Tribhuvan University
Institute of Engineering
Central Campus, Pulchowk

Course: M.Sc. in Transportation Engineering

Subject: Operation Research in Transportation System

Assignment 4: Road Surface Optimization

2024-05-07

Submitted By:
Pragyan Shrestha
079MsTrE014

Submitted To:
Dr. Jagat Kumar Shrestha
Subject Professor

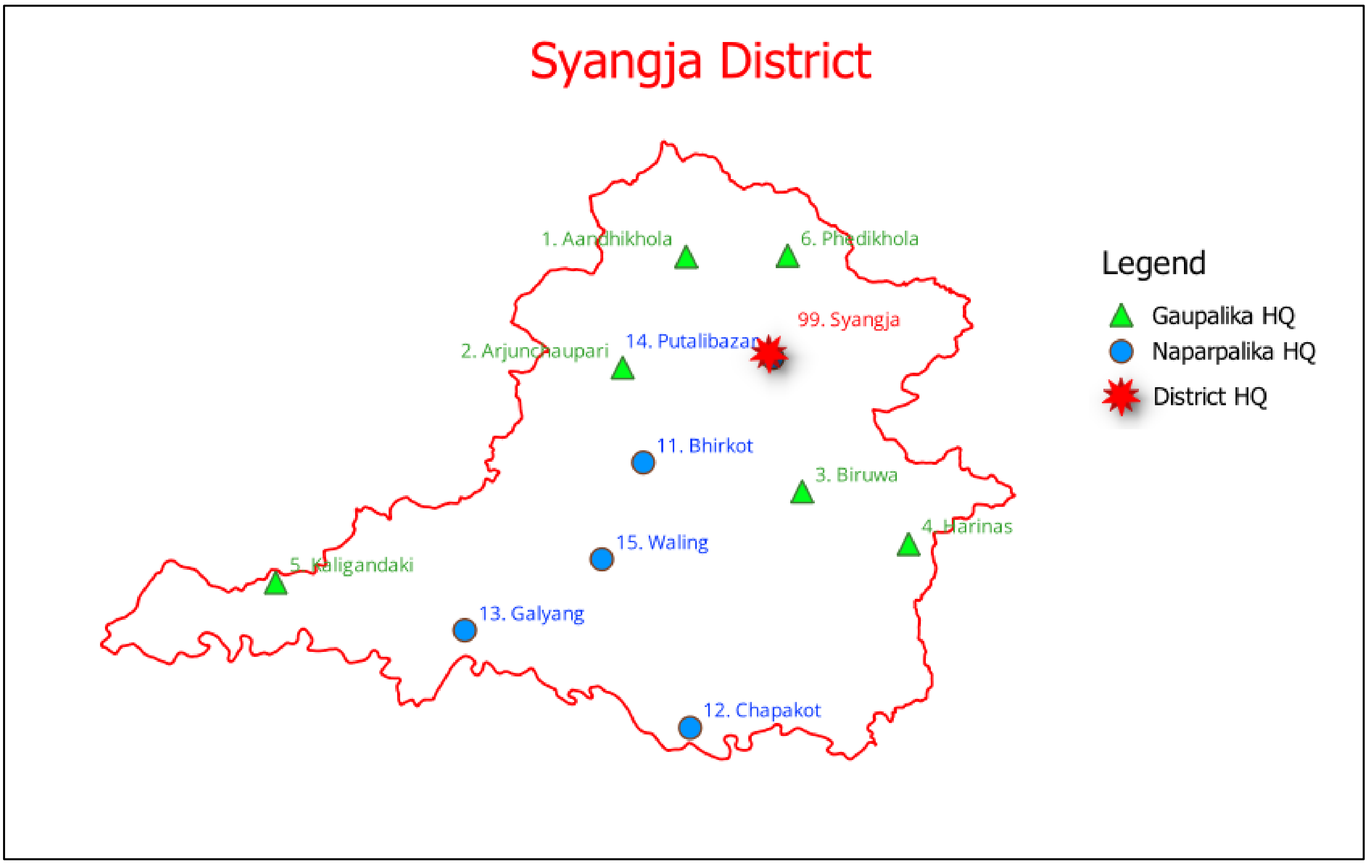
Contents

Assignment

- Containing a network diagram with all links to the connected node.
- Prepare distance matrix. Find a short path matrix from the distance matrix.
- Considering ward centers in the Municipality will be complex, please avoid it.
- Show the GP/NP centers in Google Maps and coordinate in Excel.
- Solve your network for operation cost minimization and population coverage maximization

Introduction

District chosen: Syangja



ID	Gaupalika HQ	Lat	Long	Population
1	Aandhikhola	28.16	83.82	13094
2	Arjunchaupari	28.09	83.78	14045
3	Biruwa	28.02	83.89	14001
4	Harinas	27.99	83.96	13191
5	Kaligandaki	27.97	83.55	17955
6	Phedikhola	28.16	83.88	10899

ID	Nagarpalika HQ	Lat	Long	Population
11	Bhirkot	28.04	83.79	22645
12	Chapakot	27.89	83.82	22969
13	Galyang	27.94	83.68	31034
14	Putalibazar	28.10	83.87	41743
15	Waling	27.98	83.76	50488

