Facility Location Problem

Rahul S Hoskeri
Electrical Engineering and Computer Science
University of California, Merced
rhoskeri@ucmerced.edu

Asoke Datta
Electrical Engineering and Computer Science
University of California, Merced
adatta2@ucmerced.edu

Sungjin Im
Electrical Engineering and Computer Science
University of California, Merced
sim3@ucmerced.edu

I. INTRODUCTION

Clustering algorithm analysis is a very well researched area. K-Means, K-Median and Spectral clustering are few types of these clustering algorithms where K-median clustering is a variation of K-means, where the objective is to minimize the distance from the set of given vertices to the set of chosen center/s.

K-Median problem can be formulated to solve the Facility Location Problem. In this problem we are given a set of clients in metric space and we need assign all the clients to at least one of the facility closest to it among the given set of points to minimize the sum of cost of opening the facility and cost of connecting the clients to these facilities.

In this implementation project we look at three algorithmic type approaches to solve facility location problem.

- Local Search (1-Swap) K-Median heuristics.
- Linear Programming with rounding.
- Primal and Dual approach.

***Index Terms*—K-median, Local Search (1-Swap), Facility Location, Linear Programming, Rounding, Primal and Dual Programming.**

II. PROBLEM FORMULATION

A. K-Median Problem

K-median problem can be stated as the following integer programming equations[1]:
Let V, here $V \in \{1, 2, 3, ...500\}$ represent the set of cities and S, where $S \subseteq V$, denote the set of chosen facilities

Objective:

$$\min \sum_{i,j \in V} x_{i,j} d_{i,j}$$

subject to:

$$\sum_{j \in V} y_j \le k$$

$$\sum_{j \in V} x_{i,j} = 1$$
$$x_{i,j} \le y_j$$
$$x_{i,j}, y_j \in 0, 1$$

$d_{i,j}$= euclidean distance from point i to j
$x_{i,j}$={0,1} decision variable to indicate if point i is served by j center.
y_j={0,1} decision variable to indicate if j is chosen center or not.

B. Facility Location Problem

This K-median integer programming can be modified to facility location problem integer programming by adding the facility opening cost to the objective function.
The integer programming for uncapacitated facility location problem(UFLP) can be stated as below.
Let f be the facility opening cost.

Objective:

$$\min \sum_{i \in V} f y_i + \sum_{i,j \in V} x_{i,j} d_{i,j}$$

subject to:

$$\sum_{i \in V} x_{i,j} = 1 \qquad \forall j \in V$$
$$x_{i,j} \le y_j \qquad \forall i, j \in V$$
$$x_{i,j}, y_j \in 0, 1 \qquad \forall i, j \in V$$

$d_{i,j}$= euclidean distance from point i to j
$x_{i,j}$={0,1} decision variable to indicate if point i is served by j facility.
y_j={0,1} decision variable to indicate if facility j is open or not.

We proceed with these integer programming definitions.

III. System Configuration

IV. Facility Location Problem using Local Search (1-Swap) K-Median heuristics

A. Implementation Details and Modifications

Local search method in simple words is if given a set of medians S already, we try to exchange these medians with points from set V-S only if objective is minimised. Local search generally allows to perform p swaps, which means you can swap p centers at a time and here we only swap 1 center at a time and hence 1-swap local search.

In this project for the initial set of medians we randomly choose centers from given set of vertices V. In general it can be initialised using greedy approach or reverse greedy approach. For simplicity we choose centers randomly here and the algorithm is fully deterministic.

The local search allows us to set a threshold on how low the objective cost decreases from swapping should we allow the swapping of centers. In this case we swap centers if the new objective cost is at least 1% better than the original objective cost.

Here we allow a point to be assigned to itself as it has the least distance between themselves which is 0.

The project implemented here checks for the lowest objective cost for facility location problem by varying $k \in \{1, 2, 3, ...50\}$, $k \in V$ and reports the lowest objective cost for the corresponding k value from the Local Search (1-Swap) K-Median heuristics. The code and implementation can be found in the github repository [2].

