

Application for facility location problem in Waste Management with K-Mean clustering algorithm

Author: Albert Khaidarov



Outline

- Problem description
- Methodology
 - K-means clustering
 - Solving FLP with k-means
- System requirements
- Technical choices
- Work plan
- Demo



Facility Location Problem

- Facility location problem is fundamental problem studied in operational research and theoretical computer science.
- Applications in:
 - data mining, image processing, information retrieval, etc.
- Many economical decision problems concern selecting and/or placing certain facilities to serve given demands efficiently.
- Examples:
 - Setting up manufacturing plants, distribution centers, hospitals, fire stations, etc.
- FLP is used in Waste Management for determining optimal locations for processing waste.



Problem definition

- Each location includes a capacity
 - The locations and capacities are garbage collection facilities
 - Their capacity is the annual amount of waste accumulated in tonnes
-
- The result of the clustering will be an optimal placement of k facilities that process the accumulated waste
 - The optimization minimizes the distance between accumulation sites and processing plants, with considering their capacities



K-means Clustering Algorithm

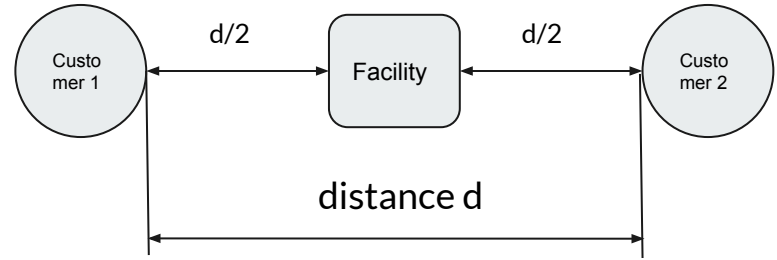
Finding the k cluster centers and assign the objects to their nearest cluster center

- K is the number of vehicles
- Centroid-based clustering
- Finding the k cluster centers and assign the objects to their nearest cluster center
- Gets the minimum squared distance
- Each customer is assigned to exactly one cluster

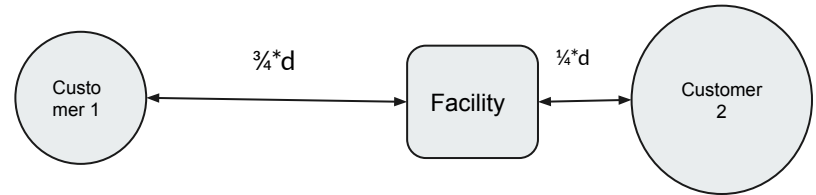
In case if we have capacities for each customer, we can use “center of mass”

Center of mass: example

When customers have equal capacities: $c_1 = c_2$



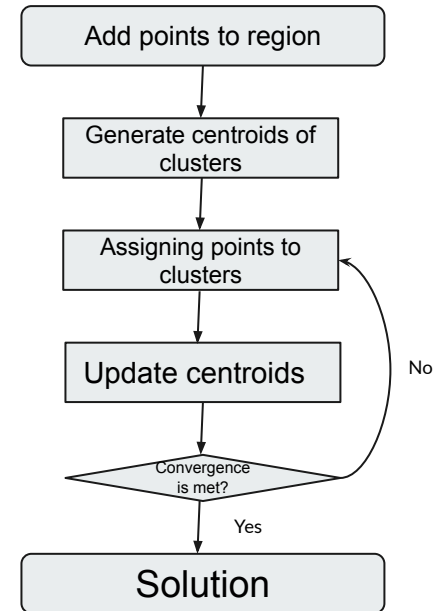
When customers have different capacities: $c_1 = \frac{1}{2} * c_2$



Steps in k-means for FLP

1. Choose the number of clusters, k
2. Randomly generate centroids of clusters,
3. Assign each point to the cluster based on its capacity and distance to centroid.
4. Recompute the new cluster centroids.
5. Repeat steps 3 and 4 until convergence criterion is met.

As a solution we have coordinates of clusters and cluster centers.





System requirements

- Map
 - scale, zoom and move
- Graphical interface
 - drawing on the map
 - choosing number of clusters
 - choosing number of accumulation sites
- Run in parallel
 - multithreading



Technical choices

- Programming languages
 - Java
 - JavaFX WebView
 - Node JS
 - Openlayers library



Work plan

- Programming part
- Testing
- Technical documentation
- User documentation



Thank you!

Any questions?