ID	District HQ	Lat	Long	Population
99	Syangja	28.10	83.87	253024

Population data from censusnepal.cbs.gov.np/results/population, Census 2021

Methodology

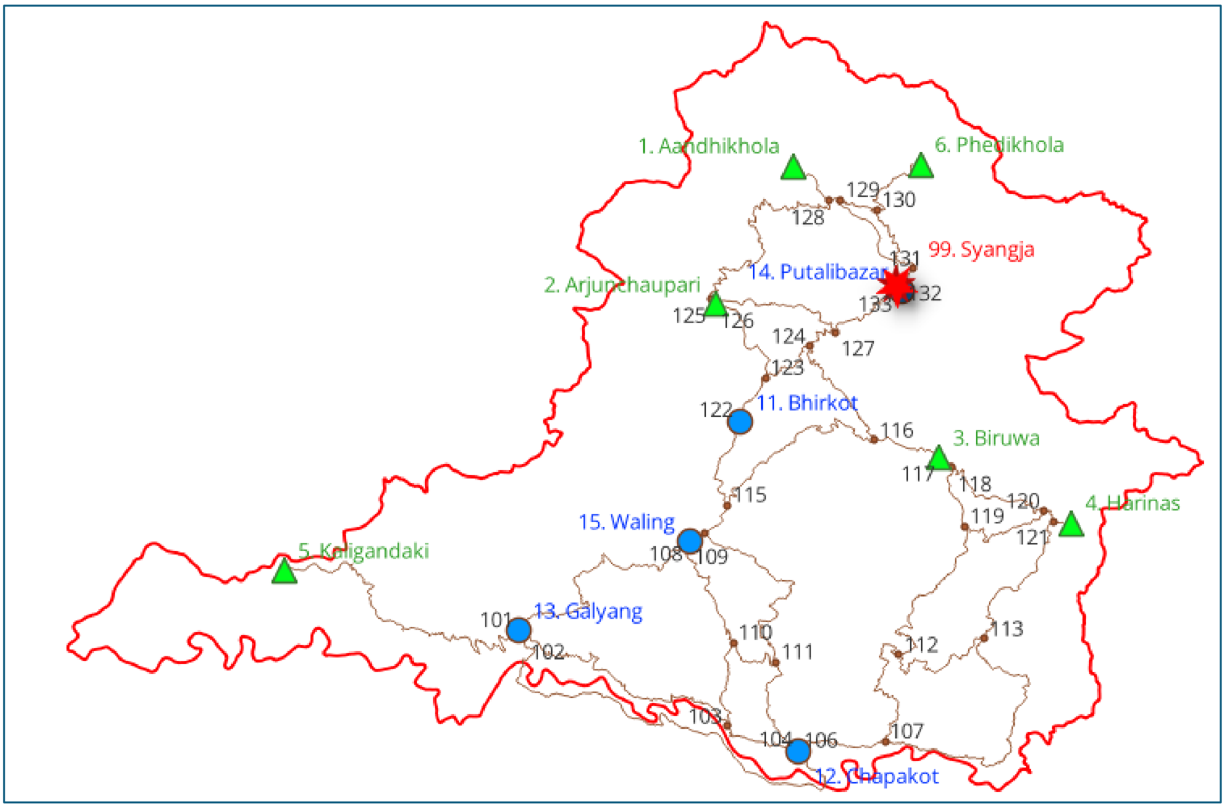
The following steps will guide the assignment problem:

1. Road Network of the chosen district will be obtained.
2. District headquarters and palika headquarters will be identified.
3. A Short Path network will be formed for connecting the headquarters.
4. A Minimum Spanning Tree (MST) will be formed to connect the headquarters.
5. Multiobjective optimization will be formed on the MST network on three different road surface types: earthern, gravel, and pitch for the objectives:
    a. Maximum population coverage
    b. Minimum cost

Road Network Development

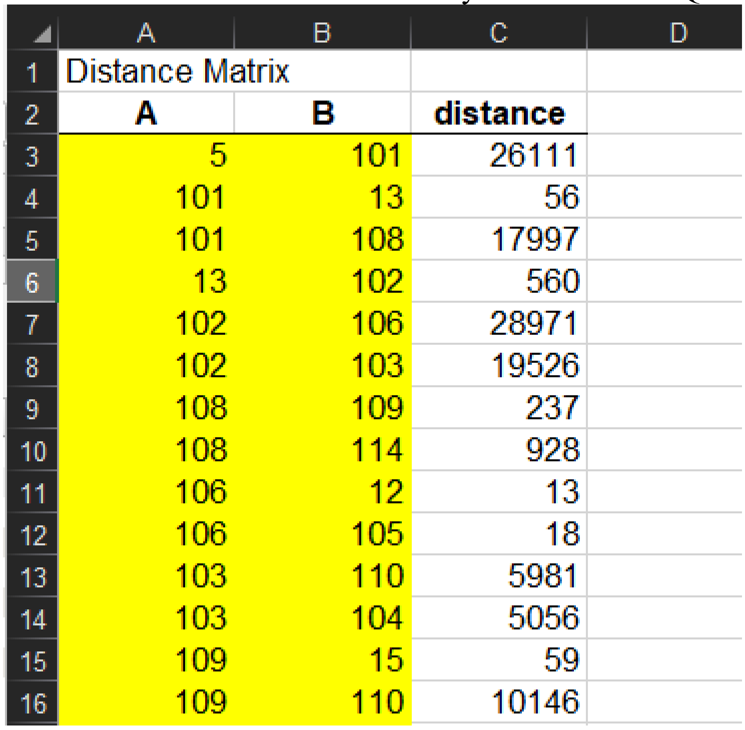
The following workflow was followed to create the network:

1. Identify the headquarters.
2. Code the headquarters.
    a. 1-6 for the six Gaupalikas (Ga Pa) headquarters
    b. 11-15 for the five Nagarpalikas (Na Pa) headquarters
    c. 99 for the district headquarters
3. Using maps.google.com, find the path from A point to B point.
4. Using mapstogpx.com, export the path to gpx format.
5. Import gpx file in QGIS.
6. Manually add intermediate nodes in the road network. Intermediate nodes are given numbers from 100 and above.



Distance Matrix and Analysis

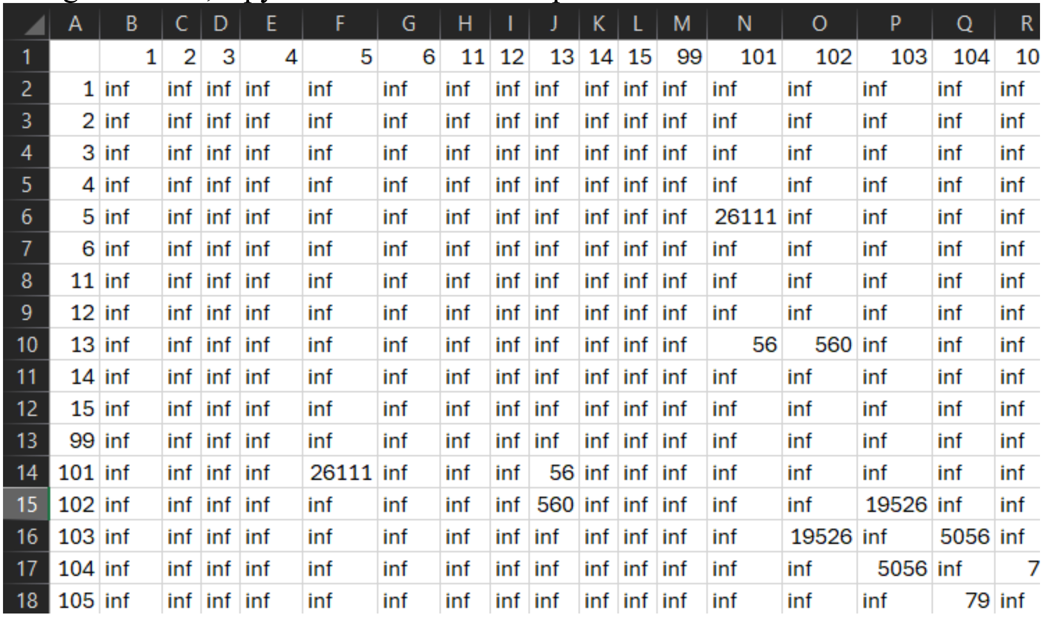
1. Distances from A to B were manually taken off from QGIS in spreadsheet.