B. Observations

K-Median 1-Swap Local Search algorithm was used to obtain the facility opening cost (f) setting k=20, $|V| = 500$ and the cities were provided as latitude and longitude coordinates.

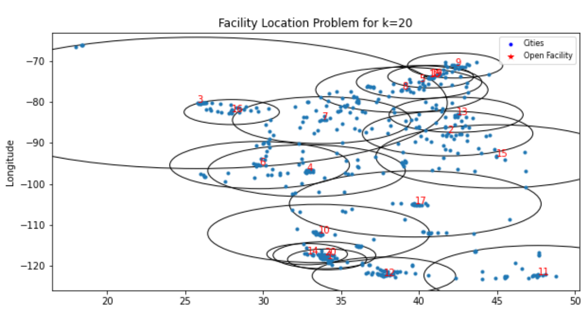


Fig. 1. Facility assignment for randomly chosen centers

The objective cost for k-median clustering before local search 1-swap [Fig 1] was found to be ≈ 1014.90

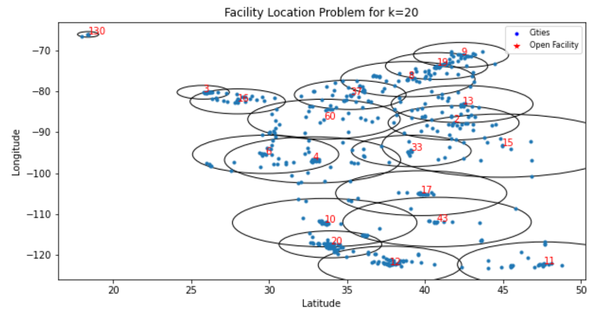


Fig. 2. Facility assignment after 1-swap local search implementation

The objective cost for k-median clustering after local search 1-swap [Fig 2] was found to be ≈ 803.54

Thus facility opening cost was set to be f=(objective cost after local search 1-swap)/20 ≈ 40.

The lowest objective cost for facility location was found at k=17 with objective cost for facility opening cost before 1-swap as $\approx$ [Fig 3] and objective cost for facility opening cost after 1-swap as ≈ 1599.35 [Fig 4].

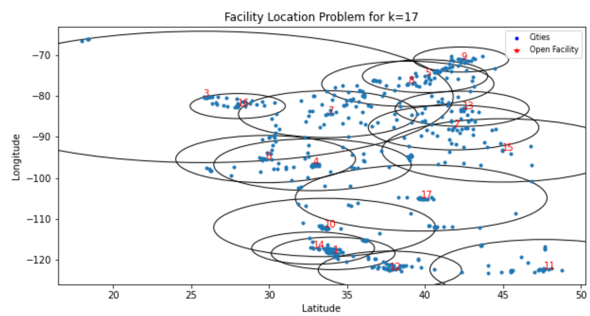


Fig. 3. Facility assignment before 1-swap local search implementation for k=17

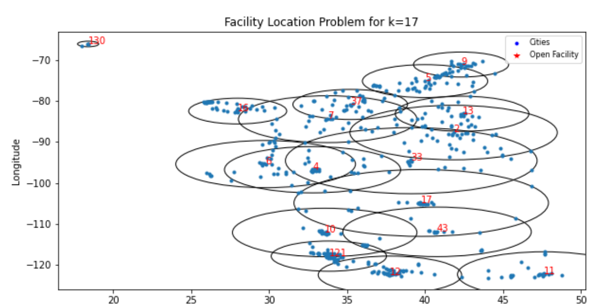


Fig. 4. Facility assignment after 1-swap local search implementation for k=17

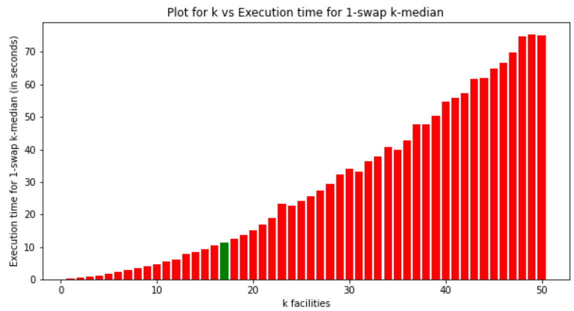


Fig. 5. Facility assignment after 1-swap local search implementation

The execution time for performing the 1-swap local search k median for facility location [Fig 5].The green bar represents the execution time for local optimum found at k=17.

