Minimizing patient risk in location-based scenarios

Awais Akram

Master's Thesis

UNIVERSITY OF
EASTERN FINLAND

School of Computing

Computer Science

July 2021

UNIVERSITY OF EASTERN FINLAND, Faculty of Science and Forestry, Joensuu
School of Computing
Computer Science

Awais Akram: Minimizing patient risk in location-based scenarios
Master's Thesis, 41 p.

Supervisors of the Master's Thesis: Professor Pasi Fränti and Dr. Radu Mariescu-Istodor

July 2021

Abstract: We consider someone to be *at risk* if they cannot reach a specific place in a specified amount of time. More specifically, we will consider coronary heart disease patients in Finland who must reach the nearest hospital in 90 minutes or less otherwise the consequences could be fatal. To solve this problem, we use clustering; a Machine Learning technique that groups data based on similar features. K-means is a very popular partition-based clustering algorithm. However, k-means cannot be applied to the locations as such, as that would result in minimizing the distance *as-the-crow-flies*. In this work, we present efficient methods to incorporate road-network travel times into the clustering process to more accurately model the risk threshold. We experiment using patent information from Finland and demonstrate how clustering can be used to find a better location for hospitals which reduces the risk by 4% (135 people at risk compared to the recent situation, 832).

Keywords: Clustering algorithms, K-means, Mean squared error, Nearest neighbor trick, K-medoids, Sigmoid.

CR Categories (ACM Computing Classification System, 1998 version):

Acknowledgments

In the name of Allah, the most gracious and the most merciful. All praises to Allah and his blessing for the completion of this thesis. I thank God for all the opportunities, trials, and strength that have been showered on me to finish writing the thesis. My humblest gratitude to the Holy Prophet Muhammad (Peace be upon him) whose way of life has been continuous guidance for me.

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Pasi Fränti for the continuous support of my research, for his patience, and immense knowledge. He helped me come up with the thesis topic and guided me in all the time of research and writing of this thesis. It has been a great pleasure and honor to have him as my supervisor. I am deeply grateful for all of his care for me during my stay at UEF.

Besides my supervisor, I am highly thankful to my thesis instructor Dr. Radu Mariescu-Istodor, for his precious suggestions and insightful comments, stimulating discussions, and all the support which gave me great help in improving my work and completing my master's thesis.

I would like to thank Dr. Oili Kohonen who gave me this opportunity to study the IMPIT program at the University of Eastern Finland and helped me with planning my courses.

I'm deeply indebted to my Uncle Eisa Bin Bashir whose thoughtful consideration and guidance helped me in all the difficult phases of my academic and professional life.

I would sincerely like to thank all my beloved friends who were with me and supported me through thick and thin. Most importantly I would like to thank Saim Tanveer and Hassan Shahid for their valuable suggestions and constant support.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support, kind prayers, and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

May God shower the above-cited personalities with success and honor in their life.

Abbreviations

API	Application programming interface
C	Centroid, mid-point of a cluster
K	Number of clusters
KKJ	Finland uniform coordinate system
KM	K-means, a clustering algorithm
MSE	Mean squared errors
P	Partition
RS	Random swap, a clustering algorithm
SSE	Sum of squared errors
TSE	Total squared errors
WGS84	World geodetic system 84

Contents

1 Introduction

Geolocation clustering is a Machine Learning technique that involves the grouping of data present in a geographical coordinate system. Clustering algorithms can be computationally intensive when playing with a large amount of data. K-means algorithm is a very common and well-known partition-based clustering technique that accommodates and supports a large amount of data for clustering. Researchers have proposed many variants of k-means to increase speed, accuracy, and results. To explore the suitable variant with the dataset of thousands of locations, we will evaluate how to optimize it by using different distance methods. Implementing the random swap through k-means with initial centroids as original health center locations will lead to the optimization and choosing the center points for the subgroups in our datasets. It will also help us to study and compare it with other distance methods i.e., bird distance, estimated travel distance, estimated travel time, and sigmoid estimated travel time.

