IE 685 MSc-PhD Research Project 1 : Study of Map based data for Location Decisions

by

Jayshankar Yadav
Roll No. 22N0460

under the guidance of

Prof. Narayan Rangaraj

Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay
Mumbai 400 076

Contents

List of Figures

Abstract

The location-allocation model finds the best facility location which optimizes the total distance and total cost. In any country or state location of the manufacturer or distribution center is important to fulfill the demand of the people. In this report, there is an approach for allocating the distribution center in Rajasthan. and applied the green field concept as e giving example of hospital allocation. also gives the approach to calculate the distance and time from source to destination.

Chapter 1

Introduction

You have seen lots of times that any country or state buys products from other countries or states. here will be possibility that they are not be able to manufacture products which is necessary for peoples. so to handle this situation there will be one possibly that they can buy products from other states or countries. Let suppose they agree to buy products from the other state or countries. So we have to locate some distribution center which fulfills the demand of the customer. To allot distribution center we have to see that it minimizes the distance, time and cost. As we are planning to allocate new distribution center this situation is called Greenfield. And Greenfield analysis is a technique commonly use during the early stage of supply chain design to find the optimal number and location of the Distribution center. also uses QSRM(Open Source Routing Machine) to find the optimal driving distance and driving time.

Suppose we have to locate the first Hospital one state. So our question is where do we locate? this is also the situation of greenfields. we should optimize the distance and time between hospitals and patients. In this project, we have talk about the Rajasthan state.

Apart from this, here we have use the QGIS(Quantum Geographic Information System) tool. In that, we can find which is the shortest path or Fastest path from one point to another point on the Google map. Also, we can find the Driving time and distance and this result gives that of exactly the present time.

Chapter 2

Problem statements and Data

Mainly we have solved two problems one is related to the Distribution center and second is related to hospital allocation.

for the first problem we have taken a Manufacturer company which is outside Rajasthan taken as Assam [26.2006,92.9376]. From here distribution center will take products. The distribution center will try to fulfill the demand of the city. In the data of the Distribution center we have taken as the Different - different places of different districts of Rajasthan. so we have taken as total 32 places to locate the Distribution center. To locate we have taken the longitude and latitude of the Distribution center. So for this we Have find each longitude and latitude of each Distribution center using google searched each one by one. In this Distribution file also have included the fixed-cost of the each DC. which also randomly taken by us. example in Ajmer the fixed of the Distribution center is 355536005. After that we took the assembly cost. which will for each product transported from the Distribution center to the cities. example Ajmer has 644 on each product transported from Ajmer distribution center to cities. After that In the file we have taken as the $quantity_{\_0}$, $cost_{\_0}$, $quantity_{\_1}$, $cost_{\_1}$, $quantity_{\_2}$ and $cost_{\_2}$. For this also we have randomly generated data but logically. when the quantity is transported from the Distribution center to cities if quantity is less than $quantity_{\_0}$ then only fixed costs get charged. If the quantity is greater than $quantity_{\_0}$ and less than $quantity_{\_1}$ then cost will be $cost_{\_0}$. Similarly when the data quantity is greater than $quantity_{\_1}$ and less than $quantity_{\_2}$ the cost would be $cost_{\_1}$. when quantity is greater than $quantity_{\_2}$ than cost would be $cost_{\_2}$. and cost is measured in units.
Below we have attached some part of the data.

District	Latitude	Longitude	Assembly	Fixed_cost	cost_0	cost_1	cost_2	quantity_0	quantity_2	quantity_1
Ajmer	26.460578	74.641756	768	355536005	706	652	605	200000	300000	250000
Alwar	27.566581	76.610602	732	248486041	727	658	627	200000	300000	250000
Banswara	23.545683	74.448134	641	372439811	718	688	627	200000	300000	250000
Baran	25.10111	76.513669	785	235117850	746	691	608	200000	300000	250000
Barmer	25.743971	71.393059	736	443329973	707	651	633	200000	300000	250000
Bharatpur	27.213921	77.500799	691	418120932	743	679	612	200000	300000	250000
Bhilwara	25.347483	74.637973	782	412088181	716	677	648	200000	300000	250000

Figure 2.1: Data

The second file is the demand file for each cities. So In that file, There are 15008 cities of Rajasthan and there longitude and latitude. This file directly found on google in the form of CSV. In that, we have only added the Demand of each cities. And demand will be generated by us randomly.

This Optimization problem is Integer programming problem. Here we taken all variables as the Integer.

So our aim is find the minimum of number to open Distribution center which satisfy the whole demand.

N: set of all location's of Distribution center

J: set of all preferred location where we can allot Distribution center

K : Number of cities

L: Marginal level capacity of Bracket DC (4)

Parameters:

F_j: Fixed operating quantity of the jth DC

d_k : kth cities demand

t_j : transportation cost from manufacture to DC_j

t_{jk}: Transportation cost from DC_j to $city_k$

a_j : Unit assembly cost of each product at DC_j

c_jl : Operating cost of DC_j at level l

Q_{jl}^h : Maximum capacity of DC_j at level l

Q_{jl}^l : Minimum capacity of DC_j at level l

Variables:

OPC_j: Total Operating cost of DC_j

$I_{jl} = 1$ when DC_j located at level l, 0 otherwise

x_{jl} : Quantity shipped from Manufacturer to DC_j at level l

y_{jk} : Quantity shipped from Distribution center to Cities

Objective:

min $\sum_{j\in J}[OPC_j + \sum_{l\in L}(t_j x_{jl} + a_j x_{jl}) + \sum_{k\in K} t_{jk} y_{jk}]$ + Operating cost of manufacturer

Operating cost of the DC

$F_j + \sum_{p\in l}(Q^h_{jp} - Q^l_{jp})c_{jp} + (x_{jl} - Q^h_{jl})c_{jl} <= OPC_j \forall j \in J, l \in L$

flow balance:

$\sum_{k\in K} y_{jk} = \sum_{l\in L} xjl \ \forall j \in J$

Demand:

$\sum_{j\in J} y_{jk} = d_k \forall k \in K$

DC location Constraint:

$Q^l_{jl} I_{jl} <= x_{jl} <= Q^h_{jl} I_{jl} \forall j \in J, l \in L$

$x_{jl}, y_{jk} \in \{0, 1, 2, ...\} \forall j, k, l$

$I_{jl} \in \{0, 1\} \forall j, l$

Here we will explain you the whole model with the code.

