
MA5895 : Numerical Optimization

Combined Assignment Report

Cell Tower Problem

Name: Pragnesh Rana

Roll number: ME17S301

Summary : The solution to find out the optimal location of cell tower sites is discussed here by keeping constrain over budget and population. To simulate the given problem, first random population is generated. The whole population is divided into regions using K-Means clustering. The centroid of each region is used as a potential site to plant to the cell tower. Out of all potential sites, the goal is to obtain the optimal locations such that total cost to plant the towers does not exceed the budget, and the maximum population is also served. The formulation is obtained by an instance of mixed-integer linear programming and solved by branch and bound technique using Gurobi's python based solver.

Contents

1 Problem Statement	1
2 Assumptions	2
3 Formulation	2
4 Approach to solve the problem with example	3
5 Implemented algorithm procedure	6
6 Results	7
6.1 Result with few values of input	7
6.2 Largest problem size you can solve	11
6.3 Computational time as a function of input parameters	11
7 References	12

Problem Statement

In city planning, the issue is to find out optimal sites to plant the cell towers given the population and budget constraints. To solve the problem, divide the population into small regions based on the maximum capacity to plant the cell tower. Each site location may also cover other specified numbers of locations. Out of all possible cell tower sites, find out the optimal locations which can serve the maximum population keeping control over the budget.

Assumptions

Assumed that,

- The city is a rectangular region.
- One region can be covered by more than one cell tower as they might have a different frequency.
- The cost of each tower is assumed as 30,00,000 ₹ [1] and other cost is 10% times the population(self+neighboring region) covered by each network.
- Each cellular location(centroid) can cover three neighbouring regions apart from covering its region. (Can be easily changed but in whole analysis kept fixed)
- The euclidean distance is used for measuring distance.
- The resultant location of the cell site is average value of covered centroids (Average value of coordinates of centroid and its neighbouring centroid).

Formulation

Formulation of the problem is divided into two parts.

1. Clusters of population data

The population is divided into K-clusters using k-means algorithm given number potential cites to plant the cell towers. The partition is made in such way that each person is assigned with nearest possible mean location. Or generate K-clusters which minimizes the within-cluster sum of squares(variance).

The objective function for clustering is defined as[2],

$$J = \operatorname{argmin}_s \sum_{i=1}^k \sum_{x \in s_i} ||x - \mu_i||^2 \quad (1)$$

where,

- μ_i = mean of points s_i
- s = Population sets s_1, s_2, \dots, s_k
- k = Total required clusters

2. Optimal location of sites

After obtaining all the clusters, now objective is to find out the optimum locations which can cover maximum population keeping control over the budget. The formulation of problem is an instance of mixed integer programming(MIP)[3].

The formulation is given as,

$$\begin{aligned}
& \text{Maximize } Z = \sum_{j \in R} p_j \cdot y_j \\
& \text{S.T, } \sum_{(i,j)} x_i \geq y_j \quad \forall j \in \mathcal{R} \\
& \sum_{i \in T} c_i \cdot x_i \leq \text{Budget}
\end{aligned} \tag{2}$$

where,

- j - Index of regions $\in R$
- i - Index of optimal sites to build tower $\in T$
- T - Set of potential sites
- R - Set of regions
- c_i - Cost of setting up a tower at site- i
- p_j - Population at region- j
- y_j - Binary variable for assignment of region { 1 if region is covered else 0 }
- x_i - Binary variable for tower { 1 if tower is build else 0 }

In Words: The objective function is to maximize the total covered population constraining over region and budget. The first constrain denotes that all location should be covered by at least one tower. Following, constrain denotes that total cost should not exceed the budget.

The MIP(Mixed Integer Programming) problem is solved by linear-programming based branch-and-bound algorithm[4]. The solution is implemented using Gurobi (academic version) python based library[5].