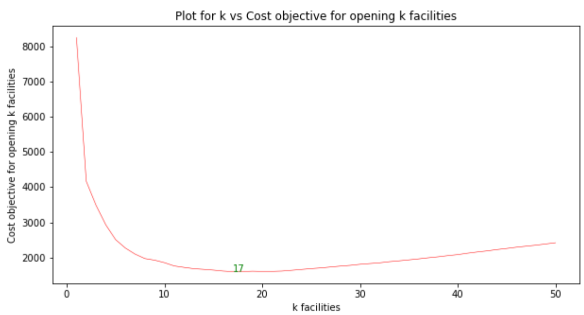


Fig. 6. Facility assignment after 1-swap local search implementation

The variation of objective cost for facility location problem as function of k after 1-swap local search [Fig 6].The number in green represents the lowest objective cost found at k=17.

V. Linear Programming with rounding

To convert facility location integer programming to linear programming we begin by relaxing constraints $x_{i,j}$ and y_j. Updated minimization problem looks like below.

$$\begin{gathered} min \sum_{i \in V} f y_i + min \sum_{i,j \in V} x_{i,j} d_{i,j} \\ \text{subject to: } \sum_{i \in V} x_{i,j} = 1 \qquad \forall j \in V \\ x_{i,j} \leq y_j \qquad \forall i,j \in V \\ x_{i,j} \in [0,1] \\ y_j \in [0,1] \end{gathered}$$

To implement the linear programming, We used Python utility for Linear Programming(PuLP)[3]. By running local search algorithm we set f(facility opening cost) to 40 and all the respective constraints and decision variable required by the PuLP. After running the PuLP solver we get **linear objective of 1687**. Also values for decision variable y_i and x_{ij}. Distribution of y_i values are as follows figure 7

A. *Rounding : 6 Approximation*

The rounding stage proceeds as follows:

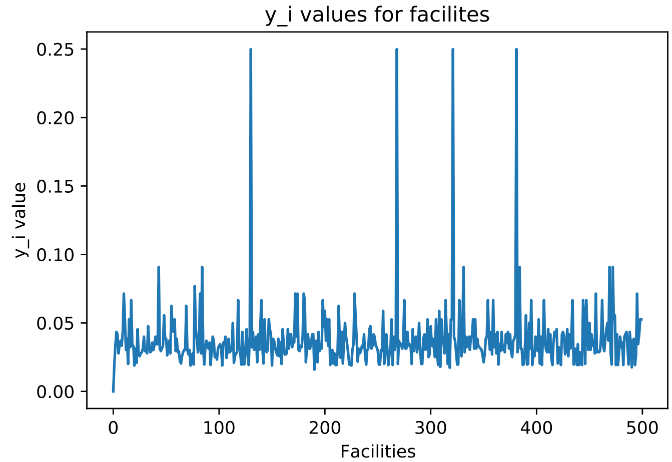


Fig. 7. y values from linear solver

1) First pick the terminal j with the smallest connection cost, Δj , under the fractional solution (x, y). For this terminal, we open the facility.We obtain Δj by

$$\sum_{i \in V} d_{ij} x_{ij}$$

Distribution of Δj values follows figure 8.