First our aim is find the distance between the location of manufacture and all the distribution center. for that we have use the great circle distance method in the code. It find the distance between two points on the circumference of the sphere.

```
def loc_destination(longitude_origin = LOC_longitude,latitude_origin = LOC_latitude,dataframe_dest =dataframe_DC):
    dataframe = pd.DataFrame()
    lst = []
    for i in range(len(dataframe_dest)):
        longitude_des = dataframe_dest["Longitude"][i]
        latitude_des = dataframe_dest["Latitude"][i]
        x = (latitude_origin,longitude_origin )
        y = (latitude_des,longitude_des)
        dist = great_circle(x,y).kilometers
        lst.append(round(dist,ndigits = 0))
    dataframe["Distance"] = lst
    return dataframe
```

Figure 2.2: Distance between Manufacturer to DC

Similarly, we have found the distance between the Distribution center and the Each city.

After we have found the transportation cost. by assuming that the cost of transportation from the Manufacturer to the Distribution center per kilometer is taken as 0.005. For the cost of transportation from the Distribution center to the city is considered as, If distance is less 80 KM is cost will be 1 and and if the distance is more than 80 KM and less than 120 KM then cost will be 2. Similarly if the distance is more than 120 KM and less than 180KM then the cost will be 3. And if distance is more than 180KM then cost is 4.

Now WE have previously taken as the Marginal level capacity of the Distribution center. here We have divided the Distribution center into 4 levels. Now we have found the fixed cost for each level.

For level 0 fixed cost is same as the original fixed cost for the quantity should be less than 200000. For fixed cost level 1 is fixed cost - $quantity_{\_0} * cost_{\_0}$ where the quantity is more than 200000 and less than 250000.

for fixed cost level 2 is equal to fixed cost + $(quantity_{\_1} - quantity_{\_0)}) * cost_{\_0} - (quantity_{\_1} * cost_{\_1})$.

Similarly fixed cost level 3 is equal to fixed cost + $(quantity_{\_1} - quantity_{\_0}) * cost_{\_0} + (quantity_{\_2} - quantity_{\_1}) * cost_{\_1} - quantity_{\_2} * cost_{\_2}$.

e	Longitude	Assembly	Fixed_cost	cost_0	cost_1	cost_2	quantity_1	quantity_2	quantity_0	Fixed_cost_level_1	Fixed_cost_level_2	Fixed_cost_level_3
8	74.641756	715	355536005	716	670	624	200000	300000	150000	248136005	257336005	271136005
1	76.610602	733	236742759	719	675	644	200000	300000	150000	128892759	137692759	146992759
3	74.448134	728	208431351	727	677	648	200000	300000	150000	99381351	109381351	118081351
0	76.513669	689	286875507	701	685	636	200000	300000	150000	181725507	184925507	199625507
1	71.393059	750	435633318	747	650	604	200000	300000	150000	323583318	342983318	356783318
1	77.500799	630	378032713	734	682	602	200000	300000	150000	267932713	278332713	302332713
3	74.637973	753	319715999	744	686	625	200000	300000	150000	208115999	219715999	238015999

Figure 2.3: fixed levels

for this constraints

$F_j + \sum_{p \in l}(Q^h_{jp} - Q^l_{jp})c_{jp} + (x_{jl} - Q^h_{jl})c_{jl} <= OPC_j \forall j \in J, l \in L$

here the fixed cost of operating quantity at Distribution Center j and quantity shipped to that level* cost is less than the total operating cost.

```
#Constraints for cost functions for Level-0
model.constraint_3 = ConstraintList()
for j in no_DCs:
    model.constraint_3.add(expr = DC_data["Fixed_cost"][j]*model.Ijl[j,0] <= model.operating_cost[j])

# constraints for cost function for level 1
model.constraint_4 = ConstraintList()
for j in no_DCs:
    model.constraint_4.add(expr = DC_data["Fixed_cost_level_1"][j]*model.Ijl[j,1] + model.xjl[j,1]*DC_data["cost_0"][j] <= model.operatin

# for level 2
model.constraint_5 = ConstraintList()
for j in no_DCs:
    model.constraint_5.add(expr = DC_data["Fixed_cost_level_2"][j]*model.Ijl[j,2] + model.xjl[j,2]*DC_data["cost_1"][j] <= model.operatin

# level 3
model.constraint_6 = ConstraintList()
for j in no_DCs:
    model.constraint_6.add(expr = DC_data["Fixed_cost_level_3"][j]*model.Ijl[j,3] + model.xjl[j,3]*DC_data["cost_2"][j]<= model.operating
```

Figure 2.4: constraint code

Here we can see that at level 0 the fixed cost when it is open should be less than the operating cost. for level 1 the constraint will be the fixed cost of the Distribution center at level 1 when it open plus the quantity transported from the Distribution center to city at level 1 multiplied by $cost\_0$ is should be less than total operating cost of the distribution center j. Similarly for level 2, fixed cost of level 2 when it is open plus the quantity transported from the Distribution center j to cities multiplied by $cost\_1$ is should be less than the Total operating cost of distribution center j. Similarly for level 3, the fixed cost at level level 3 when it open plus the total quantity transported from Distribution center j to cities multiplied by the $cost\_2$ should be less than the Total operating cost of the distribution center j.