	A	B	C	D
1	Distance Matrix			
2	**A**	**B**	**distance**	
3	5	101	26111	
4	101	13	56	
5	101	108	17997	
6	13	102	560	
7	102	106	28971	
8	102	103	19526	
9	108	109	237	
10	108	114	928	
11	106	12	13	
12	106	105	18	
13	103	110	5981	
14	103	104	5056	
15	109	15	59	
16	109	110	10146	

See distances.xlsx.

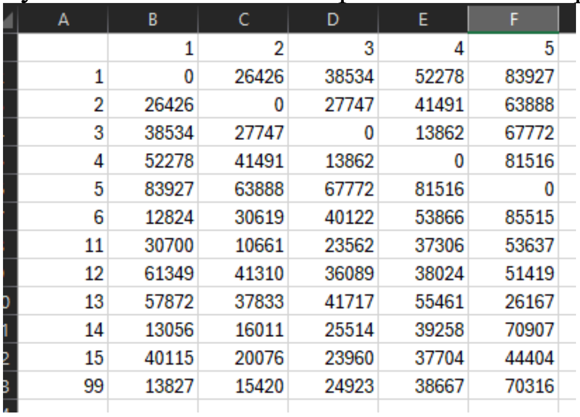
2. Using ChatGPT, a python code was developed to create the distance matrix.



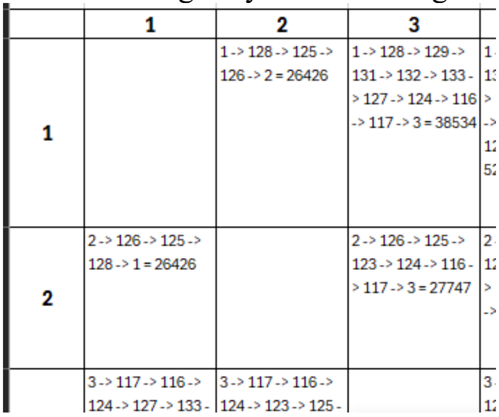
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		1	2	3	4	5	6	11	12	13	14	15	99	101	102	103	104	10
2	1	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
3	2	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
4	3	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
5	4	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
6	5	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	26111	inf	inf	inf	inf
7	6	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
8	11	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
9	12	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
10	13	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	56	560	inf	inf	inf
11	14	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
12	15	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
13	99	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf
14	101	inf	inf	inf	inf	26111	inf	inf	inf	56	inf	inf	inf	inf	inf	inf	inf	inf
15	102	inf	inf	inf	inf	inf	inf	inf	inf	560	inf	inf	inf	inf	inf	19526	inf	inf
16	103	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	19526	inf	5056	inf
17	104	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	5056	inf	7
18	105	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	79	inf

See matrix.py, and matrix.csv.

3. Python codes were developed to create shortpath matrix using Floyd-Warshall algorithm.



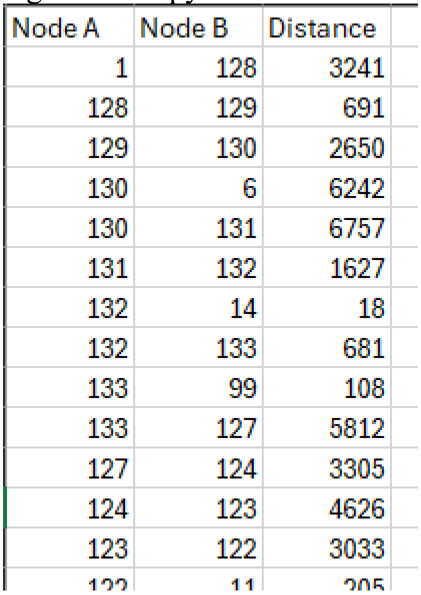
A	B	C	D	E	F
	1	2	3	4	5
1	0	26426	38534	52278	83927
2	26426	0	27747	41491	63888
3	38534	27747	0	13862	67772
4	52278	41491	13862	0	81516
5	83927	63888	67772	81516	0
6	12824	30619	40122	53866	85515
11	30700	10661	23562	37306	53637
12	61349	41310	36089	38024	51419
13	57872	37833	41717	55461	26167
14	13056	16011	25514	39258	70907
15	40115	20076	23960	37704	44404
99	13827	15420	24923	38667	70316



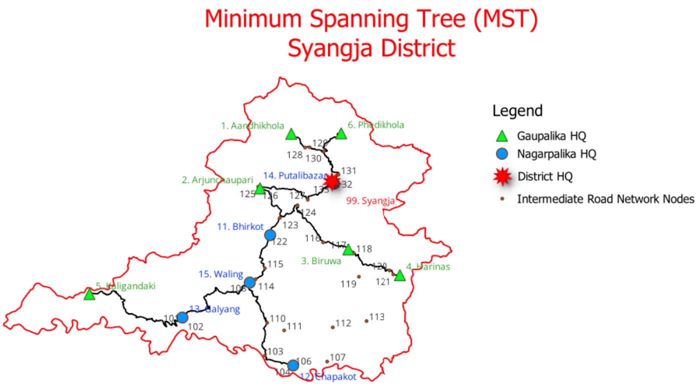
	1	2	3	
1		1 -> 128 -> 125 -> 126 -> 2 = 26426	1 -> 128 -> 129 -> 131 -> 132 -> 133 -> 127 -> 124 -> 116 -> 117 -> 3 = 38534	1 13 > -> 12 52
2	2 -> 126 -> 125 -> 128 -> 1 = 26426		2 -> 126 -> 125 -> 123 -> 124 -> 116 -> 117 -> 3 = 27747	2 12 > ->
	3 -> 117 -> 116 -> 124 -> 127 -> 133 -	3 -> 117 -> 116 -> 124 -> 123 -> 125 -		3 12

See shortest distance.py, shortest path.py, shortdist.csv, and shortpath.xlsx.