1.1 Motivation

This research is purely focused on a collection of patients which is also known as the STEMI dataset. A total number of 17,346 patients and 22 real health center facilities are used to cluster and present on a web application. STEMI dataset contains patients that are suffering from myocardial infarctions (heart disease) resulting in patient expiry or other fatal consequences. The major objective of this research is to relocate the health centers in such a way that the number of patients at risk becomes lower as compared to the current situation. The research was conducted by trying different variants of distance methods and techniques to observe what is beneficial for saving lives that are at risk. Firstly, the preliminary task was to minimize the total bird distance and see how the at-risk patient count changes. Utilizing bird distance was also helpful for comparing results at the end of our research. Bird distance can be helpful if patients are transferred to the nearest health facility using air medical services such as a helicopter ambulance. Secondly, estimated distance-travel distance was minimized to observe if that is related to the patients that are at risk. Thirdly, estimated travel time was minimized to accommodate the maximum number of patients to be within reach of the nearest hospital. In the very last we took objective function sigmoid into account.

One major problem is to determine and calculate the distance and time from the patient to the health center. One option is to use the true road network cost calculation which is offered by many third-party applications, such as Google Maps, and Open-Source Routing Machine, but that is costly in terms of money and time. No doubt, the above cost calculation is real-time and accurate but dealing with a large collection of dataset requires hours to calculate the road cost.

Currently, there are 832 patients at risk in Finland if we assign them to the nearest original health center facility. These patients are not able to reach hospitals within 90-minute of time.

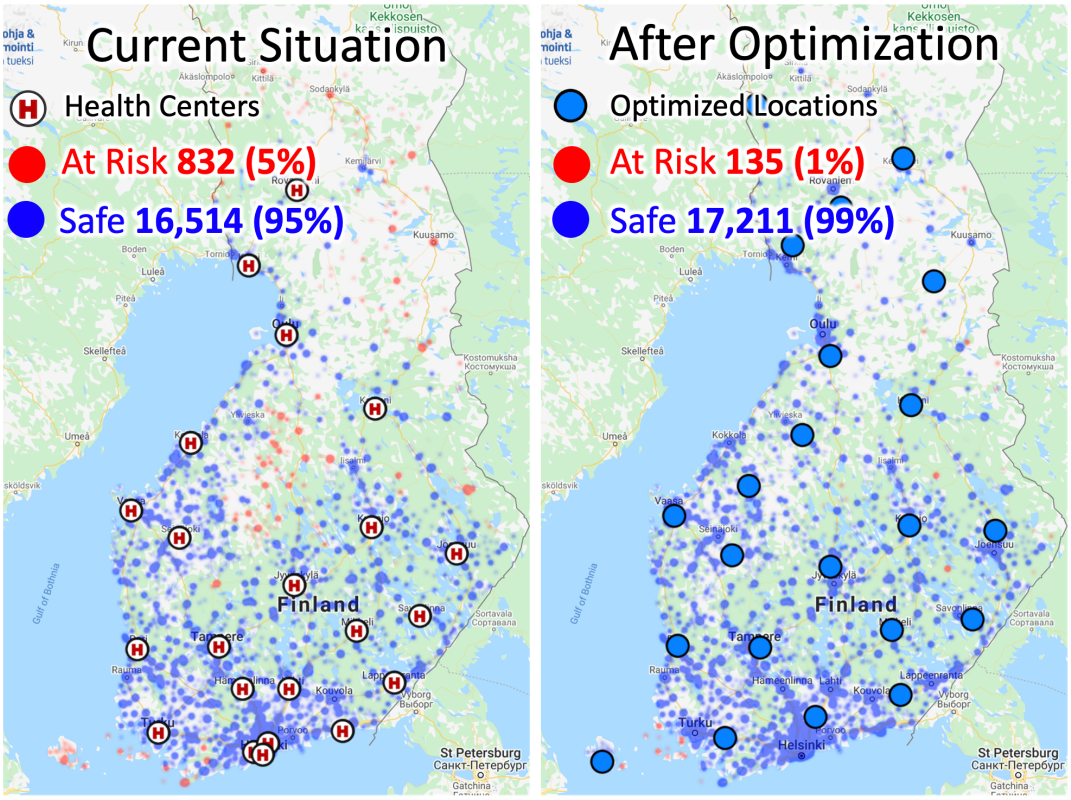

Figure 1. Risk situation in Finland before and after the relocation of health centers.

Figure 1 explains how the risk has changed after optimization. If we look at the left map, we can see the current position of health centers located in Finland. On the contrary, the right-side image contains the positions of relocated health centers after optimization. The areas on the map having red color indicate the patients that are unable to reach the hospital within 90-minute of timespan. These people are at serious risk. The red color can be seen a lot in southern Finland, Jyväskylä, Oulu, and Lapland.