DC location Constraint:
$Q^l_{jl} I_{jl} <= x_{jl} <= Q^h_{jl} I_{jl} \forall j \in J, l \in L$

This constraint is for the quantity at the distribution center at level l should be less than higher capacity of that level and greater than the lower capacity of that level. for level 3 higher capacity is taken as M.

Objective:
min $\sum_{j \in J}[OPC_j + \sum_{l \in L}(t_j x_{jl} + a_j x_{jl}) + \sum_{k \in K} t_{jk} y_{jk}]$ + operating cost of the

manufacturer

Objective is to minimize the cost operating cost of the Distribution center j plus the transportation cost of product transformed from manufacturer to the Distribution center j at level l and total assembly cost of each product transformed to the distribution center plus the transportation cost of each product transported from the Distribution center to different cities plus the operating cost of Manufacturer which is fixed.

Second Problem:

This is also an Integer Programming problem. Because Here I have taken all the variables as the Integer.

This problem is based on greenfield. Here we have to locate hospital at a place that minimizes the cost.

Here we taken in data the district where we can locate the hospital. In the data there longitude and latitude and construction cost are present. And construction cost is taken by us randomly. In the demand file there are city name and longitude and latitude and patients who need hospital for daily basis.

Let,

N: set of all location's of Hospital

I: set of all preferred location where we can allot hospital

K : Number of cities

Parameters:

c_{ij} : cost of the transporatation between District i to city j.

d_i : Construction cost of hospital at District i

p_j: Patient at city j

B : Budget for construction hospital

x_i :1 if Hospital is located at district i, 0 otherwise

Objective Function:

$min \sum_{i \in I} (\sum_{j \in k} c_{ij} * p_j + d_i) * x_1$

construction cost must be less than Budget

$\sum_{i \in I} (x_i * d_i) <= B$

How many hospital we have to open
$\sum_{i \in I} x_i = 1$
$x_i = 1$ for hospital is located at district i, 0 otherwise

Here we can see that our objective function is to minimize the cost of transportation between the hospital to patients and construction cost of hospital at ith District.

```
rt_c_DC_customer = distance_DC_customer.copy(deep= True)
rt_c_DC_customer = pd.DataFrame(np.where(transport_c_DC_customer <= 120,10,transport_c_DC_customer))
rt_c_DC_customer = pd.DataFrame(np.where(((120 < transport_c_DC_customer) & (transport_c_DC_customer <= 180)), 20,transport_c_DC_customer
rt_c_DC_customer = pd.DataFrame(np.where(180< transport_c_DC_customer,30,transport_c_DC_customer))
```

Figure 2.5: cost code

Here we have taken the if distance is less than 120KM then cost is 10 and if distance is in between the 120 and 180 then cost is 20. If the distance is more than 180 then cost is 30.

Chapter 3

Results

In the First Integer programming problem we have use the pyomo package to solve the Optimization Problem. and solver is used as Gurobi.

In this problem, we have found the objective function value which is = **24889692532.96**.

So first will be that if problem was the Integer programming then why objective value is in decimal.

So the Reason behind that, we have found the cost of transportation also in that we have taken as the 0.005 for per kilometer, because of this our objective value is in decimal.

So here we had to find the Minimum number of distribution centers which satisfies the whole demand.

By solving the problem we have got 20 Optimal Distribution centers out of 32 Distribution centers.

Open Distribution center name are: Ajmer, Bharatpur, Bikaner, Bundi, Chittaurgath, churu, Dausa, Dhaulpur, Ganganagar, Hanumangarh, Jaipur, Jalor, Kota, Nagaur, Pali, Sawai madhopur, Sikar, Sirohi, Tonk, Udaipur.