Approach to solve the problem with example

- In a specified rectangular region, using the limit of the region, population is generated by five blobs with appropriate *sigma* such that spread is enough to cover enough region and 30% of the population is obtained by random spread over a given region.
- In Fundamental of cellular network[6] it was mentioned that cellular network region could be approximated by hexagon so, the whole population is divided into given small number of regions(cluster) using k-means algorithm which follows the shape of a polygon(Approximation of pentagon or hexagon.)
- It was also mentioned that tower near to each other does not have the same frequency so, an overlap of frequency in one region may not be an issue[6]. In a simulation, the tower located at the centroid of one region can cover three(or specified) nearest regions.
- Once all the regions is obtained, the centroid of each region denotes potential site to plant the cell tower.

- Out of all potential cites the optimal sites are obtained such that it can cover the maximum population given budget constrain.
- This problem is another example of maximal covering location problem[3]and set cover problem. The mathematical formulation of the problem is given in section-3 in the form of Mixed Integer Programming(MIP).
- The solution of the problem is obtained by Gurobi Optimizer python-based package. Further detail is given in section-3.
- Let's take an example, Input:

Total Population = 1,30,000

Count of Covering Region = 4 (self+3 Neighbor)

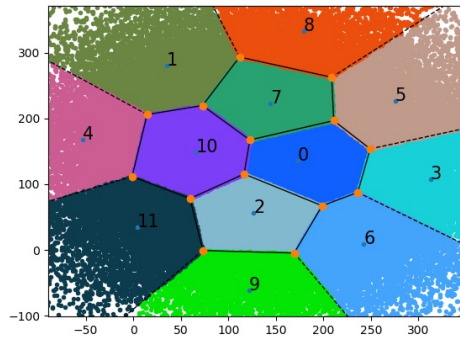
Maximum Capacity to Plant Towers or Total regions = 12

Cost per each tower = 30,00,000 ₹

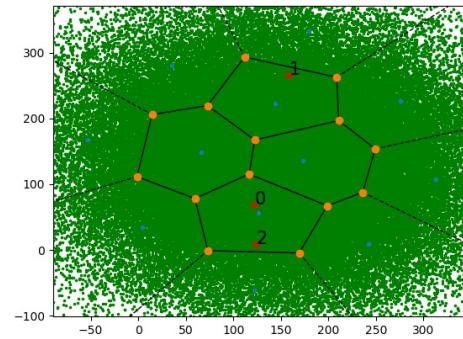
Total budget = 10,00,000 * MAX Capacity + 10% Total Population = 1,20,13,000 ₹

- Whole population is divided into 12 clusters as given in fig.-1(a). Each cluster centroid is a potential site for the final cell location. However, the goal is to find out the optimal location of sites with specified constrains ???. Problem is solved using gurobi solver which works based on maximum covering location algorithm with mixed integer programming.

The obtained result by running the simulation using aforementioned parameters is given below,



(a) Voronoi Diagram



(b) Final cites to plant towers

Figure 1: Fig.-1(a) shows voronoi diagram obtained by diving the whole population into cluster using K-Means algorithm and Fig.-1(b) shows the final cites to plant the towers (By RED MARKED DOTS).

Region	0	1	2	3	4	5	6	7	8	9	10	11	Total
Population	44369	6539	11246	7466	4998	7909	8601	10757	4503	4743	12511	6898	130000
Cost to build (₹)	3034514	3028266	3061623	3060339	3025948	3062592	3023455	3061383	3025205	3026205	3066372	3028755	30418113
Budget(₹)													12013000

Table 1: Summary of population and cost associated with each region along with available budget.