One reason for risk between Oulu and Jyväskylä is that health center facilities are located in the cities. People who are in that middle region or the countryside areas have to travel to Oulu or Jyväskylä depending on which city is near to them. This could be fixed by creating a new hospital between these two cities, but this requires resources such as money, time, and land permission. We could do better by relocating health center facilities by our algorithm.

The risk drops to just 1% resulting in only 135 patients at risk out of 17,346 patients. One important thing to note here is we did not add a new hospital when we optimized using our algorithm. We just relocated the health center positions to save people who are at risk.

If we talk about the same example of the risk that we mentioned above was between Oulu and Jyväskylä, we can see that the region between them becomes blue indicating no more patients at risk. This happens because the health center located in Kokkola was moved towards the east. People who are in that region now have to travel to Kokkola which is less than 90 minutes away. Another observation made was the health centers moved to more peripheral areas like out of the city with our proposed method. Relocation of the health center is expensive in terms of money and resources.

1.2 Literature review

Using Euclidean distance as a distance function in an optimization algorithm is not always significant [Yiu & Mamoulis, 2004]. Using road networks as a distance function provides more meaningful relocation of health facilities. They are not that time-efficient like Euclidean distance, but the results are of great importance and meaningful. It takes several constraints while performing optimization such as traffic, route diversion, rough terrain, and elevation. For incorporating real road-network in optimization algorithm we used a fast travel-time estimator [Mariescu-Istodor and Fränti, 2021] which give results faster and accurate as compared to real road-network values.

Location Analysis is required when planning to relocate health facilities [ReVelle & Eiselt, 2005]. Location and layout problems are both vital and need to be taken into account when planning about relocation. The location of the health facility for

repositioning and their layout helps in making a critical decision. There could be a lot of patients visiting the health center in case of emergency and relocating it makes the situation worse. The capacity of the healthcare center also matters if we talk about highly populated regions such as southern Finland. The health facility either be too large in an area where there are not enough patients or too small in a highly populated area. The locking mechanism of health facilities is helpful if the decision-maker needs to perform relocation on some pre-selected health facilities. Our optimization algorithm has this feature to lock some health facilities when optimizing and relocating only unlocked health facilities in the optimization process.

The discrete hierarchal location model for planning public facilities features the demand level of facilities [Teixeira & Antunes, 2008]. The maximum and minimum capacity constraints are also present along with assigning users to facilities. The single assignment like assigning one patient to one health facility and nearest assignment, assigning the patient to its nearest facility are taken into account. Optimization for the algorithm gets hard when there are several constraints applied to it. Optimization aims to achieve cost minimization, and accessibility maximization if we model it according to our needs. We have modeled the optimization for risk minimization so that a maximum number of patients should be in the reach of their nearest health facility.

Clustering with meaningful constraints like knowing road network infrastructure with traffic information is preferred [Wenting, Jun, & Zhijian, 2009]. It has been used in many real applications nowadays. It helps in knowing the rural and urban space areas and how the road network is implemented and can help in meaningful results. These constraints are defined by the user. The usage of Overhead Graph returns us both travel distance and travel from location p to location q with a very small error rate. The results are reliable and retrieved in a much faster way as compared to other third-party services like Google Maps or Open-Source Routing Machine. Both have pros and cons like time consumption and too costly in terms of pricing.

Two models were suggested to handle the uncertainty in the strategic planning of healthcare facilities [Mestre, Oliveira & Barbosa-Póvoa, 2015]. The model helps when there is no complete information present or there are some uncertain parameters. Healthcare planners may then focus on thinking and using the model for the initial

setup of relocation or assigning and allocating patients to the healthcare facilities. Running the optimization without preparing for it may cause consequences. In our research, if we don't model the at-risk, getting the required output is ambiguous. So the at-risk definition, by modeling the risk as 90 minutes are needed to get the accurate results.

Optimization objective may include minimization of total travel time [Zarrinpoor, Fallahnezhad & Pishvaee, 2018]. Our research is also linked to the total travel time along with the modeling of risk. Uncertainties such as provider side uncertainty (from decision-maker), receiver side uncertainty (from optimization algorithm), and in-between uncertainty could occur leading to failure and making the worst decisions. Stochastic programming and a robust system of optimization can overcome these uncertainties. Our dataset contains historic data from 2015-2018 which is used for stochastic programming. For robust optimization, there are always default or predefined techniques in the optimization code to work properly if there are incomplete parameters.