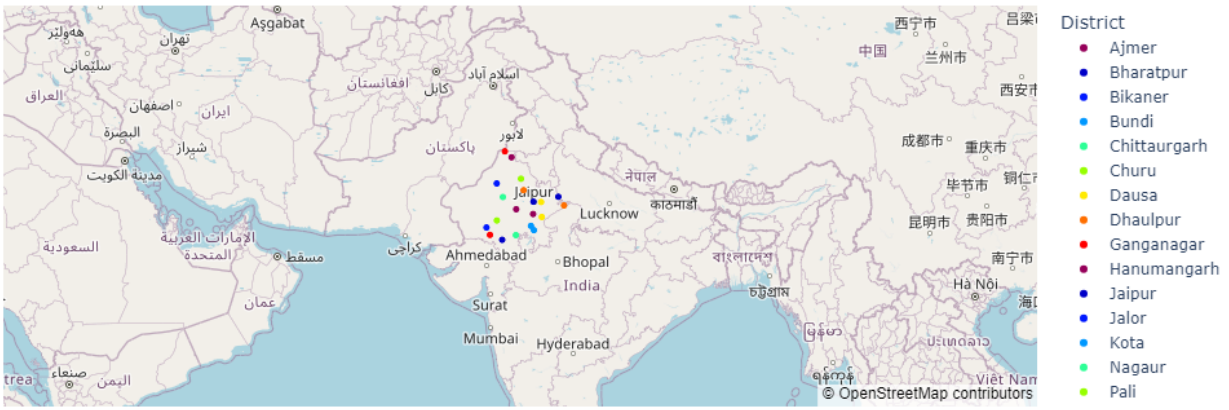

Figure 3.1: Output

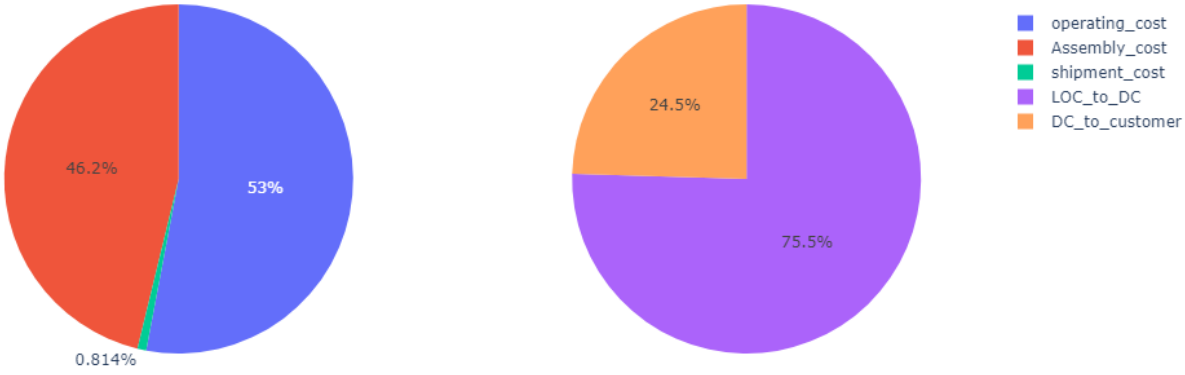

Figure 3.2: Output

Solution of the problem:

```
Gurobi 10.0.1: Solver status ok
Solver termination condition optimal
Objective 24889692532.96
city_name Achalpura 833.0
city_name Agolai 993.0
city_name Ajmer 922.0
city_name Akepura 830.0
city_name Akwās 1134.0
city_name Aliganj 809.0
city_name Almās 1175.0
city_name Ālniyāwās 1185.0
```

Figure 3.3: Output_1

Here In this model cities satisfy their demand by the distributor. Here we can see that warni has 566 product but its demand is 1112, so other products are received to warni by from different distributors.

```
city_name Vasra 939.0
city_name Vavri 1069.0
city_name Vijāna 1135.0
city_name Vilota 1128.0
city_name Viroli 977.0
city_name Viropura 1015.0
city_name Virpura 1092.0
city_name Wāgarbegra 1010.0
city_name Wan 1144.0
city_name Warni 566.0
city_name Waron 1158.0
city_name Wāsa 936.0
```

Figure 3.4: Output for cities

```
city_name Wāli 1043.0
city_name Wando 842.0
city_name Wankaka Gūrha 1149.0
city_name Wānu 1187.0
city_name Wari 987.0
city_name Warkan 885.0
city_name Warni 546.0
city_name Watera 1010.0
city_name Welār 814.0
city_name Zālimpura 1058.0
city_name Zurji ka Khera 962.0
city_name Adhakankar 889.0
city_name Adol 812.0
```

Figure 3.5: filled demand of warni from other distributor

Ainchhāka	27.79469	76.70205	1041
Ainchwāra	27.73084	77.28392	1099
Ainta	27.1547	71.24869	1548
Airāberi	25.8422	73.95402	1315
Aitri	25.29658	73.74452	1047
Aiwād	27.28516	73.96047	1336
Ajabgarh	27.18688	76.2909	1593
Ajabpura	26.53285	76.24564	1617
Ajai	27.06338	75.65931	1076
Ajaipāl ka I	25.72648	75.64365	1138
Ajaipura	26.88629	77.8817	1499

Figure 3.6: Output of cities 100

So now I have taken sample data where only 100 cities in the data and Distribution center are taken to solve problem. After solving the we get solution is: And In our original data there demand is also same.

```
city Ahmda Ka Bās 1076.0
city Ahor 1177.0
city Aībra 1342.0
city Aidalpur 1571.0
city Āīdāna 1240.0
city Ainchhāka 1041.0
city Ainchwāra 1099.0
city Ainta 1548.0
city Airāberi 1315.0
city Aitri 1047.0
city Aiwād 1336.0
city Ajabgarh 1593.0
city Ajabpura 1617.0
city Ajai 1076.0
city Ajaipāl ka Dūngri 1138.0
city Ajaipura 1499.0
Open DC is :Jaipur
```

Figure 3.7: Output of cities 100

In this, we also have find the actual driving time and distance between the Distribution center to cities. Here we use the OSRM (Open Source Routing Machine). It is an open-source routing engine that provides an interactive route plannin and analysis of the network.

Using OSRM we can quickly calculate the optimal route distance in meter and time in seconds.

```
ist_dict = dict()
or i in range(len(df1)):
    for j in range(len(df2)):
        r = requests.get(f"http://router.project-osrm.org/route/v1/driving/{df1['Longitude'][i]},{df1['Latitude']
        try:
            dist_dict[f'({i},{j})']=json.loads(r.content)['routes'][0]['legs'][0]['duration'],json.loads(r.conten
        except:
            print(r.content)
```

Figure 3.8: distance and Duration between DC and City

The idea of code is available on the ORSM site. Using this calculated some real-life driving distance and driving time.

District	City	Duratio	Distanc
Ajmer	Āb ki Dhāni	7959.7	165409.3
Ajmer	Abadsar	12639.5	238096.7
Ajmer	Ābākheri	8469.5	175704.2
Ajmer	Abdara	15789.6	346507.4
Ajmer	Abdul Khār	19446.5	395486.1
Ajmer	Abdul Rahi	23737.8	479011.1
Ajmer	Abdullāh ki	23155.5	503356.4
Ajmer	Abhaipur	15606.8	336330.4
Ajmer	Abhaisingh	12464.8	230783.9
Ajmer	Abhāneri	10205.5	213494.3
Ajmer	Abhaypura	14889.8	311868.6
Ajmer	Abhipur	8632.2	180684.8
Ajmer	Ābu	16653.3	359651.3
Ajmer	Ābu Road	15330.5	347524.4
Alwar	Āb ki Dhāni	19772.6	427756.5
Alwar	Abadsar	14060.5	294259.9
Alwar	Ābākheri	19687.7	418832.3
Alwar	Abdara	27602.5	608854.6
Alwar	Abdul Khār	15426.4	265249.1
Alwar	Abdul Rahi	35550.7	741358.4
Alwar	Abdullāh ki	34968.4	765703.6
Alwar	Abhaipur	4178.7	75973.7

Figure 3.9: distance and Duration between DC and City

In the second problem, we have found that at optimal total cost = 203122077.0 at district churu. As we have found the for one hospital construction. we by changing the constraint we calculate 2,3,4,.. hospital, within given Budget of construction.