RegionCenter	NeighborCenter0	NeighborCenter1	NeighborCenter2
0	7	2	10
1	7	10	4
2	0	10	9
3	6	5	0
4	10	1	11
5	3	7	0
6	3	2	9
7	0	10	8
8	7	5	1
9	2	6	11
10	0	7	2
11	2	10	4

Table 2: Table of centroid and its three nearest centroid index

- Centroid and its three nearest neighbor centroids are given in table-2 which indicates if cell tower is planted on specific region center, it can also cover three nearest regions along with its region. The objective is to find out the best possible sites which can cover maximum population with restriction of budget. For example if cell tower is planted at centroid location-0 then it can cover regions with centroid location 7, 2, and 10, which can be verified from figure-1(a).
- The problem is solved by running the algorithm considering specified constrain. Obtained result is given in fig-1(b). The centroid 2,8 and 9 are resultant place obtained by algorithm.
- The tower at centroid location-2 can cover region 2,0,10,9. Likewise, tower at location 8 and 9 also covers the regions 8,7,5,1 and 9,2,6,11.
- Final location of cell site is obtained by taking average value coordinates of centroids, as location 2 can cover regions-2,0,10 and 9.
- It is observed that the region 2 and 9 are covered twice. The region 4 and 3 are not covered due to budget constrain.
- Overall, to cover 86.95% population, 75.94% budget is utilized.

Implemented algorithm procedure

Algorithm 1 To find out optimal cell tower location

```
1: population: Size of the population
2: NOB: Number of Blobs
3: NOC: Number of Clusters based on capacity to plant towers
4: DistBtnCentroidNNeighbor : Calculates distance of each centroid to all other centroids
5: procedure GENERATEPOPULATION(population,NOB)           ▷ To generate artificial Data
6:   BlobPopulation = Generate blobs(NOB)                 ▷ Used scikit learn
7:   RandomPopulation = Generate data points in specified limit   ▷ random function
8:   TotalPopulationData = BlobPopulation + RandomPopulation
9:   return TotalPopulationData

10: procedure GENERATECLUSTERS(TotalPopulationData,NOC)
11:   labels , ClusterCentroid = KMEAN(TotalPopulationData,NOC)   ▷ scikit's kmean algo
12:   return labels, ClusterCentroid

13: procedure DISTBTNCENTROIDNNEIGHBOR(ClusterCentroid)
14:   AllocatedFacilityData = Empty           ▷ To store index of nearest specified neighbors
15:   for ClusterCentroid do
16:     Dist = find Distance from all centroids(ClusterCentroid)   ▷ Euclidean distance
17:     AllocatedFacilityData.append(argmin(Dist)[1 : 4])           ▷ Specified neighbors, Here 3
18:   return AllocatedFacilityData

19: procedure CELLTOWERPROBLEM(AllocatedFacilityData,RegionWisePopulation)
20:   GUROBI API( )
21:   Obj = Define Objective Function to cover maximum population
22:   C1= Define Constrains to cover whole population
23:   C2= Define Constrains for Budget
24:   OptimizedLocations = Run API(Obj, C1, C2)
25:   return OptimizedLocations

26: procedure FINALLOCATIONS(OptimizedLocation,ClusterCentroid,AllocatedFacilityData)
27:   FinalLocations = Empty           ▷ Array
28:   for OptimizedLocation do
29:     FinalLocations.append(average(AllocatedFacilityData[OptimizedLocation]))
30:   return FinalLocations

31: procedure SIMULATE( )
32:   TotalPopulationData = GENERATEPOPULATION(population,NOB)
33:   labels, ClusterCentroid = GENERATECLUSTERS(TotalPopulationData,NOC)
34:   VORONOIDIAGRAM(ClusterCentroid)
35:   RegionWisePopulation = SEPERATEDATABYLABEL(labels,TotalPopulationData)
36:   AllocatedFacilityData = DISTBTNCENTROIDNNEIGHBOR(ClusterCentroid)
37:   OptimizedLocations =CELLTOWERPROBLEM(AllocatedFacilityData,RegionWisePopulation)
38:   FinalLocations = FINALLOCATIONS(OptimizedLocation,ClusterCentroid,AllocatedFacilityData)
```

Results

Result with few values of input

- Case:1

Input:

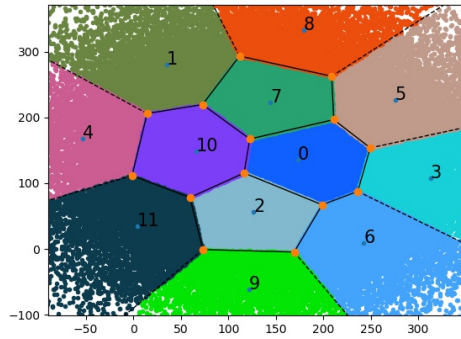
Total Population = 1,30,000

Count of Covering Region = 4 (self+3 Neighbor)

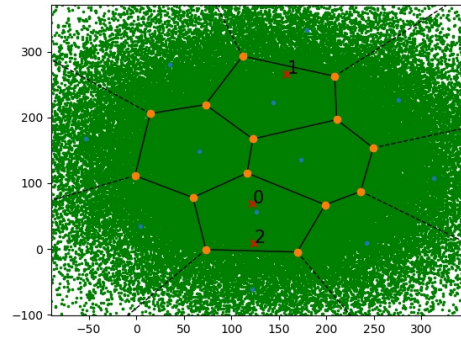
Maximum Capacity to Plant Towers or Total regions = 12

Cost per each tower = 30,00,000 ₹

Total budget = 10,00,000 * MAX Capacity + 10% Total Population = 1,20,13,000 ₹



(a) Voronoi Diagram



(b) Final sites to plant towers

Figure 2: Fig.-2(a) shows voronoi diagram obtained by dividing the whole population into cluster using K-Means algorithm and Fig.-2(b) shows the final sites to plant the towers (By RED MARKED DOTS).

The locations to plant cell towers are obtained by average value of region shown in fig.-2(a) marked by centroid 2,8,9 and its 3 neighboring regions. Final Location is given in fig-2(b) by red dots. **To cover 86.95% of population 75.94% budget is utilized.**

- Case:2

Input:

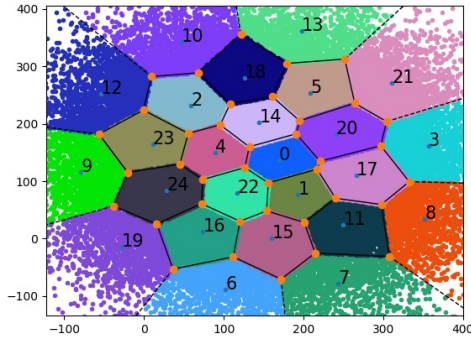
Total Population = 1,30,000

Count of Covering Region = 4 (self+3 Neighbor)

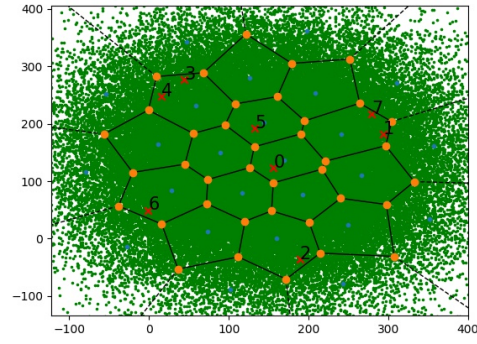
Maximum Capacity to Plant Towers or Total regions = 25

Cost per each tower = 30,00,000 ₹

Total budget = 10,00,000 * MAX Capacity + 10% Total Population = 2,50,13,000 ₹



(a) Voronoi Diagram



(b) Final cites to plant towers

Figure 3: Fig.-3(a) shows voronoi diagram obtained by diving the whole population into cluster using K-Means algorithm and Fig.-3(b) shows the final cites to plant the towers (By RED MARKED DOTS).