4. Using the shortdist.csv, a Minimum Spanning Tree (MST) was also created using Prim's algorithm in python.



Node A	Node B	Distance
1	128	3241
128	129	691
129	130	2650
130	6	6242
130	131	6757
131	132	1627
132	14	18
132	133	681
133	99	108
133	127	5812
127	124	3305
124	123	4626
123	122	3033
122	11	205



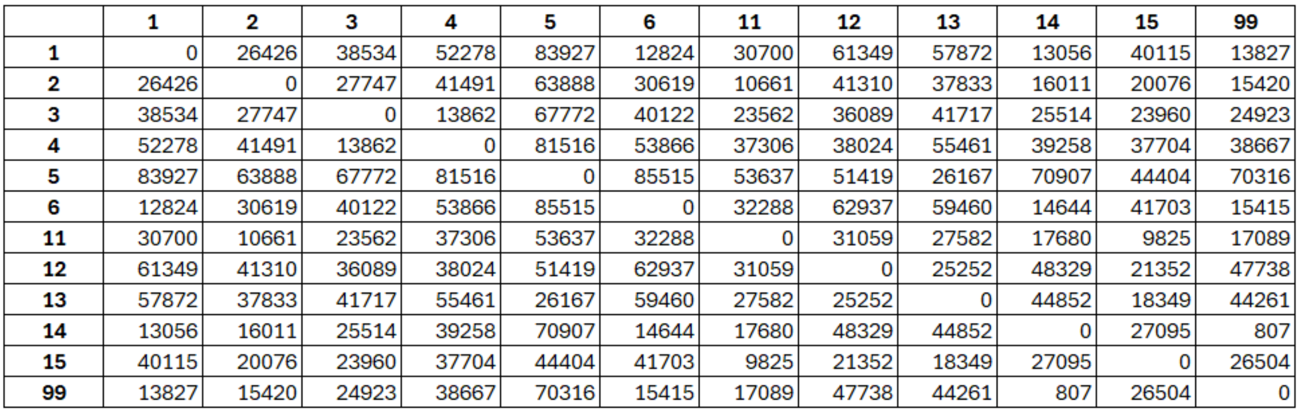
See mst.py, and mst.xlsx.

5. Population table for the MST was also created. Population was assigned to the links as per the population served by that link while going from different palika HQs to the district HQ. *See mstPopulation.xlsx.*

6. Using the scirpt *maxtrix.py*, *mstmatrix.csv* and *popnmatrix.csv* were created too.

Short Path Matrix

Short Path Matrix: Syangja District



	1	2	3	4	5	6	11	12	13	14	15	99
1	0	26426	38534	52278	83927	12824	30700	61349	57872	13056	40115	13827
2	26426	0	27747	41491	63888	30619	10661	41310	37833	16011	20076	15420
3	38534	27747	0	13862	67772	40122	23562	36089	41717	25514	23960	24923
4	52278	41491	13862	0	81516	53866	37306	38024	55461	39258	37704	38667
5	83927	63888	67772	81516	0	85515	53637	51419	26167	70907	44404	70316
6	12824	30619	40122	53866	85515	0	32288	62937	59460	14644	41703	15415
11	30700	10661	23562	37306	53637	32288	0	31059	27582	17680	9825	17089
12	61349	41310	36089	38024	51419	62937	31059	0	25252	48329	21352	47738
13	57872	37833	41717	55461	26167	59460	27582	25252	0	44852	18349	44261
14	13056	16011	25514	39258	70907	14644	17680	48329	44852	0	27095	807
15	40115	20076	23960	37704	44404	41703	9825	21352	18349	27095	0	26504
99	13827	15420	24923	38667	70316	15415	17089	47738	44261	807	26504	0

Short Path Matrix: Syangja District

079MSTrE014 Pragyan Shrestha

1-6:	Ga Pa HQ
11-15:	Na Pa HQ
99:	District HQ
101-133:	Intermediate Nodes

Gaupalika Headquarters	
1	Aandhikhola
2	Arjunchaupari
3	Biruwa
4	Harinas
5	Kaligandaki
6	Phedikhola

Nagarpalika Headquarters	
11	Bhirkot
12	Chapakot
13	Galyang
14	Putalibazar
15	Waling