At the end of this we have learn about QGIS. QGIS (Quantum Geographic Information system) is tool where we can analyze the geospatial information. Using this we have found out the shortest distance, fastest distance and driving time.

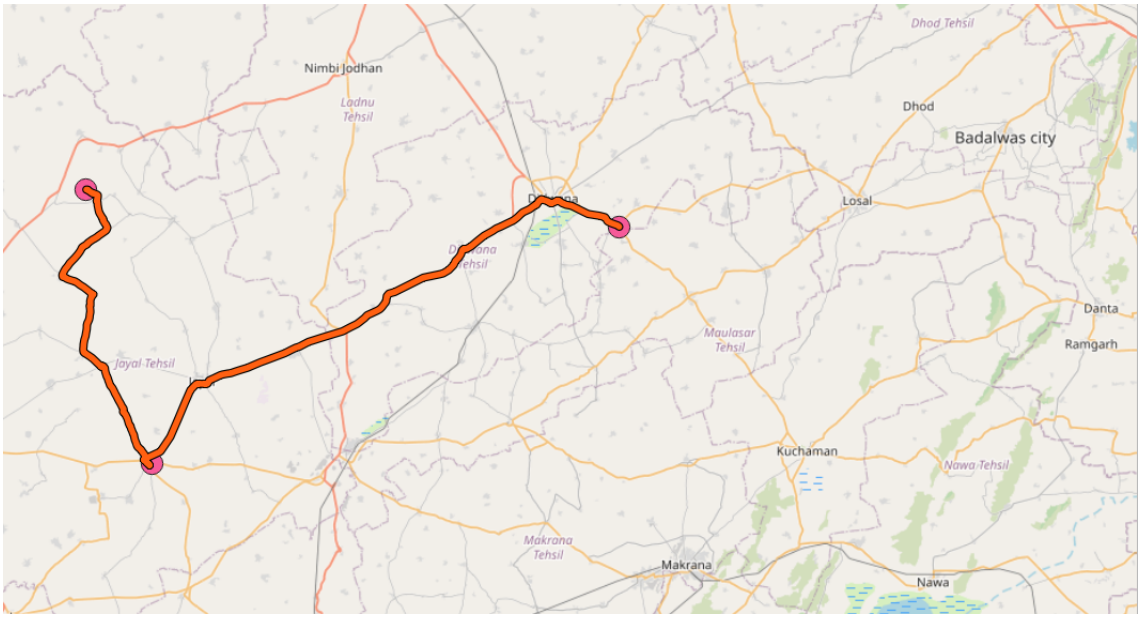

Figure 3.10: Fastest path

Here we can see that orange color path is showing the Fastest path between the two points consider on the map.

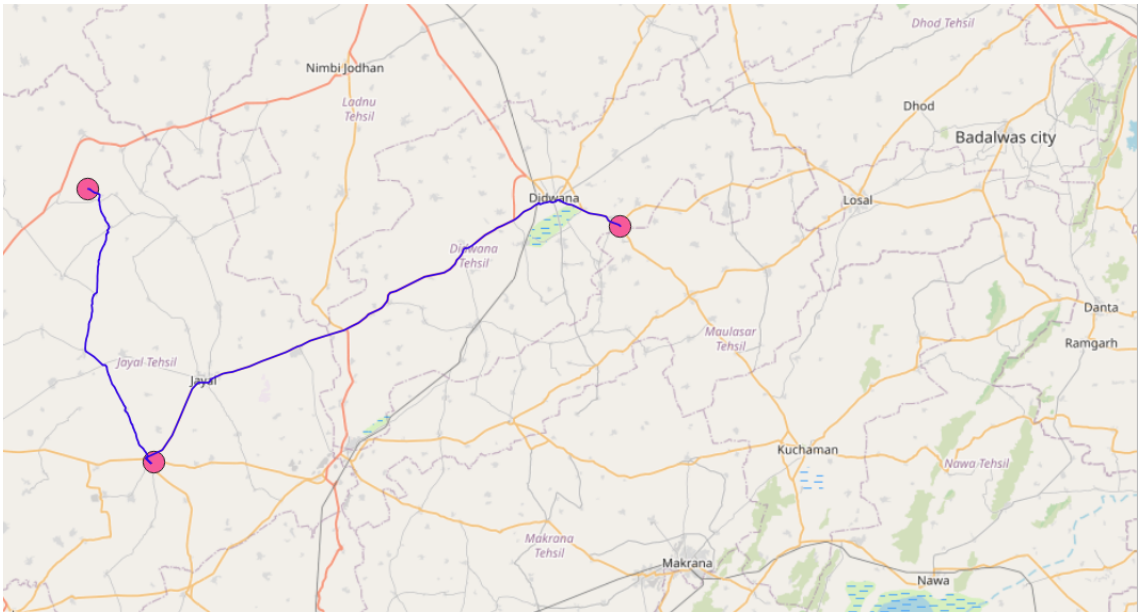

Figure 3.11: Shortest path

Here we can observe that this is the shortest path as compare with the previous one fig. the main difference between shortest and fastest is that in shortest it would take minimum road wise distance from one point to another and in fastest path it would consider that there would not be any traffic. so it takes less time.