The locations to plant cell towers are obtained by average value of region shown in fig.-3(a) marked by centroid 0,3,7,10,12,14,19,21 and its 3 neighboring regions. Final 8 location is given in fig-3(b) by red dots. **To cover 94.55% of population 96.55% budget is utilized.**

• Case:3

Input:

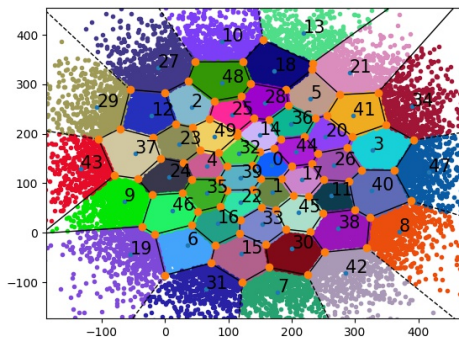
Total Population = 1,30,000

Count of Covering Region = 4 (self+3 Neighbor)

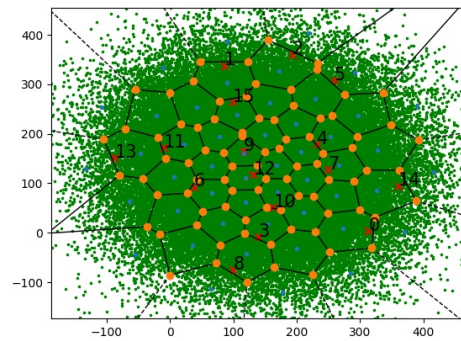
Maximum Capacity to Plant Towers or Total regions = 50

Cost per each tower = 30,00,000 ₹

Total budget = 10,00,000 * MAX Capacity + 10% Total Population = 5,20,13,000 ₹



(a) Voronoi Diagram



(b) Final sites to plant towers

Figure 4: Fig.-4(a) shows voronoi diagram obtained by diving the whole population into cluster using K-Means algorithm and Fig.-4(b) shows the final sites to plant the towers (By RED MARKED DOTS).

The locations to plant cell towers are obtained by average value of region shown in fig.-3(a) marked by centroid 8,10,13,15,20,21,24,26,31,32,33,37,39,43,47,48 and its 3 neighboring regions. Final 16 locations are given in fig-3(b) by red dots. **To cover 97.73% of population 96.27% budget is utilized.**

- **Case:4**

Input:

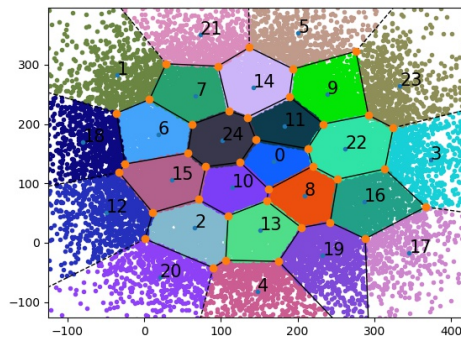
Total Population = 50,000

Count of Covering Region = 4 (self+3 Neighbor)

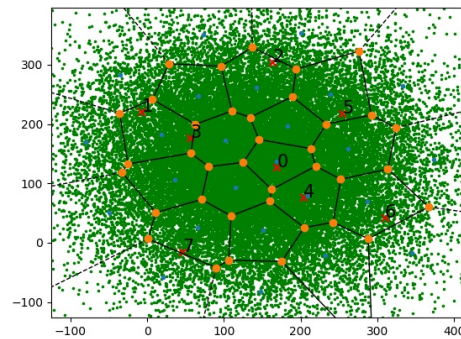
Maximum Capacity to Plant Towers or Total regions = 25

Cost per each tower = 30,00,000 ₹

Total budget = 10,00,000 * MAX Capacity + 10% Total Population = 2,50,05,000 ₹



(a) Voronoi Diagram



(b) Final sites to plant towers

Figure 5: Fig.-5(a) shows voronoi diagram obtained by diving the whole population into cluster using K-Means algorithm and Fig.-5(b) shows the final sites to plant the towers (By RED MARKED DOTS).