District Headquarters	
99	Syangja

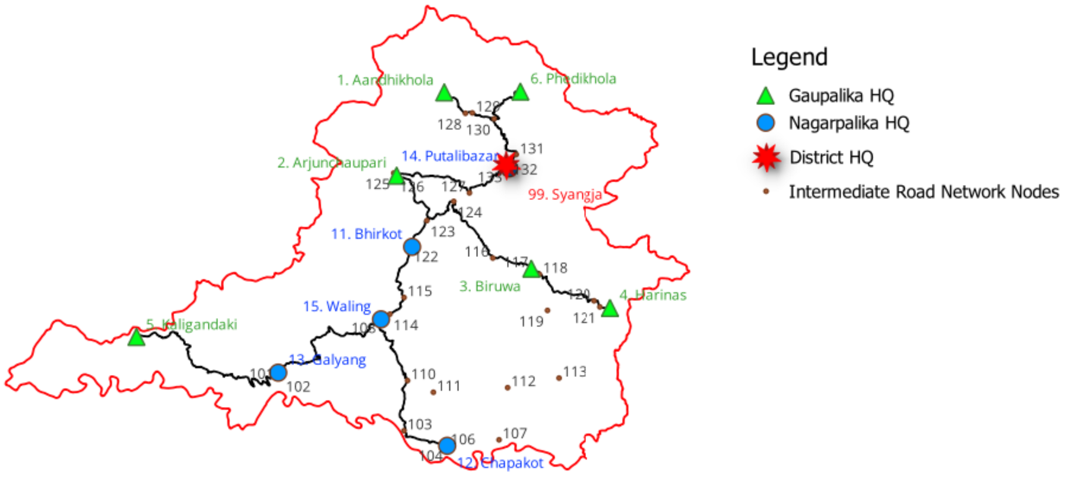
	1	2	3	4	5	6	11	12	13	14	15	99
1		1 -> 128 -> 125 -> 126 -> 2 = 26426	1 -> 128 -> 129 -> 131 -> 132 -> 133 -> 127 -> 124 -> 116 -> 117 -> 3 = 38534	1 -> 128 -> 129 -> 131 -> 132 -> 133 -> 127 -> 124 -> 116 -> 117 -> 118 -> 120 -> 121 -> 4 = 52278	1 -> 128 -> 129 -> 131 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 101 -> 5 = 83927	1 -> 128 -> 129 -> 130 -> 6 = 12824	1 -> 128 -> 129 -> 131 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 11 = 30700	1 -> 128 -> 129 -> 131 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 109 -> 110 -> 103 -> 104 -> 105 -> 106 -> 12 = 61349	1 -> 128 -> 129 -> 131 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 101 -> 13 = 57872	1 -> 128 -> 129 -> 131 -> 132 -> 14 = 13056	1 -> 128 -> 129 -> 131 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 109 -> 15 = 40115	1 -> 128 -> 129 -> 131 -> 132 -> 133 -> 99 = 13827
2	2 -> 126 -> 125 -> 128 -> 1 = 26426		2 -> 126 -> 125 -> 123 -> 124 -> 116 -> 117 -> 3 = 27747	2 -> 126 -> 125 -> 123 -> 124 -> 116 -> 117 -> 118 -> 120 -> 121 -> 4 = 41491	2 -> 126 -> 125 -> 123 -> 122 -> 115 -> 114 -> 108 -> 101 -> 5 = 63888	2 -> 126 -> 127 -> 133 -> 132 -> 131 -> 130 -> 6 = 30619	2 -> 126 -> 125 -> 123 -> 122 -> 11 = 10661	2 -> 126 -> 125 -> 123 -> 122 -> 115 -> 114 -> 108 -> 109 -> 110 -> 103 -> 104 -> 105 -> 106 -> 12 = 41310	2 -> 126 -> 125 -> 123 -> 122 -> 115 -> 114 -> 108 -> 101 -> 13 = 37833	2 -> 126 -> 127 -> 133 -> 132 -> 14 = 16011	2 -> 126 -> 125 -> 123 -> 122 -> 115 -> 114 -> 108 -> 109 -> 15 = 20076	2 -> 126 -> 127 -> 133 -> 99 = 15420
3	3 -> 117 -> 116 -> 124 -> 127 -> 133 -> 132 -> 131 -> 129 -> 128 -> 1 = 38534	3 -> 117 -> 116 -> 124 -> 123 -> 125 -> 126 -> 2 = 27747		3 -> 117 -> 118 -> 120 -> 121 -> 4 = 13862	3 -> 117 -> 116 -> 115 -> 114 -> 108 -> 101 -> 5 = 67772	3 -> 117 -> 116 -> 124 -> 127 -> 133 -> 132 -> 131 -> 130 -> 6 = 40122	3 -> 117 -> 116 -> 124 -> 123 -> 122 -> 11 = 23562	3 -> 117 -> 118 -> 119 -> 112 -> 107 -> 105 -> 106 -> 12 = 36089	3 -> 117 -> 116 -> 115 -> 114 -> 108 -> 101 -> 13 = 41717	3 -> 117 -> 116 -> 124 -> 127 -> 133 -> 132 -> 14 = 25514	3 -> 117 -> 116 -> 115 -> 114 -> 108 -> 109 -> 15 = 23960	3 -> 117 -> 116 -> 124 -> 127 -> 133 -> 99 = 24923
4	4 -> 121 -> 120 -> 118 -> 117 -> 116 -> 124 -> 127 -> 133 -> 132 -> 131 -> 129 -> 128 -> 1 = 52278	4 -> 121 -> 120 -> 118 -> 117 -> 116 -> 124 -> 123 -> 125 -> 126 -> 2 = 41491	4 -> 121 -> 120 -> 118 -> 117 -> 3 = 13862		4 -> 121 -> 120 -> 118 -> 117 -> 116 -> 115 -> 114 -> 108 -> 101 -> 5 = 81516	4 -> 121 -> 120 -> 118 -> 117 -> 116 -> 124 -> 127 -> 133 -> 132 -> 131 -> 130 -> 6 = 53866	4 -> 121 -> 120 -> 118 -> 117 -> 116 -> 124 -> 123 -> 122 -> 11 = 37306	4 -> 121 -> 113 -> 107 -> 105 -> 106 -> 12 = 38024	4 -> 121 -> 120 -> 118 -> 117 -> 116 -> 115 -> 114 -> 108 -> 101 -> 13 = 55461	4 -> 121 -> 120 -> 118 -> 117 -> 116 -> 124 -> 127 -> 133 -> 132 -> 14 = 39258	4 -> 121 -> 120 -> 118 -> 117 -> 116 -> 115 -> 114 -> 108 -> 109 -> 15 = 37704	4 -> 121 -> 120 -> 118 -> 117 -> 116 -> 124 -> 127 -> 133 -> 99 = 38667
5	5 -> 101 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 131 -> 129 -> 128 -> 1 = 83927	5 -> 101 -> 108 -> 114 -> 115 -> 122 -> 123 -> 125 -> 126 -> 2 = 63888	5 -> 101 -> 108 -> 114 -> 115 -> 116 -> 117 -> 3 = 67772	5 -> 101 -> 108 -> 114 -> 115 -> 116 -> 117 -> 118 -> 120 -> 121 -> 4 = 81516		5 -> 101 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 131 -> 130 -> 6 = 85515	5 -> 101 -> 108 -> 114 -> 115 -> 122 -> 11 = 53637	5 -> 101 -> 13 -> 102 -> 103 -> 104 -> 105 -> 106 -> 12 = 51419	5 -> 101 -> 13 = 26167	5 -> 101 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 14 = 70907	5 -> 101 -> 108 -> 109 -> 15 = 44404	5 -> 101 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 99 = 70316
6	6 -> 130 -> 129 -> 128 -> 1 = 12824	6 -> 130 -> 131 -> 132 -> 133 -> 127 -> 126 -> 2 = 30619	6 -> 130 -> 131 -> 132 -> 133 -> 127 -> 124 -> 116 -> 117 -> 3 = 40122	6 -> 130 -> 131 -> 132 -> 133 -> 127 -> 124 -> 116 -> 117 -> 118 -> 120 -> 121 -> 4 = 53866	6 -> 130 -> 131 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 101 -> 5 = 85515		6 -> 130 -> 131 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 11 = 32288	6 -> 130 -> 131 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 109 -> 110 -> 103 -> 104 -> 105 -> 106 -> 12 = 62937	6 -> 130 -> 131 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 101 -> 13 = 59460	6 -> 130 -> 131 -> 132 -> 14 = 14644	6 -> 130 -> 131 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 109 -> 15 = 41703	6 -> 130 -> 131 -> 132 -> 133 -> 99 = 15415