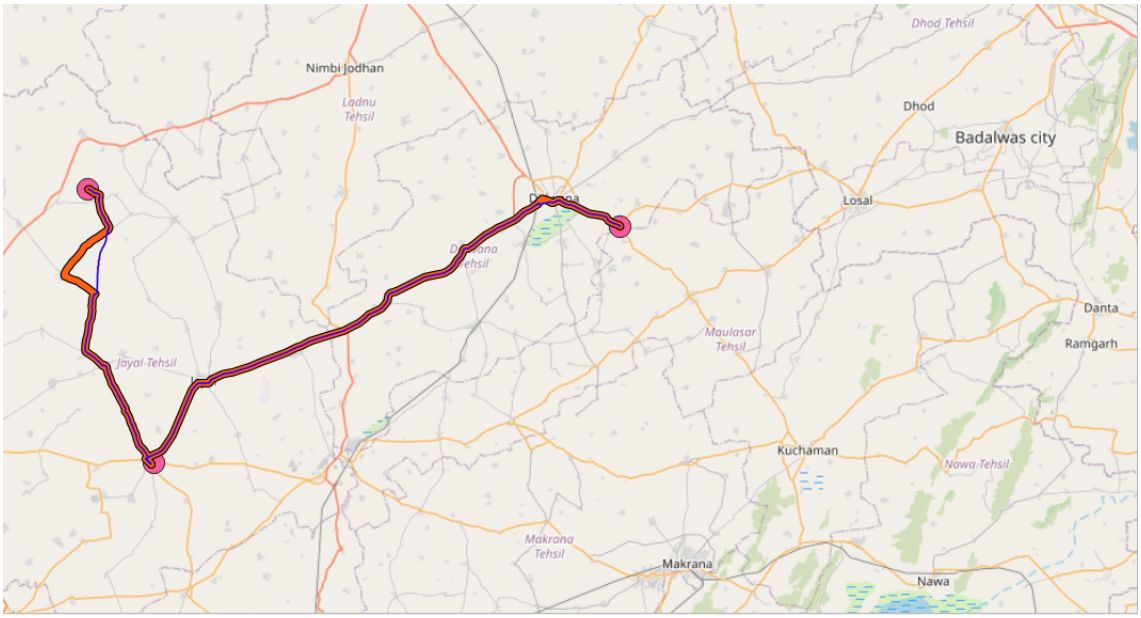

Figure 3.12: Relation between fastest and shortest path
Here we can observe that blue curve is showing shotest path and orange path is showing fastest path.

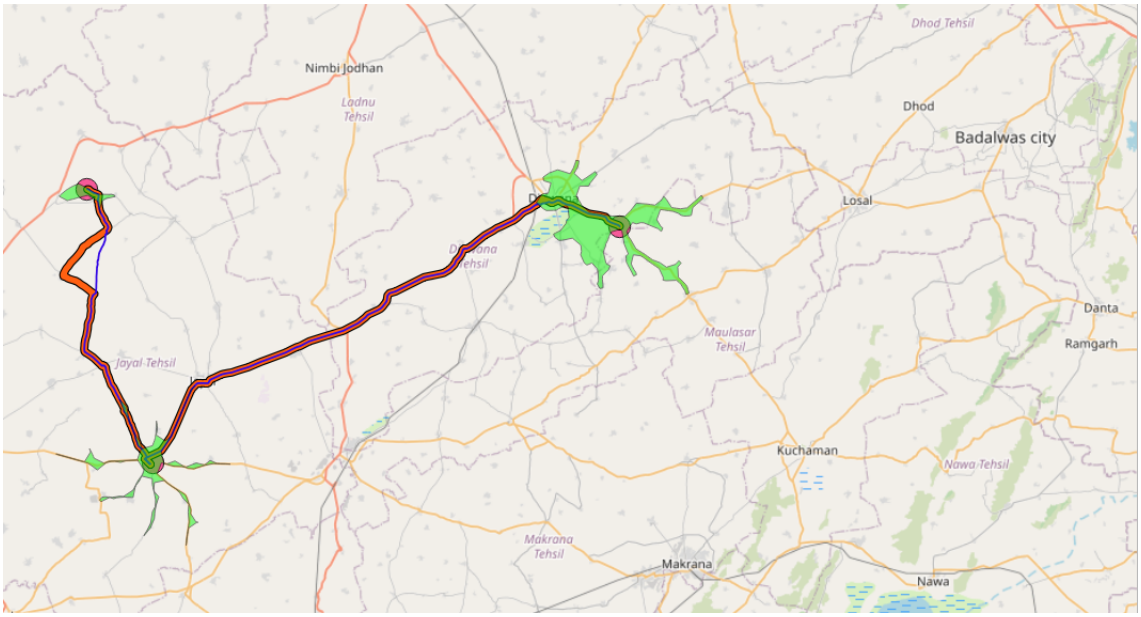

Figure 3.13: area travel in 10 minutes
Here we can see that the region which is in green it shows that in between 10 minutes how much distance we can cover by car. there is some area which is broad and some are narrow, it totally depends on the route of availability.