Displaying a large amount of data on a web mapping application is difficult [Rezaei and Fränti, 2018]. It can be done by clustering the data when displaying resulting in good visualization. Common problems occur while displaying data like, map clutters if the data is not clustered when displaying and loading all the locations and displaying them on the map requires data transmission over the internet. The slowness of the web map application disturbs the user experience and should not be ignored. The clustering tool was then used in this research to visualize the healthcare centers before and after optimization which helped us to display the health facilities on the map.

2 Clustering

Clustering is the process of subgrouping items that are similar. Geolocation clustering is the process of clustering geolocations that are near to each other in multiple groups. These groups containing all geolocations are then called clusters and their middle point is known as centroid. There are various types of clustering. Each type has its advantages and disadvantages. Choosing the right clustering type for our research problem is important. Some famous types of clustering are:

1) Connectivity-based clustering
2) Centroids-based clustering
3) Density-based clustering
4) Fuzzy clustering

Connectivity-based clustering is a method of unsupervised learning clustering. Most commonly, top to bottom approach is used for clustering a set of items that are in a hierarchy. This clustering order is pre-defined. The clusters are obtained after the decomposition of objects based on their hierarchy. This is also known as hierarchal clustering [Banitaan, Nassif, and Azzeh, 2015].

Dealing with different cluster sizes, shapes, and densities while performing clustering is difficult. A graph-theoretical clustering method can be used to overcome these problems. The problem is divided into two subproblems that are separated cluster problems and touching cluster problems. The problems were overcome in two phases. First, by using two round minimum spanning trees that create a graph and find clusters that are separated. Second, clusters that have been formed in the first phase are partitioned to check if there is an overlap happening between them [Zhong, Miao, and Wang, 2010]

Centroid-based clustering is known to be the most efficient way of doing clustering. Different distance functions can be used for finding the distance from points to the centroids and assigning them in relative clusters. The *number of clusters (K)* should be defined before doing clustering. We can either assume the *K* when performing centroid-based clustering or can use some techniques to find an accurate number of clusters for our dataset. Centroid-based clustering is also efficient when working with a large number of datasets [Omran, Engelbrecht, and Salman, 2007].

Density-based clustering helps in making clusters that are in geometrical shape. These shapes could be circular or elliptical. Most of the time there is inconsistency in our dataset, which is also known as noise or outlier and it cannot be ignored. This type of clustering helps us in generating clusters that are inclusive of noise [Steinbach, Ertöz, and Kumar, 2004].

Fuzzy clustering helps in assigning one data point to multiple clusters which is normally not done in other clustering algorithms. The data point is always assigned to the nearest cluster it belongs. But in the case of fuzzy clustering, we have the option to assign one data point to multiple clusters. This type of clustering is mostly used when the dataset has a high number of overlapping data points [Bezdek, 1981].

2.1 K-means and variants

K-means algorithm is the widely used and most simple clustering algorithm. K-means uses centroid as the representative while performing clustering. This algorithm helps in classifying the given dataset into a given number of clusters *K*. Each cluster is assigned a cluster center which is known as centroid. Centroids are randomly chosen *K* points in the dataset. These centroids are placed far away from each other. Now, each point that belongs to its cluster is being assigned till no point is left unassigned. The main objective of K-means is to minimize the Total Squared Error (TSE). K-means averages the locations resulting in lesser TSE when used with Euclidean Distance [Malinen, Mariescu-Istodor, and Fränti, 2011].

There are multiple variants of K-means that are used to achieve multiple objectives. K-medoids is a variant of K-means in which the representatives are medoids. These medoids are restricted to be from the dataset points.

K-medoids use medoids as the representative in the algorithm and they are randomly chosen points from the dataset i.e., patient locations. The advantage of using K-medoids over K-means is that we choose the new representative from the clusters resulting in no health centers forming over lakes. Patients do not live on lakes, so one thing is clear that we do not have final centroids forming over a lake ever. Another benefit

is that K-medoids is more flexible because we can choose several different distance methods according to our needs.

