7 is plotted below:

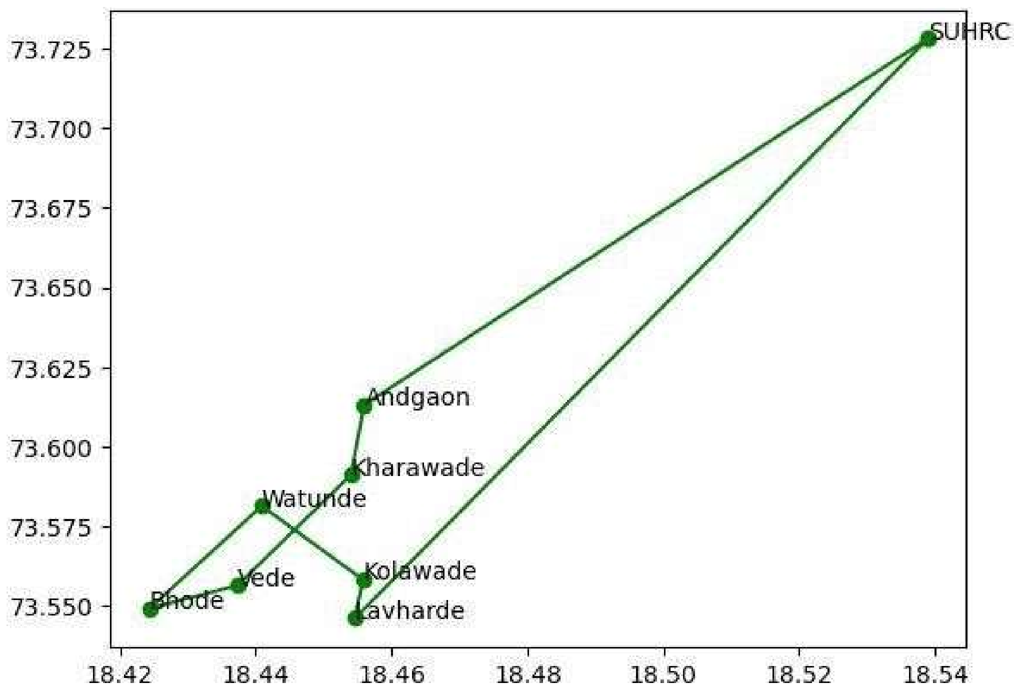

Figure 4.7 Route for MMU for Day 0

Route for vehicle for day 0 is as represented in Figure 4.7 is:

SUHRC → Lavharde→ Kolawade →Watunde→ Bhode→ Vede→ Kharawade→ Andgaon→ SUHRC

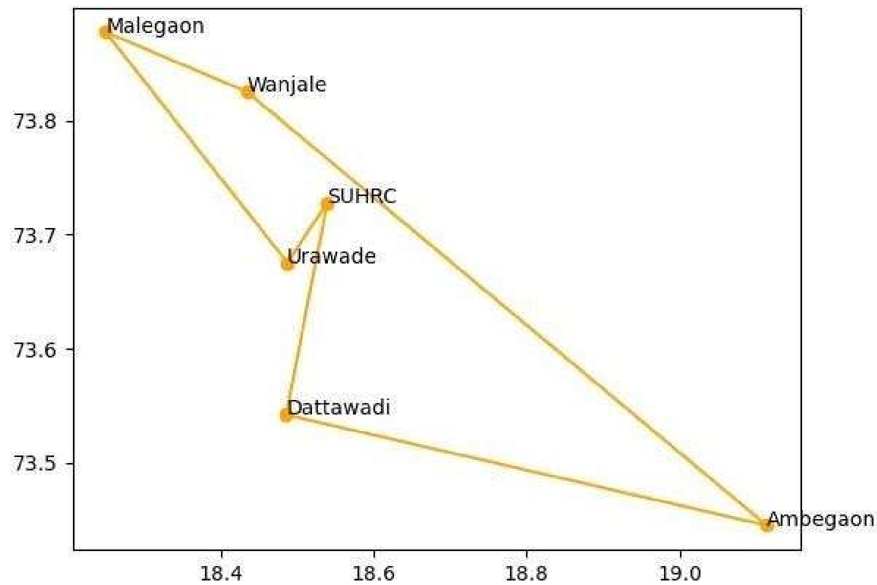

Figure 4.8 Route for MMU for Day 1

Figure 4.8 shows the Network diagram for MMU for Day 1 according to the coordinates taken from google maps.