11	11 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 131 -> 129 -> 128 -> 1 = 30700	11 -> 122 -> 123 -> 125 -> 126 -> 2 = 10661	11 -> 122 -> 123 -> 124 -> 116 -> 117 -> 3 = 23562	11 -> 122 -> 123 -> 124 -> 116 -> 117 -> 118 -> 120 -> 121 -> 4 = 37306	11 -> 122 -> 115 -> 114 -> 108 -> 101 -> 5 = 53637	11 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 131 -> 130 -> 6 = 32288		11 -> 122 -> 115 -> 114 -> 108 -> 109 -> 110 -> 103 -> 104 -> 105 -> 106 -> 12 = 31059	11 -> 122 -> 115 -> 114 -> 108 -> 101 -> 13 = 27582	11 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 14 = 17680	11 -> 122 -> 115 -> 114 -> 108 -> 109 -> 15 = 9825	11 -> 122 -> 123 -> 124 -> 127 -> 133 -> 99 = 17089
12	12 -> 106 -> 105 -> 104 -> 103 -> 110 -> 109 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 131 -> 129 -> 128 -> 1 = 61349	12 -> 106 -> 105 -> 104 -> 103 -> 110 -> 109 -> 108 -> 114 -> 115 -> 122 -> 123 -> 125 -> 126 -> 2 = 41310	12 -> 106 -> 105 -> 107 -> 112 -> 119 -> 118 -> 117 -> 3 = 36089	12 -> 106 -> 105 -> 107 -> 113 -> 121 -> 4 = 38024	12 -> 106 -> 105 -> 104 -> 103 -> 102 -> 13 -> 101 -> 5 = 51419	12 -> 106 -> 105 -> 104 -> 103 -> 110 -> 109 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 131 -> 130 -> 6 = 62937	12 -> 106 -> 105 -> 104 -> 103 -> 110 -> 109 -> 108 -> 114 -> 115 -> 122 -> 11 = 31059		12 -> 106 -> 105 -> 104 -> 103 -> 102 -> 13 = 25252	12 -> 106 -> 105 -> 104 -> 103 -> 110 -> 109 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 14 = 48329	12 -> 106 -> 105 -> 104 -> 103 -> 110 -> 109 -> 15 = 21352	12 -> 106 -> 105 -> 104 -> 103 -> 110 -> 109 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 99 = 47738
13	13 -> 101 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 131 -> 129 -> 128 -> 1 = 57872	13 -> 101 -> 108 -> 114 -> 115 -> 122 -> 123 -> 125 -> 126 -> 2 = 37833	13 -> 101 -> 108 -> 114 -> 115 -> 116 -> 117 -> 3 = 41717	13 -> 101 -> 108 -> 114 -> 115 -> 116 -> 117 -> 118 -> 120 -> 121 -> 4 = 55461	13 -> 101 -> 5 = 26167	13 -> 101 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 131 -> 130 -> 6 = 59460	13 -> 101 -> 108 -> 114 -> 115 -> 122 -> 11 = 27582	13 -> 102 -> 103 -> 104 -> 105 -> 106 -> 12 = 25252		13 -> 101 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 14 = 44852	13 -> 101 -> 108 -> 109 -> 15 = 18349	13 -> 101 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 99 = 44261
14	14 -> 132 -> 131 -> 129 -> 128 -> 1 = 13056	14 -> 132 -> 133 -> 127 -> 126 -> 2 = 16011	14 -> 132 -> 133 -> 127 -> 124 -> 116 -> 117 -> 3 = 25514	14 -> 132 -> 133 -> 127 -> 124 -> 116 -> 117 -> 118 -> 120 -> 121 -> 4 = 39258	14 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 101 -> 5 = 70907	14 -> 132 -> 131 -> 130 -> 6 = 14644	14 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 11 = 17680	14 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 109 -> 110 -> 103 -> 104 -> 105 -> 106 -> 12 = 48329	14 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 101 -> 13 = 44852		14 -> 132 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 109 -> 15 = 27095	14 -> 132 -> 133 -> 99 = 807
15	15 -> 109 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 131 -> 129 -> 128 -> 1 = 40115	15 -> 109 -> 108 -> 114 -> 115 -> 122 -> 123 -> 125 -> 126 -> 2 = 20076	15 -> 109 -> 108 -> 114 -> 115 -> 116 -> 117 -> 3 = 23960	15 -> 109 -> 108 -> 114 -> 115 -> 116 -> 117 -> 118 -> 120 -> 121 -> 4 = 37704	15 -> 109 -> 108 -> 101 -> 5 = 44404	15 -> 109 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 131 -> 130 -> 6 = 41703	15 -> 109 -> 108 -> 114 -> 115 -> 122 -> 11 = 9825	15 -> 109 -> 110 -> 103 -> 104 -> 105 -> 106 -> 12 = 21352	15 -> 109 -> 108 -> 101 -> 13 = 18349	15 -> 109 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 132 -> 14 = 27095		15 -> 109 -> 108 -> 114 -> 115 -> 122 -> 123 -> 124 -> 127 -> 133 -> 99 = 26504
99	99 -> 133 -> 132 -> 131 -> 129 -> 128 -> 1 = 13827	99 -> 133 -> 127 -> 126 -> 2 = 15420	99 -> 133 -> 127 -> 124 -> 116 -> 117 -> 3 = 24923	99 -> 133 -> 127 -> 124 -> 116 -> 117 -> 118 -> 120 -> 121 -> 4 = 38667	99 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 101 -> 5 = 70316	99 -> 133 -> 132 -> 131 -> 130 -> 6 = 15415	99 -> 133 -> 127 -> 124 -> 123 -> 122 -> 11 = 17089	99 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 109 -> 110 -> 103 -> 104 -> 105 -> 106 -> 12 = 47738	99 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 101 -> 13 = 44261	99 -> 133 -> 132 -> 14 = 807	99 -> 133 -> 127 -> 124 -> 123 -> 122 -> 115 -> 114 -> 108 -> 109 -> 15 = 26504	