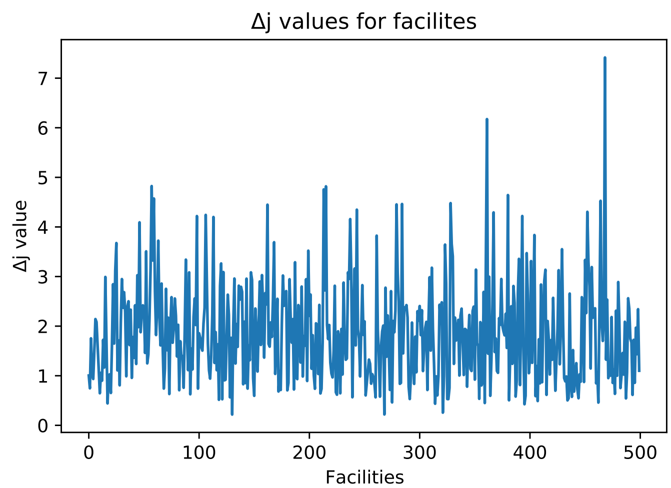


Fig. 8. Δj value distribution

2) We can therefore open the facility j(as facility opening cost is same across all the cities) without increasing the facility opening cost by not opening any other facility in B_j. B_j is defined as follows.

$$B_j = \{i : d_{ij} \leq 2\Delta j\}$$

Value distribution of B_j depicted below Figure 9 . There may be a terminal k such that $j \notin B_k$ and $B_j \cap B_k \neq \emptyset$. We assign such terminals to j.

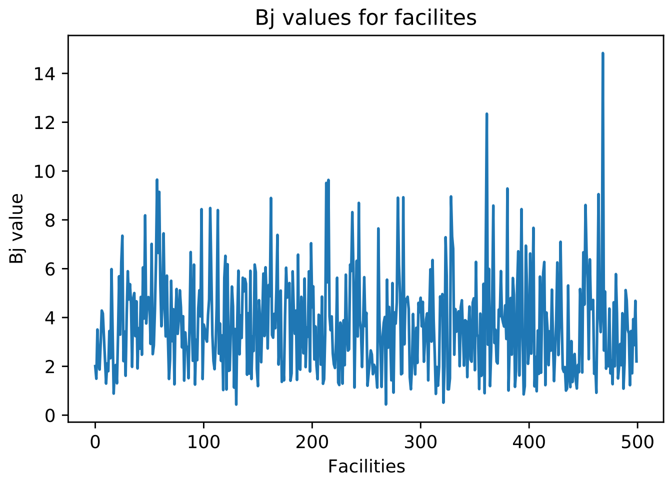


Fig. 9. Bj Value distribution

3) We repeat the above rounding and clustering procedure till all the terminals have been assigned to some facility.

   We achieve an **objective of 2299 which a connection cost 619, where we open in total 42 facilities**. Figure 10 depicts the final picture after running the 6 approximation rounding.