Route for Day 1:

SUHRC →Dattawadi →Ambegaon →Wanjale →Malegaon →Urawade →SUHRC

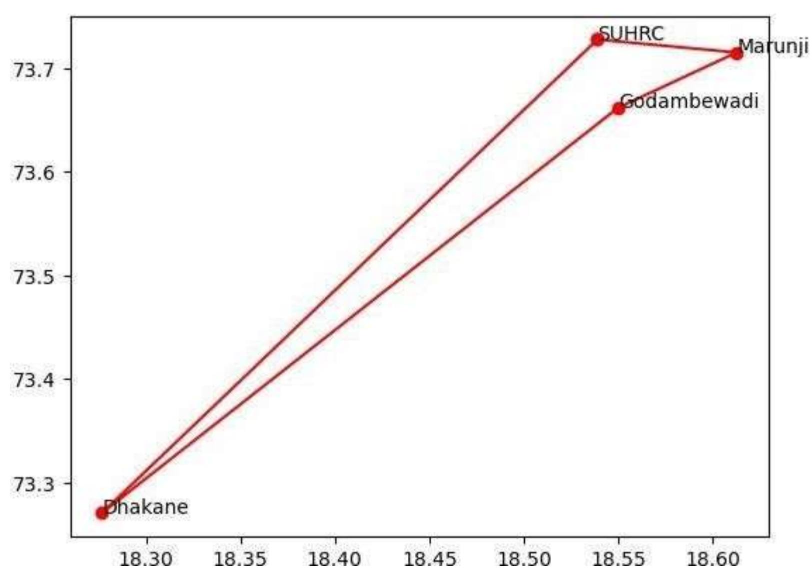

Figure 4.9 Route for MMU for Day 2

Figure 4.9 shows the Network diagram of the Route for MMU for Day 2 according to the coordinates taken from google maps.

Route for Day 2:

SUHRC →Dakhane →Godambewadi →Marunji →SUHRC

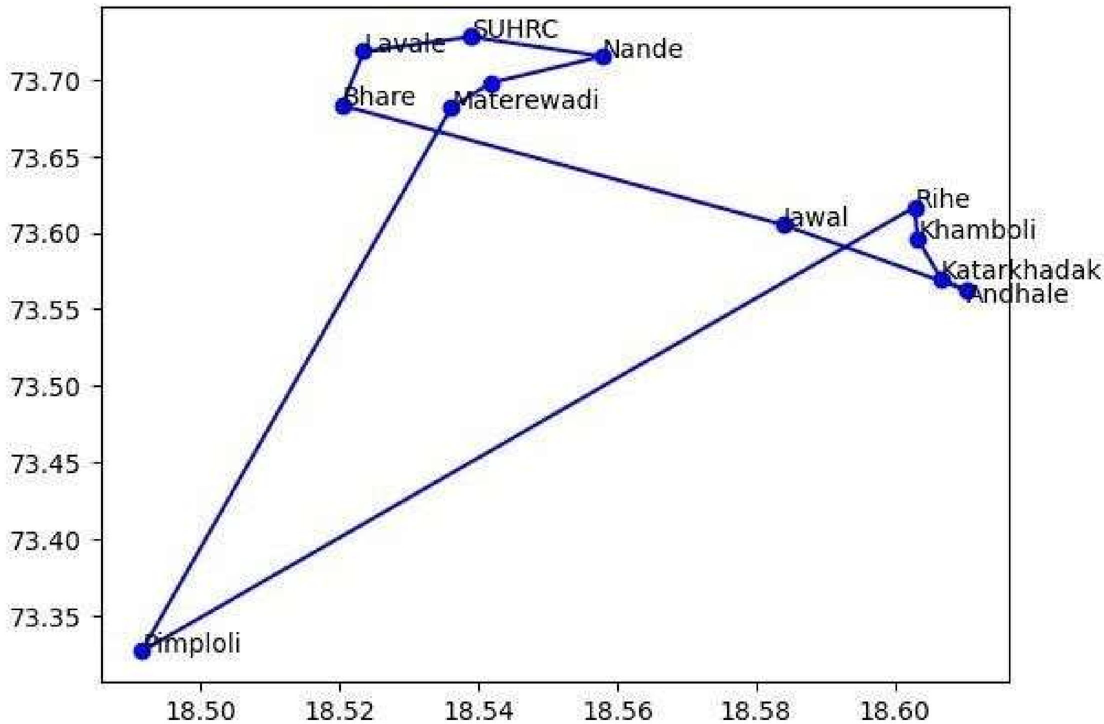

Figure 4.10 Route for MMU for Day 3

Figure 4.10 shows the Network diagram of the Route for MMU for Day 3 according to the coordinates taken from google maps.