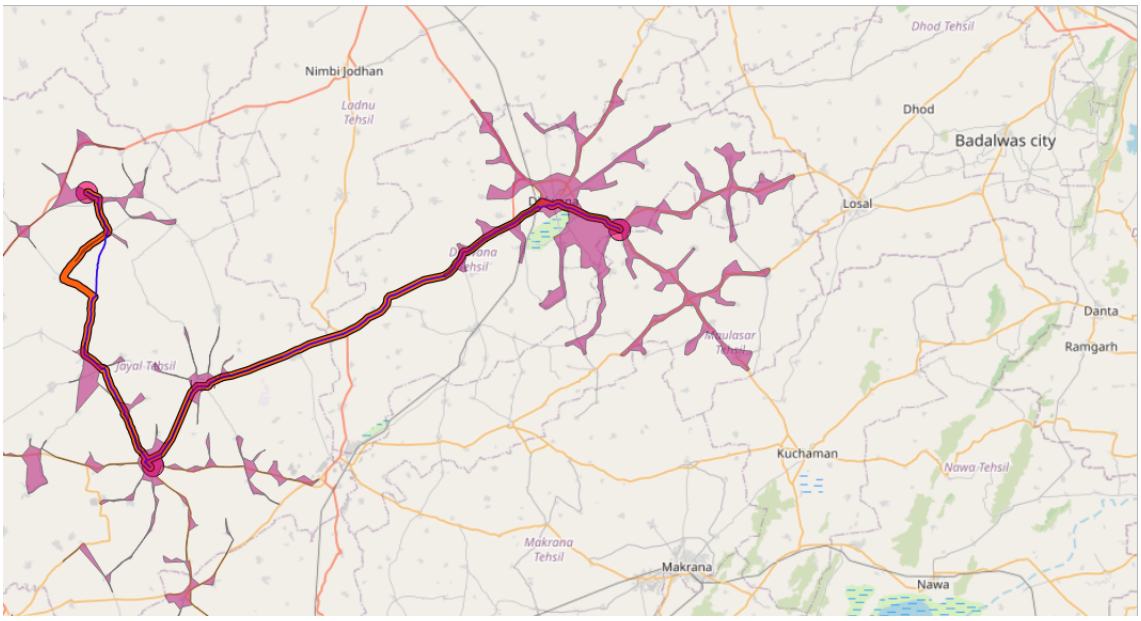

Figure 3.14: area travel in 20 minutes
similarly, we can observe how much we can travel by road, in 20 minutes which is given in pink color.

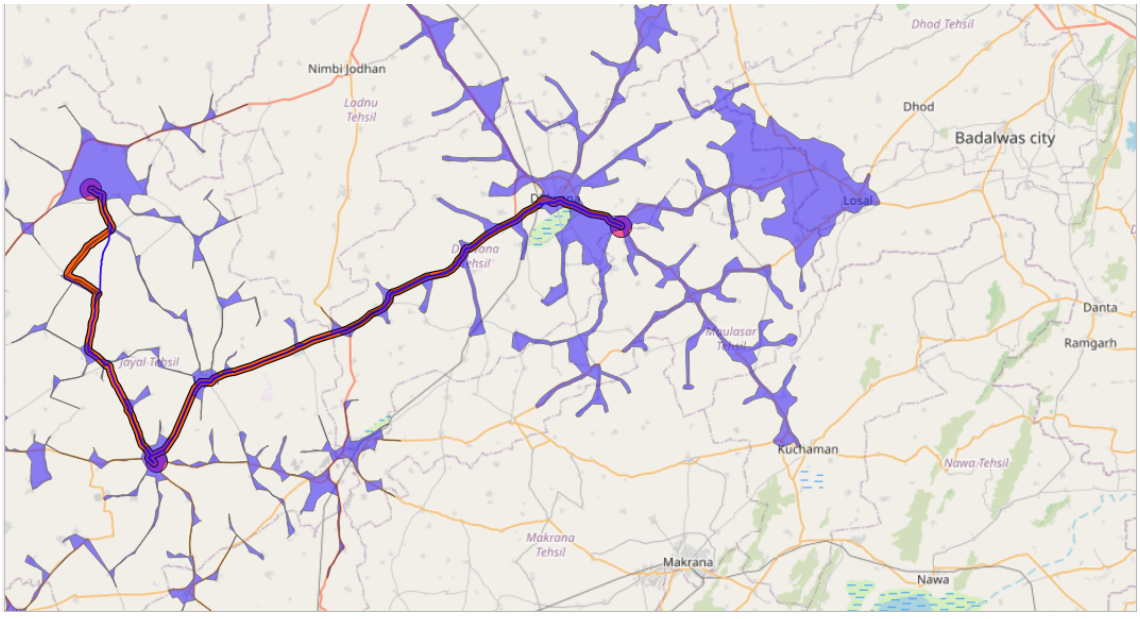

Figure 3.15: area travel in 30 minutes
similarly, we can observe how much we can travel by road, in 30 minutes which is given in blue color.