Minimum Spanning Tree

Node A	Node B	Population served of palika	Population served
5	101	5	17,955
101	13	13	31,034
101	108	13, 5	48,989
108	109	15	50,488
109	15	15, 12	73,457
109	110	12	22,969
110	103	12	22,969
103	104	12	22,969
104	105	12	22,969
105	106	12	22,969
106	12	12	22,969
108	114	15, 5, 13, 12	1,22,446
114	115	15, 5, 13, 12	1,22,446
115	122	15, 5, 13, 12	1,22,446
122	11	11	22,645
122	123	11, 15, 5, 13, 12	1,45,091
123	124	2, 11, 15, 5, 13, 12	1,59,136
124	116	3, 4	27,192
116	117	3, 4	27,192
117	3	3	14,001

Node A	Node B	Population served of palika	Population served
117	118	4	13191
118	120	4	13191
120	121	4	13191
121	4	4	13191
123	125	3, 4, 11, 15, 13, 12, 5	172283
125	126	3, 4, 11, 15, 13, 12, 5	172283
126	2	2	14045
126	127	2, 3, 4, 11, 15, 13, 12, 5	186328
127	133	2, 3, 4, 11, 15, 13, 12, 5	186328
133	99	all	252064
133	132	1, 6, 14	65736
132	14	14	41743
132	131	1, 6	23993
131	130	1, 6	23993
130	129	1	13094
129	128	1	13094
128	1	1	13094
130	6	6	10899
101	13	13	31034
101	5	5	17955

Minimum Spanning Tree (MST)
Syangja District



Road Surface Optimization

Parameters and conditions used:

No. of MST links = 37
Budget = 50,00,00,000
Unit Operation Cost for
Earthern road = 500 per km
Gravel road = 450 per km
Pitch road = 300 per km
Unit Improvement Cost for
Earthern road = 0
Gravel road = 50,00,000 per km
Pitch road = 2,00,00,000 per km

Weightage = 1 for all

Coverage constraint: At least 25 links
Road surface constraints:

- At least 10 gravel roads
- At least 10 pitch roads

MPL Code

```
{ MOProblem.mpl }
TITLE
MOP;
INDEX
i := 1..39
j := i;

DATA
        UeOperation := 500/1000;            !500 per km
        UgOperation := 450/1000;            !450 per km
        UpOperation := 350/1000;            !350 per km
        UeImprove := 0;
        UgImprove := 5000000/1000;          !50 Lakh per km
        UpImprove := 20000000/1000;         !2 Crore per km
        mstNodes  := 39;
        Budget := 500000000;                !5

        !Total Cumulative Population Coverage = 2364073

        !MST distance matrix
        d[i,j] := (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3241,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,67,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,59,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1159,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,26111,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6242,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,205,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,56,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,18,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,59,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,108
0,0,0,0,26111,0,0,0,56,0,0,0,0,0,0,0,0,17997,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,5056,0,0,0,0,5981,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,5056,0,79,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,79,0,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,13,0,0,0,0,0,0,0,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,17997,0,0,0,0,0,237,0,928,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,59,0,0,0,0,0,0,237,0,10146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,5981,0,0,0,0,10146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,928,0,0,0,2254,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2254,0,0,0,0,0,0,6142,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4836,0,0,0,0,0,10803,0,0,0,0,0,0,0,0,0
0,0,59,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4836,0,1123,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1123,0,10051,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,10051,0,1470,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,1159,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1470,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,205,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6142,0,0,0,0,0,0,3033,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3033,0,4626,7054,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,10803,0,0,0,0,0,4626,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7054,0,0,302,0,0,0,0,0,0,0
0,67,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,302,0,9433,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,9433,0,0,0,0,0,0,5812
3241,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,691,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,691,0,2650,0,0,0
0,0,0,0,0,6242,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2650,0,6757,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6757,0,1627,0
0,0,0,0,0,0,0,0,0,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1627,0,681
0,0,0,0,0,0,0,0,0,0,0,108,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5812,0,0,0,0,681,0
);

        !Population on MST links matrix
        P[i, j] := (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13094,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,14045,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,14001,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13191,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,17955,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,10899,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,22645,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,22969,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,31034,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,41743,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,73457,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,252064
0,0,0,0,17955,0,0,0,31034,0,0,0,0,0,0,0,0,48989,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,22969,0,0,0,0,22969,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,22969,0,22969,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,22969,0,22969,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,22969,0,0,0,0,0,0,0,22969,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,48989,0,0,0,0,0,50488,0,122446,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,73457,0,0,0,0,0,0,50488,0,22969,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,22969,0,0,0,0,22969,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,122446,0,0,0,122446,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,122446,0,0,0,0,0,122446,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,27192,0,0,0,0,0,27192,0,0,0,0,0,0,0,0,0
0,0,14001,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,27192,0,13191,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13191,0,13191,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13191,0,13191,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,13191,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13191,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,22645,0,0,0,0,0,0,0,0,0,0,0,0,0,0,122446,0,0,0,0,0,0,145091,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,145091,0,159136,172283,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,27192,0,0,0,0,0,159136,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,172283,0,0,172283,0,0,0,0,0,0,0
0,14045,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,172283,0,186328,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,186328,0,0,0,0,0,0,186328
13094,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13094,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13094,0,13094,0,0,0
0,0,0,0,0,10899,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13094,0,23993,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,23993,0,23993,0
0,0,0,0,0,0,0,0,0,41743,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,23993,0,65736
0,0,0,0,0,0,0,0,0,0,0,252064,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,186328,0,0,0,0,65736,0
);

DECISION VARIABLES
        eX[i,j] WHERE (d>0 AND i<j);
        gX[i,j] WHERE (d>0 AND i<j);
        pX[i,j] WHERE (d>0 AND i<j);
```