Route for Day 3:

SUHRC → Nande → Materewadi → Pirangut → Pimploli → Rihe → Khamboli→ Katarkhadak → Andhale → Jawal →Bhare → Lavale → SUHRC

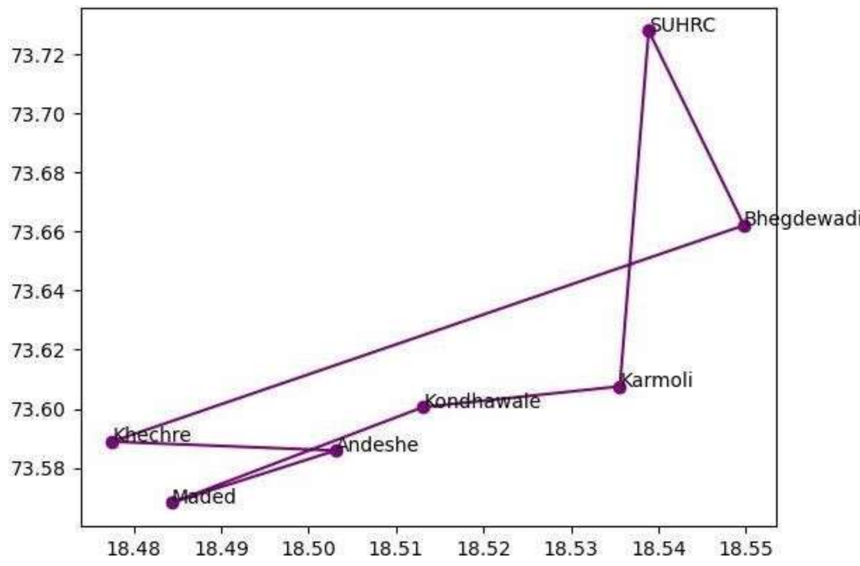

Figure 4.11 Route of MMU for Day 4

Figure 4.11 shows the network diagram of Route by MMU for Day 4 according to the coordinates taken from google maps.

Route for Day 4:

SUHRC → Karmoli → Kondhawale → Maded → Andeshe → Khechre → Bhegdewadi → SUHRC

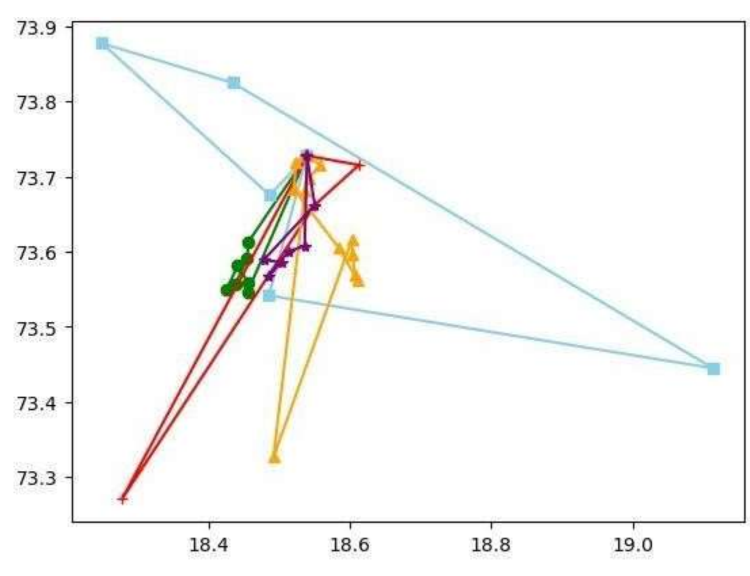

Figure 4.12 Optimal route of MMU

The above Figure 4.12 shows the complete Network diagram for the optimal route which we found out using the algorithm according to the coordinates taken from google maps.

To justify the results obtained from algorithm, classification algorithm of Machine Learning is used to calculate the category of a given dataset, and these algorithms are primarily employed to forecast the results for categorical data.

The basic condition of each of these algorithms is that they work after learning from the past data which is presented in the date wise format. Also, machine learning works on huge dataset, it should learn from records of at least few months dataset. In the dataset, we treat every column as a feature of data, as explained in Table 1. The different machine learning algorithms learn from the data we feed. For examples, Random Forest builds decision trees on different samples and takes their majority vote for classification and average in case of regression. Also, it can handle continuous as well as categorial data, which is its one of the important features. So, the working of this whole process is explained below:

Step 1: Selecting random x points from the training data set.

Step 2: Creating the decision trees linked to the chosen data points (subsets)

Step 3: Choosing appropriate number "N" for building required decision tree.

Step 4: Repeat Step 1 and Step 2

Step 5: Find each decision tree's forecasts for any new data points, then arrange them in the category that receives the most votes.

The accuracy obtained by using these steps is almost 95% for different machine learning algorithms for randomly generated data.

The different algorithms used in Figure 17 are:

KNearest Neighbor (KNN)- It is a Supervised Machine Learning Algorithm that classifies data based on similarity, assigns new data points, and saves existing data.