2.2 Random swap

[Fränti, 2018] Random swap algorithm minimizes the Total Squared Error TSE by making random swaps in the centroids. K-means or K-medoids is used as a component in random swap after performing random swap on centroids algorithm. K-means algorithm iterates over a number of iterations *n* and calculates the TSE and checks if it's smaller than the previous one. If the TSE is smaller that means the current clusters are somewhat good and we keep it and iterate again to see if we can find a better combination of clusters than we currently have. K-means usually converge very faster with the Euclidean distance. Strange behavior from the random swap is expected if we use other distance methods and TSE may start to increase in some cases also.

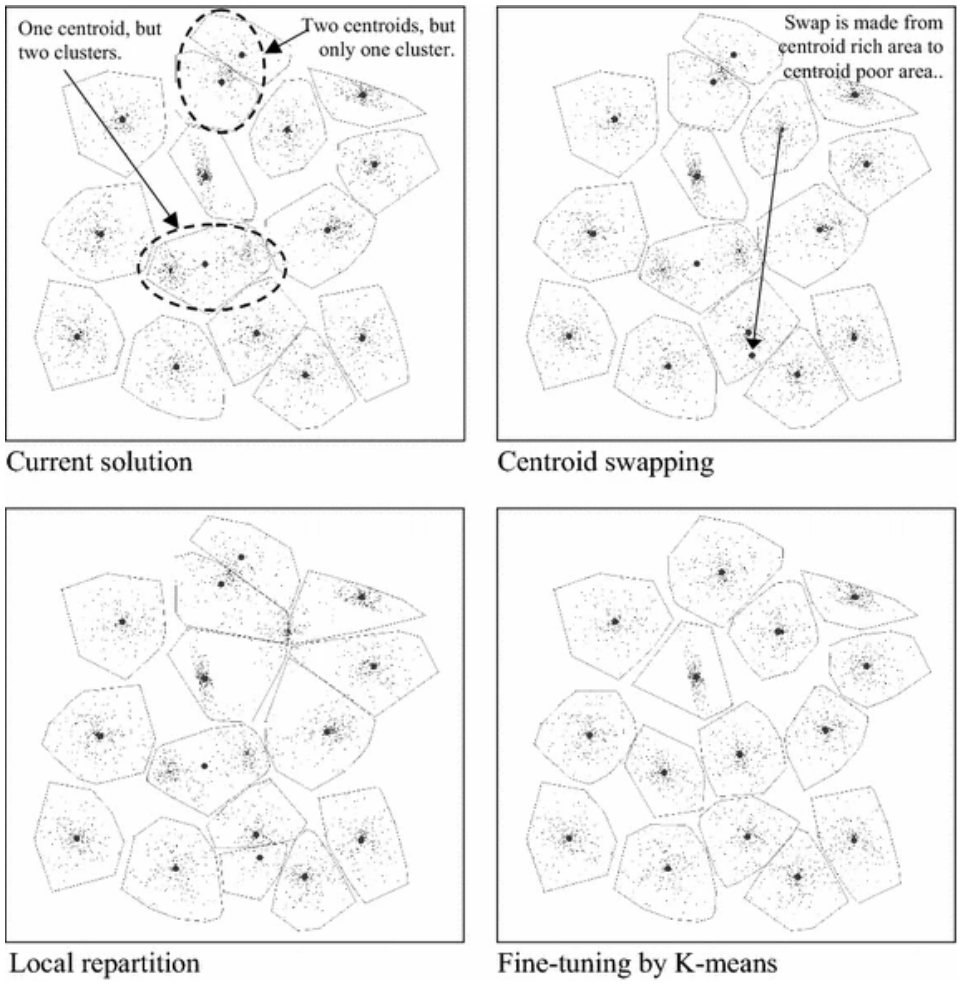

Figure 2. Demonstration of random swap algorithm and other applying K-means afterward.

Figure 2 explains the working of the random swap algorithm. The current solution shows that the clusters are poorly drawn as we can see that one centroid is not enough for holding data of two clusters, and for one cluster there are two centroids available. Random swaps are then performed on different locations to balance the clusters. One of the centroids is swapped to a place where another cluster can be created. Local repartitioning is made after the swap and K-means is applied for fine-tuning of the centroids [Fränti, 2018].