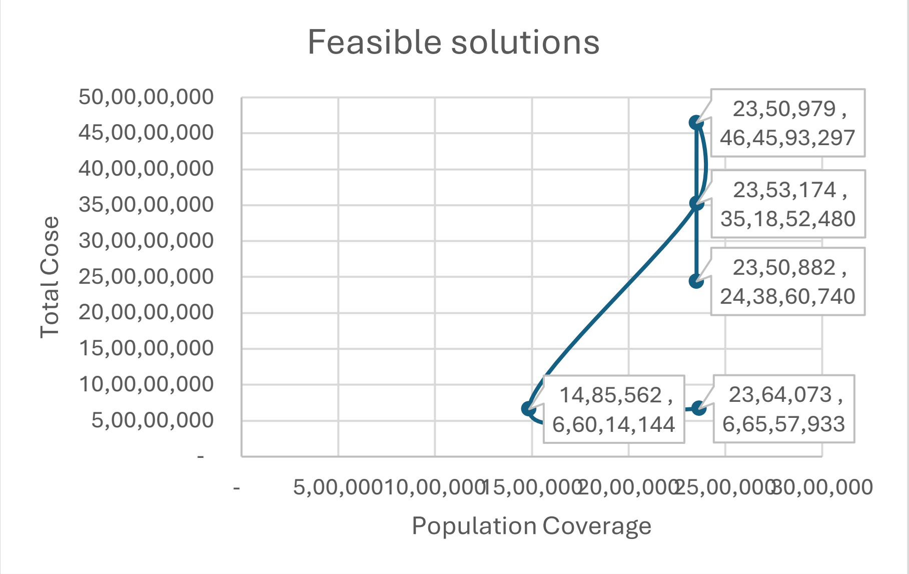
```
MACRO
	PopulationCoverage := SUM(i,j:eX*P WHERE (d>0 AND i<j)) + SUM(i,j: gX*P WHERE (d>0
AND i<j)) + SUM(i,j: pX*P WHERE (d>0 AND i<j));
	OperationCost := SUM(i,j: d*UeOperation*eX WHERE (d>0 AND i<j)) + SUM(i,j:
d*UgOperation*gX WHERE (d>0 AND i<j)) + SUM(i,j: d*UpOperation*pX WHERE (d>0 AND i<j));
	ImprovementCost := SUM(i,j: d*UeImprove*eX WHERE (d>0 AND i<j)) + SUM(i,j:
d*UgImprove*gX WHERE (d>0 AND i<j)) + SUM(i,j: d*UpImprove*pX WHERE (d>0 AND i<j));
	TotalCost := OperationCost + ImprovementCost

MODEL
	MAX PopulationCoverage
	!MIN TotalCost

SUBJECT TO
	!Constraint on assignment on all links
	cnt1: SUM(i,j: eX WHERE (d>0 AND i<j)) + SUM(i,j: gX WHERE (d>0 AND i<j)) + SUM(i,j:
pX WHERE (d>0 AND i<j))>=25;

	!Constraints on assignment on each link
cnt2: eX[1,34] + gX[1,34] + pX[1,34] <=1;
cnt3: eX[2,32] + gX[2,32] + pX[2,32] <=1;
cnt4: eX[3,24] + gX[3,24] + pX[3,24] <=1;
cnt5: eX[4,27] + gX[4,27] + pX[4,27] <=1;
cnt6: eX[5,13] + gX[5,13] + pX[5,13] <=1;
cnt7: eX[6,36] + gX[6,36] + pX[6,36] <=1;
cnt8: eX[7,28] + gX[7,28] + pX[7,28] <=1;
cnt9: eX[8,17] + gX[8,17] + pX[8,17] <=1;
cnt10: eX[9,13] + gX[9,13] + pX[9,13] <=1;
cnt11: eX[10,38] + gX[10,38] + pX[10,38] <=1;
cnt12: eX[11,19] + gX[11,19] + pX[11,19] <=1;
cnt13: eX[12,39] + gX[12,39] + pX[12,39] <=1;
cnt14: eX[13,18] + gX[13,18] + pX[13,18] <=1;
cnt15: eX[14,15] + gX[14,15] + pX[14,15] <=1;
cnt16: eX[14,20] + gX[14,20] + pX[14,20] <=1;
cnt17: eX[15,16] + gX[15,16] + pX[15,16] <=1;
cnt18: eX[16,17] + gX[16,17] + pX[16,17] <=1;
cnt19: eX[18,19] + gX[18,19] + pX[18,19] <=1;
cnt20: eX[18,21] + gX[18,21] + pX[18,21] <=1;
cnt21: eX[19,20] + gX[19,20] + pX[19,20] <=1;
cnt22: eX[21,22] + gX[21,22] + pX[21,22] <=1;
cnt23: eX[22,28] + gX[22,28] + pX[22,28] <=1;
cnt24: eX[23,24] + gX[23,24] + pX[23,24] <=1;
cnt25: eX[23,30] + gX[23,30] + pX[23,30] <=1;
cnt26: eX[24,25] + gX[24,25] + pX[24,25] <=1;
cnt27: eX[25,26] + gX[25,26] + pX[25,26] <=1;
cnt28: eX[26,27] + gX[26,27] + pX[26,27] <=1;
cnt29: eX[28,29] + gX[28,29] + pX[28,29] <=1;
cnt30: eX[29,30] + gX[29,30] + pX[29,30] <=1;
cnt31: eX[29,31] + gX[29,31] + pX[29,31] <=1;
cnt32: eX[31,32] + gX[31,32] + pX[31,32] <=1;
cnt33: eX[32,33] + gX[32,33] + pX[32,33] <=1;
cnt34: eX[33,39] + gX[33,39] + pX[33,39] <=1;
cnt35: eX[34,35] + gX[34,35] + pX[34,35] <=1;
cnt36: eX[35,36] + gX[35,36] + pX[35,36] <=1;
cnt37: eX[36,37] + gX[36,37] + pX[36,37] <=1;
```

```
cnt38: eX[37,38] + gX[37,38] + pX[37,38] <=1;
cnt39: eX[38,39] + gX[38,39] + pX[38,39] <=1;



        SUM(i,j:gX WHERE (d>0 AND i<j)) >=10;
        SUM(i,j:pX WHERE (d>0 AND i<j)) >=10;

        !Budget Constraints
        TotalCost <= Budget;

        !Contraints to find alternative solutions for Multi Objective Optimization
        PopulationCoverage < 2350978;
        !TotalCost > 66014145;

BINARY
        eX[i,j] WHERE (d>0 AND i<j);
        gX[i,j] WHERE (d>0 AND i<j);
        pX[i,j] WHERE (d>0 AND i<j);
```

Solution

Solution	Population Coverate	Operation Cost	Improvement Cost	Total Cost
1	23,64,073	77,933	6,64,80,000	6,65,57,933
2	14,85,562	14,144	6,60,00,000	6,60,14,144
3	23,53,174	72,480	35,17,80,000	35,18,52,480
4	23,50,979	73,297	46,45,20,000	46,45,93,297
5	23,50,882	75,740	24,37,85,000	24,38,60,740

See solution.xlsx for detailed road surface type of links for each solution.



Conclusion and Discussion

Road network of Syangja district was analysed to create a Minimum Spanning Tree (MST) based on the physical distance connecting palika headquarters to the district headquarters. Based on the MST distances and population served by the MST links, a multi-objective optimization was performed using the software "MPL" on maximum population coverage (service) and minimum cost for three different road surface types: earthern, gravel and pitch.

Multiple solutions for the same population coverage are possible with different road surface types. The constraints could be more realistically adjusted to get better and more meaningful optimization results.