Logistic Regression (LR)- A supervised learning model is a well-known machine learning technique that forecasts categorical dependent variables using a predetermined set of independent factors.

Linear Discriminant Analysis (LDA)- Linear discriminant analysis is one of the frequently used dimensionality reduction techniques in machine learning to address problems involving more than two classes.

Decision Tree Classifier (DTC)- A decision tree consists of two nodes: a Decision Node and a Leaf Node. In contrast to Leaf nodes, which are the results of decisions, Decision nodes are created by making decisions and have many branches.

Gaussian NB (GNB)- Two nodes—the Decision Node and the Leaf Node—make up a decision tree. A choice is made using a Decision node, which has several branches, whereas a Leaf node represents the result of that decision and does not have any more branches.

Support vector Machine (SVM) - By defining the optimal line or boundary in n-dimensional space to split n-dimensional space into classes in the future, the SVM algorithm can categorize fresh data points quickly. Using a hyperplane as a boundary is the best choice.

Random Forest (RF)- The most often used algorithm is called "Random Forest." This method's goal is to merge the findings from many decision trees to get a single conclusion.

The accuracy of different classification algorithms is okay but not good, as the dataset we had was very less (just one month). Though our generated accuracy is much higher in the case of data generated randomly which is approximately coming out near 95% for months of data.

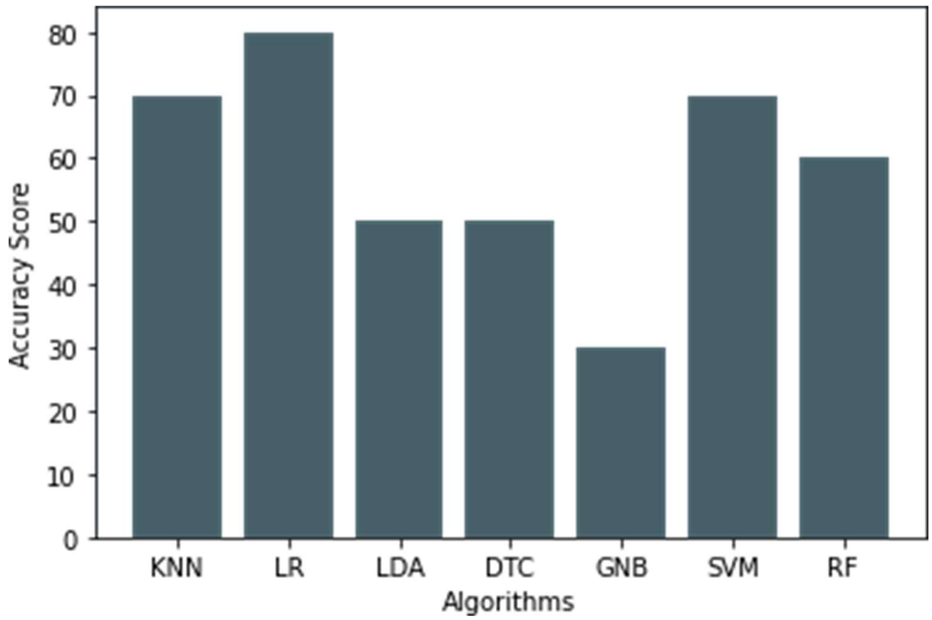

Figure 4.13 Accuracy score of different Classification algorithm heuristics

The accuracy obtained in the figure 4.13 are less as the value of testing data was not sufficient enough to make a decision by these algorithms. The real-world scenario case is taken into the consideration to find out the precise solution and accuracy of the algorithm. Also, not to forget that many assumptions were taken into the considerations to generate the solutions such as:

i. The time taken by a specialist to treat a patient is considered as equal.

ii. All the MMUs are considered as same.

iii. One village centre will be visited by one and only one MMU.

iv. One MMU has one specialist and a team.

It is important to note that Euclidean formula is used to measure the matrix for distance. The people considered are within the boundary of given village and the distance travelled by MMU considered in our problem instance is 80 km/day, i.e., a MMU can travel maximum up to 80 km per day within the range of 100 km.

4.6 SUMMARY

The purpose of this project is to create an optimized route using AI to minimize the distance traveled by MMUs to different underdeveloped village areas for regular check-ups of patients who are not able to get proper health facilities nearby. This project is helpful for society as it's a facility that can solve a lot of health-related problems for people who live in underdeveloped areas.

It also helps society in a way that would increase the overall health of the people who live in rural places, and it will also save their expenses. The main advantage of working on this project is that we get to learn about various algorithms used to optimize routes and how it is implemented using AI. The results from this project can be provided to the hospitals or NGOs where they can implement the algorithm. We are thankful to SUHRC for sharing their data and helping us to validate the algorithm.