The locations to plant cell towers are obtained by average value of region shown in fig.-5(a) marked by centroid 0,1,5,6,8,9,17,20 and its 3 neighboring regions. Final 16 locations are given in fig-5(b) by red dots. **To cover 94.86% of population 96.21% budget is utilized.**

- **Case:5**

Input:

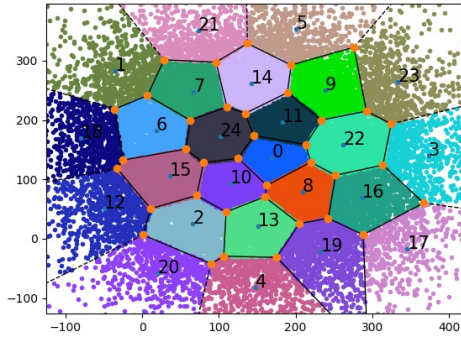
Total Population = 50,000

Count of Covering Region = 5 (self+3 Neighbor)

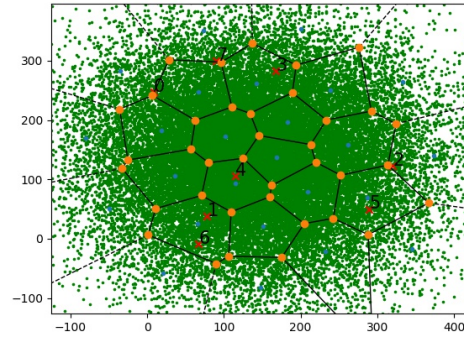
Maximum Capacity to Plant Towers or Total regions = 25

Cost per each tower = 30,00,000 ₹

Total budget = 5,00,000 * MAX Capacity + 10% Total Population = 2,50,05,000 ₹



(a) Voronoi Diagram



(b) Final sites to plant towers

Figure 6: Fig.-6(a) shows voronoi diagram obtained by diving the whole population into cluster using K-Means algorithm and Fig.-6(b) shows the final sites to plant the towers (By RED MARKED DOTS).

The locations to plant cell towers are obtained by average value of region shown in fig.-6(a) marked by centroid 1,2,3,5,10,17,20,21 and its 4 neighboring regions. Final 16 locations are given in fig.-6(b) by red dots. **To cover 100 % of population 96.24 % budget is utilized.**

• Case:6

Input:

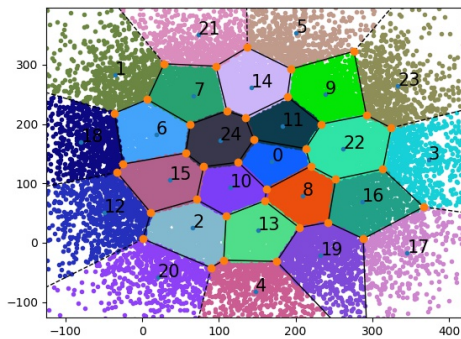
Total Population = 50,000

Count of Covering Region = 4 (self+3 Neighbor)

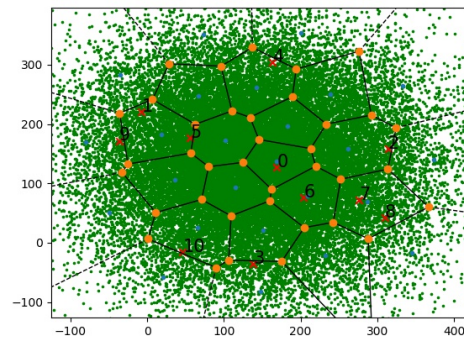
Maximum Capacity to Plant Towers or Total regions = 25

Cost per each tower = 2,00,000 ₹

Total budget = 1,00,000 * MAX Capacity + 10% Total Population = 25,05,000 ₹



(a) Voronoi Diagram



(b) Final sites to plant towers

Figure 7: Fig.-7(a) shows voronoi diagram obtained by diving the whole population into cluster using K-Means algorithm and Fig.-7(b) shows the final sites to plant the towers (By RED MARKED DOTS).