The pseudo code for the random swap is presented below:

```
Random-Swap (X, L) → (C, P)

   Input:  the number of clusters X

           original dataset locations L

   Output: cluster centers C = {c1,...,cX}

           cluster partitions P = {p1,...,pX}

   C ← random representatives (X)

   P ← optimal Partition (X, L, C)

   currentError ← MSECalculation (L, C)

   REPEAT T times

           (Cnew, j) ← Random-Swap (X, C) // new centroids

           Pnew ← Local Repartition (X, Cnew, P, j) // new partition

           (Cnew, Pnew, newError) ← K-means (X, Cnew, Pnew) // Run K-means

           IF currentError < newError THEN

                       (C, P) ← (Cnew, Pnew) // replace better centroids

                       currentError ← newError

   RETURN (C, P)
```

Typically, Euclidean/Bird distance is used as a distance method in random swap algorithm. Euclidean and Bird distance method doesn't fit well because these distances are straight line or angle, in other words, *crow-fly* distance. Patients are not birds that can fly from their homes to hospitals. Another reason is Finland's road structure, there are a lot of lakes and sometimes there is construction/road maintenance work happening which will result in diversions, possibly by taking a longer route than usual.

2.3 Distance

Distance is a very core thing when performing Geolocation clustering. Geolocation coordinates are commonly present in WGS84 format i.e., latitude and longitude. The unit for the WGS84 system is degree (º). The result of converting 1º latitude to kilometers depends on how far that latitude is from north or south. On the other hand, we have Finland uniform coordinate system also known as KKJ which deals with the GPS values in eastings and northings. The unit for easting and northing is measured in meters (m).

2.3.1 Bird distance

Bird distance also known as the Great Circle distance, is an angled distance from point A to point B. The haversine formula is used for calculating the great circle distance. Bird distance suits well with the WGS84 coordinates because we can project them on the map i.e., Google maps. The first parameter of the point is the latitude, and the second parameter of the point is considered as the longitude of the given location. As the Earth is nearly spherical it is meaningful to project the points on a map with a very less error rate of 1% on average providing a good approximation of the distance between two points.

2.3.2 Euclidean distance

The STEMI dataset contains geolocations in the WGS84 coordinate system that are passed to the clustering algorithm for clustering. Euclidean distance is meaningless if we pass WGS84 coordinates i.e., latitude and longitude. To use Euclidean distance, the WGS84 coordinates should be converted to KKJ format. KKJ is a Finland uniform coordinate system. In KKJ format the coordinates are labeled as easting and northing.

Figure 3. KKJ Finland Uniform Coordinate System.

Figure 3 shows the projection of KKJ coordinates on a map. Northing and easting increase from bottom to top and left to right respectively. Northing and easting are measured in meters and make sense when passed in the Euclidean distance formula.

Performing clustering with KKJ coordinates and using distance function as Euclidean can give good results and also helps in projecting KKJ coordinates on a map.

2.3.3 Road distance

Road distance can also be used as a distance function for clustering but when working with a large number of datasets there are limitations like time and performance. It's possible to incorporate a real road network into the clustering algorithm but calculating distance and time from each point to each centroid where our dataset contains thousands of locations it's very time-consuming. It could take days or months to perform a single experiment. The real road network has its perks like traffic and other road obstacles are taken into account and we get the most accurate results but it's not realistic to use it in our research.

Figure 4. Road distance and time retrieved by Google Maps between two locations.

Figure 4, the route is created from Joensuu to Kuopio using Google Maps navigation. It provides us the shortest route between two locations by taking traffic, road obstacles, accidents, construction, and speed limits into account. As we can see, there are two routes and Google Maps has already chosen the shortest route which will save us an extra 50 km of traveling. These results are always realistic, and most people rely on navigation maps nowadays when traveling.

2.3.4 Travel estimates

To run the optimization with real-time and accurate road network results, we propose to use an overhead graph [Mariescu-Istodor and Fränti, 2021] to estimate the travel distance and travel time between two locations.

The optimization is based on clustering and uses an overhead graph to obtain fast and accurate navigation calculations. The overhead graph always returns the fastest route from point A to point B. The Distance Matrix API is used for calculating the distance and time. Overhead graph size depends on the nodes, for this research a graph of size 512 nodes is used. The overhead graph first creates the bird distance from point A to point B and then it is scaled to the overhead of the nearest nodes resulting in the true road network path. Node sizes vary from 2 to 1024.