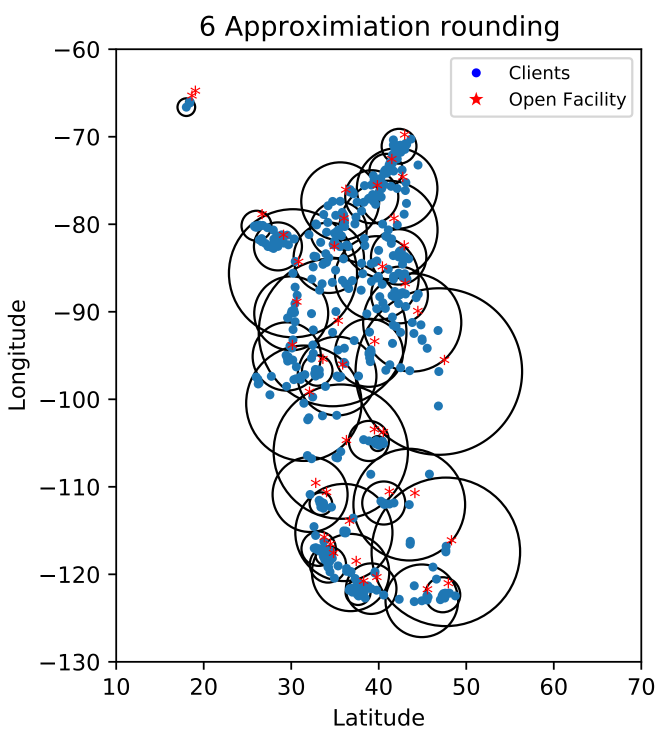
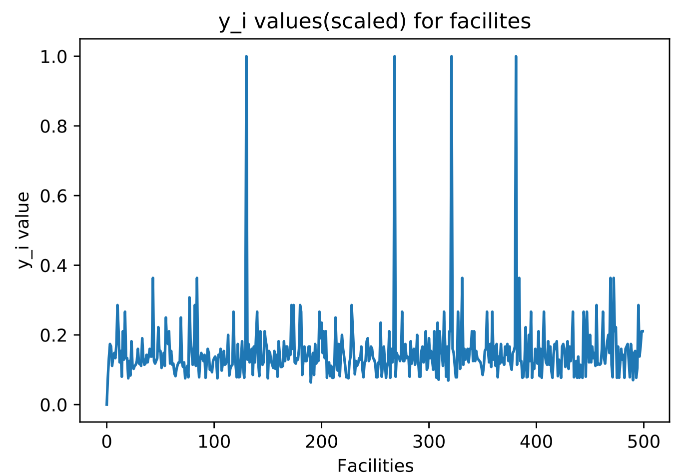


Fig. 10. 6 Approximation Rounding

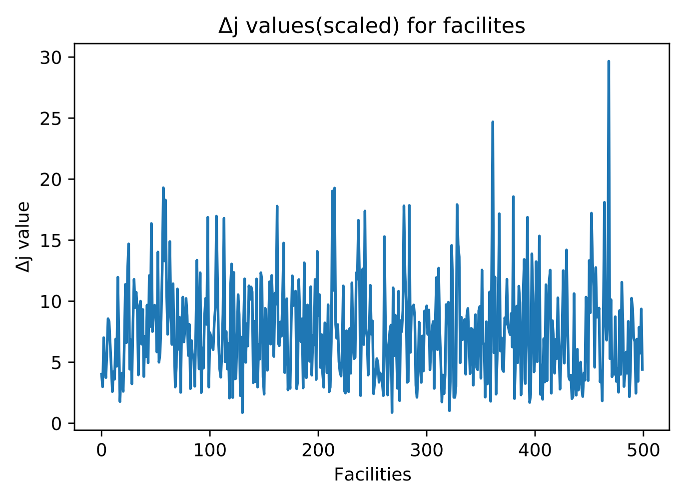
B. *Rounding : 4 Approximation*

In 6 approximation rounding, We chose the radius of the ball B_j around terminal j to be $2\Delta j$. If we reduce this radius, B_j decreases. To satisfy the constraints, we will have to scale the x_{ij}'s (and hence the y_i's) by a larger factor, thereby increasing the facility opening cost. The total connection cost goes down because the candidate facilities are closer to the terminals. We derive the approximation ratio when the radius of ball B_j is $(1+\alpha)\Delta j$ and optimize the parameter α to obtain an improved approximation guarantee. Using Markov's inequality, we find the scaling factor $(1+\alpha)/\alpha$.

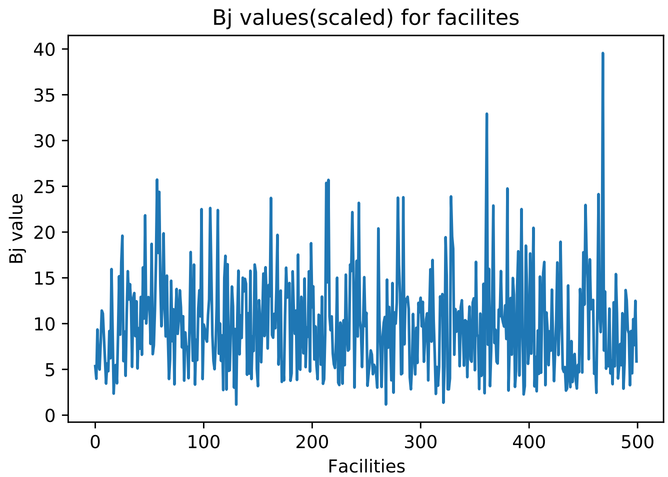
After scaling distribution of y_i is in Figure 11