The locations to plant cell towers are obtained by average value of region shown in fig.-7(a) marked by centroid 0,1,3,4,5,6,8,16,17,18,20 and its 3 neighboring regions. Final 16 locations are given in fig-7(b) by red dots. **To cover 98.54% of population 90.53% budget is utilized.**

Largest problem size you can solve

Tried to solve problem for region-size of 10,000(clusters). But got(laptop doesn't have space), MemoryError: Unable to allocate 9.69 GiB for an array with shape (130000, 10000) and data type float64.

The maximum I tried till 1000 clusters. The computational complexity is discussed in next section. Based on computational complexity largest problem size can be determined.

Computational time as a function of input parameters

Out of all input parameters, the major affecting parameter which can affect the computational cost is number of region. As number of region is directly related to matrix size. The computational time mentioned is only measured between time taken to find out final centroid location. Basically, time taken to solve Mixed Integer Programming. The result is given in fig.-8.

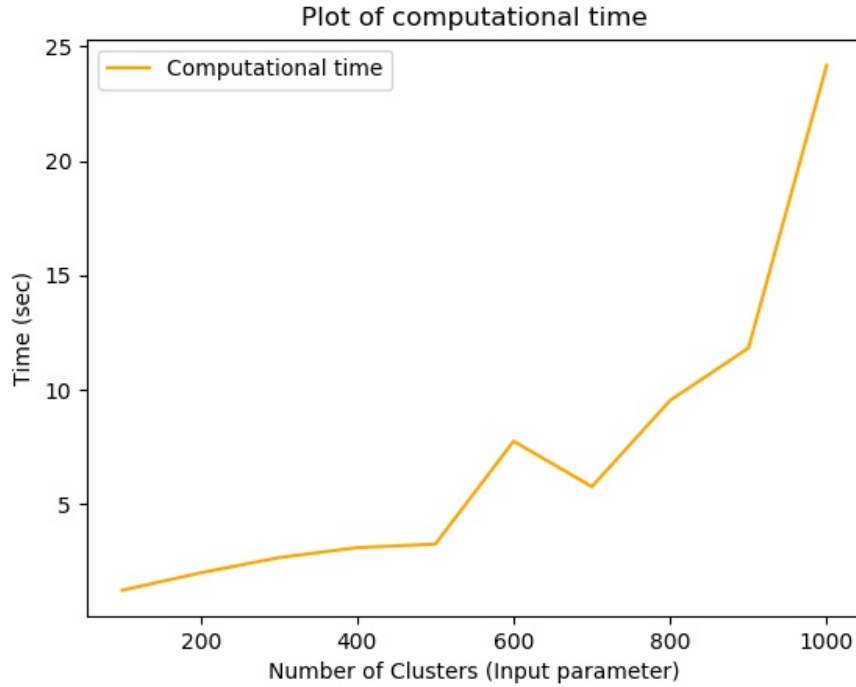


Figure 8: Plot of Computational Complexity (loglog)

Why up & downs in time?(Observation)

With change in number of regions(potencial sites), the orientation and shape of region also get changed. The gurobi solver uses branch and bound algorithm. Along with, it uses some heuristics, cutting plane methods and presolve techniques[4]. Due to change in orientation and shape of regions, the aforementioned techniques get affected computationally which may result in variation of time.

References

- [1] “Ril plans low-cost telecom towers costing just ₹ 1 lakhs,”
- [2] Wikipedia, “k-means clustering,”
- [3] R. Church and C. ReVelle, “The maximal covering location problem,” in *Papers of the regional science association*, vol. 32, pp. 101–118, Springer-Verlag, 1974.
- [4] “Mixed-integer programming (mip) – a primer on the basics,”
- [5] L. Gurobi Optimization, “Gurobi optimizer reference manual,” 2020.
- [6] D. Tipper, “Fundamentals of cellular networks,”