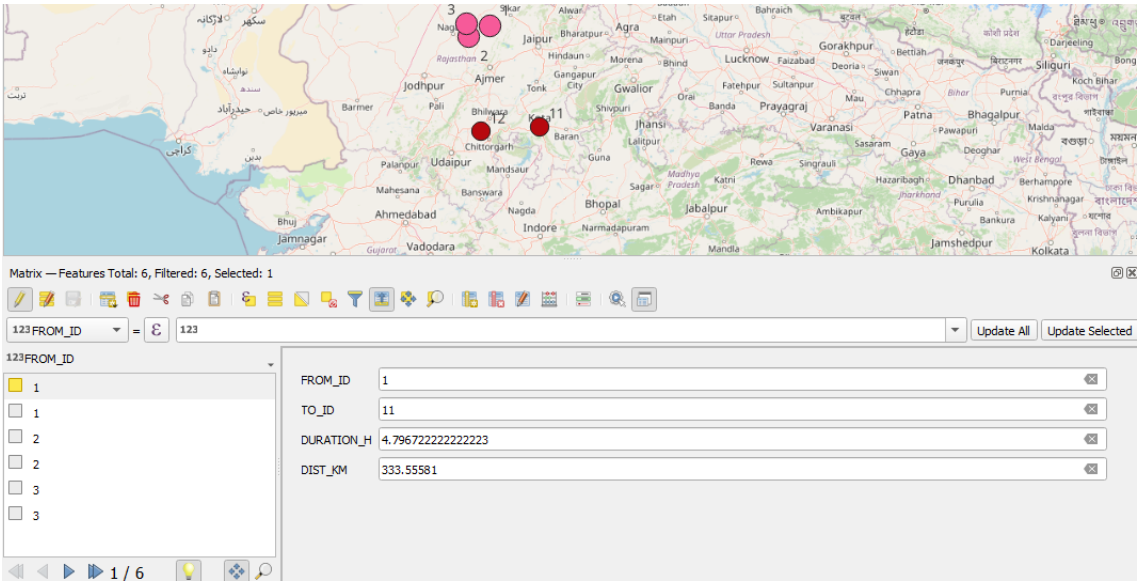

Figure 3.16: distance and time in between two point

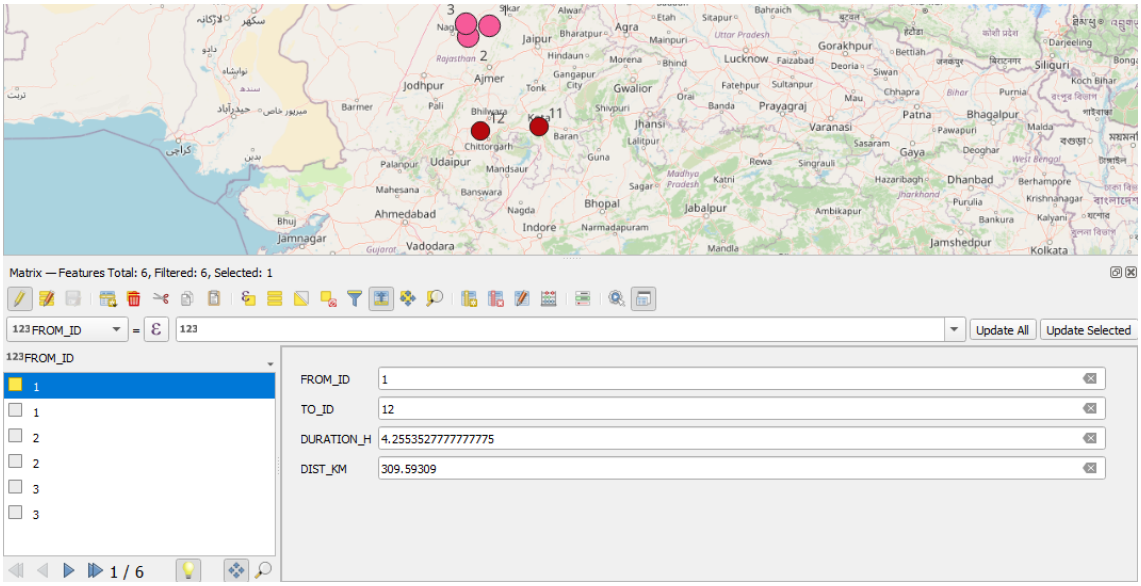

Figure 3.17: distance and time in between two point

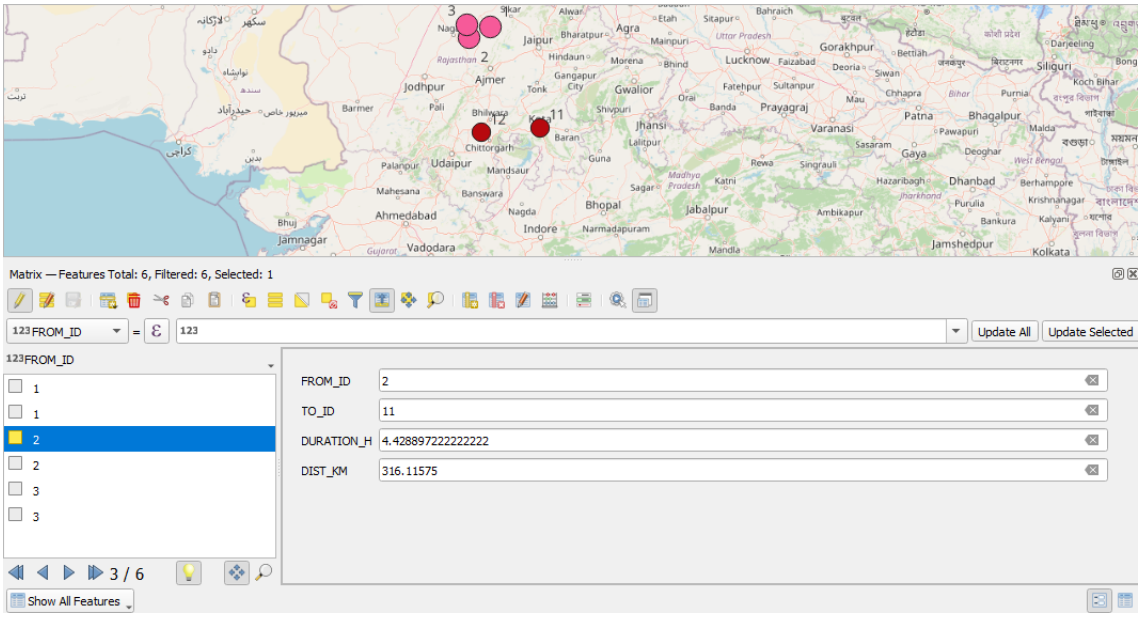

Figure 3.18: distance and time in between two point

Here using QGIS tool we have find the real-time and distance between the two points. Here we can put points which will be our start and destination points and using these points we can observe route, distance and driving time to reach destination. In this you can observe that at source we have taken 3 points and in destination 2 points taken on the map. So it find travel time and travels distance all the source to all the destination which in Kilometer and hours.

Chapter 4

Conclusion:

Here we have seen that the location-allocation model. which is used to locate the minimum number distribution centers that fulfill the demand in Rajasthan state. and minimize the total cost. the data is taken randomly by using some logic but we get optimal solution by considering distance, fixed cost, operating cost and transportation cost. also find a way we locate our first hospital. Here also learn about the QSRM and QGIS for finding distance and travel time. So in this we learn some key things also which we can use, like finding best route, traveling time and distance on the map.

Chapter 5

References:

https://www.sciencedirect.com/science/article/pii/S0198971597000100: :text=A

https://project-osrm.org/

https://github.com/vikas9087/MachineLearning-OperationsResearch-SupplyChainOptimization/blob/master/README.md

https://qgis.org/en/site/

https://github.com/xNok/OR$_{l}ocation_{r}outing_{p}roblem_{s}tudy/tree/master$ /linear$_{p}rogramming_{f}acility_{l}ocation/.ipynb_{c}heckpoints$