As we scale x_{ij}, it updates the distribution Δj and B_j. Depicts as Figure 12 and 13



Following rounding and clustering method[4] the approximation guarantee in terms of α is max $\{(1+\alpha)/\alpha, 3(1+\alpha)\}$. This is minimized when $\alpha = \frac{1}{3}$, also yields an 4 approximation solution.



We achieve an **objective of 2004 which a connection cost 1404, where we open in total 15 facilities**. Figure **??** depicts the final picture after running the 4 approximation rounding.

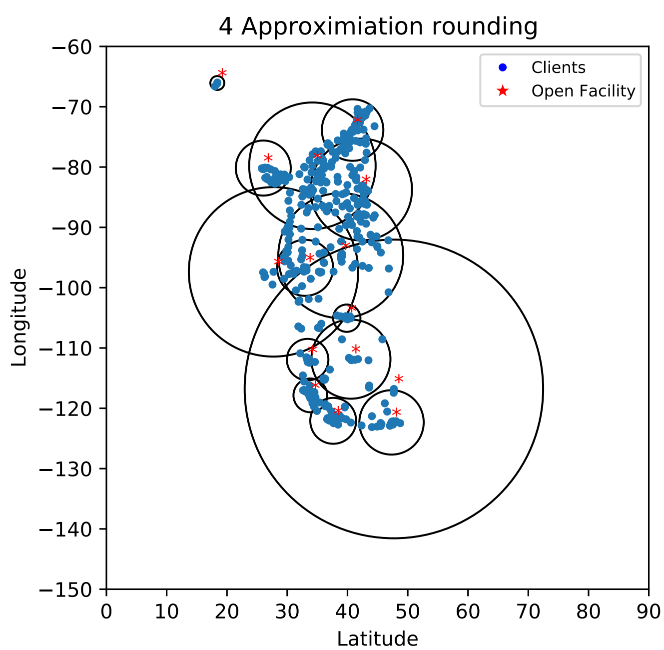


Fig. 14. Facility assignment before 1-swap local search implementation for k=17

VI. Results

A. Facility Location Problem using Local Search (1-Swap) K-Median heuristics

The lowest objective cost for facility location problem after varying k from $\{1, 2, ...50\}$ using 1-swap local search k median facility location problem was found to be ≈ 1699.35 for $k = 17$. Using facility opening cost of $f \approx 40$.

B. Linear Programming with rounding

By running linear solver we obtain objective of 1687 with connection cost 887 and facility opening cost 800. After 6 approximation rounding, we obtain objective of 2299 where connection cost is 619 and we open in total 42 facilities (Total cost 42*20) . And, in 4 approximation rounding, we obtain objective of 2004 where connection cost is 1404 and we open only 15 facilities(Total cost 15*20).

Acknowledgment

We would like to thank Professor Sungjin Im for the problem formulation and providing the dataset and also for his help and guidance.

References

[1] UCSD K-median Clustering notes. https://cseweb.ucsd.edu/~dasgupta/291-geom/kmedian.pdf
[2] Local Search (1-swap) K-Median heuristics code. https://github.com/Rahul664/EECS-279_Facility_Location
[3] PuLP urlhttps://coin-or.github.io/pulp/main/index.html
[4] Facility location via deterministic LP rounding by Anupam Gupta.
[5] https://github.com/Rahul664/EECS-279$_Facility_Location$

**** All the codes are available in github[5].**