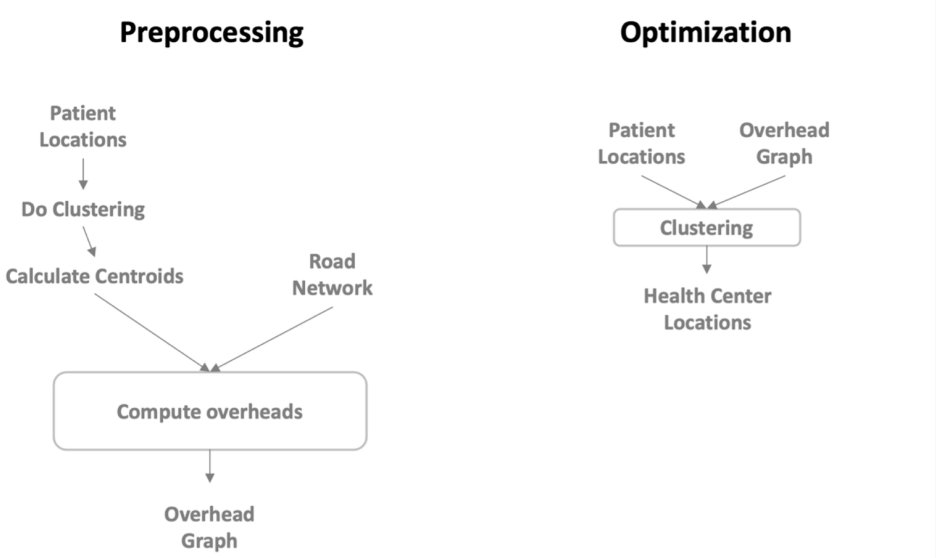

Figure 5. Diagram of the proposed method.

Figure 5, the pre-processing step produces the overhead graph needed for the actual optimization. The Euclidean distance is used when performing clustering usually, however, this is not an ideal choice when moving in the real world. This happens for many reasons such as road curvature, different speed limits, and topography. The overhead graph uses the distance matrix API and computes all pairwise overheads. The travel distance and travel time are then calculated. The results are faster and much reliable except the road traffic is not taken into the account with this overhead graph approach.

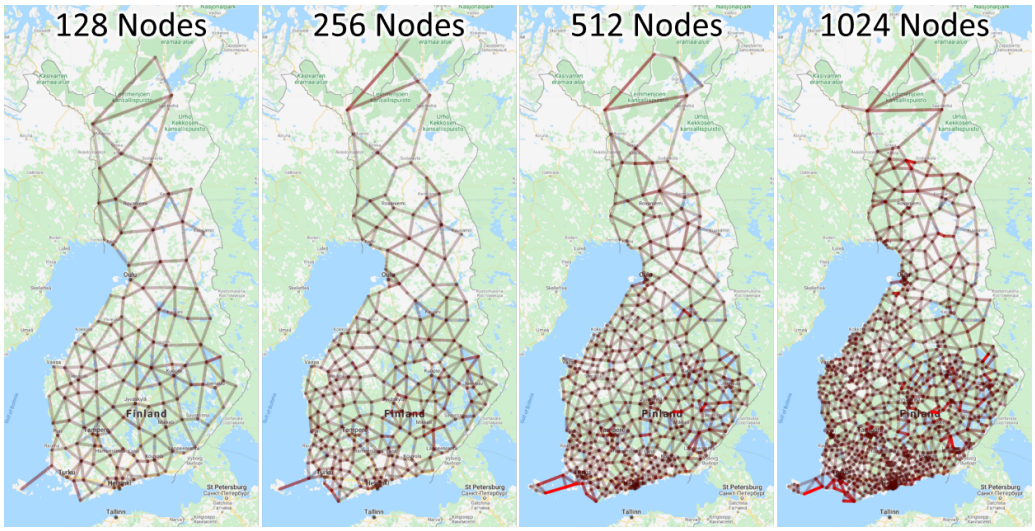

Figure 6. Four overhead graphs with varying node sizes.

Figure 6 shows four example overhead graphs with varying sizes computed based on the patient locations (original dataset) in Finland. In this way, we can estimate travel times in constant O (1) time. We proceeded with the 512 nodes graph size for our research.

Travel distance and travel time when using an overhead graph of node size 512 are beneficial because the distance and time values are very accurate and similar to real road network values. Comparing our optimization results in the end by using real road network values was done for double verification that the algorithm is implemented correctly and does not contain any kind of bugs.

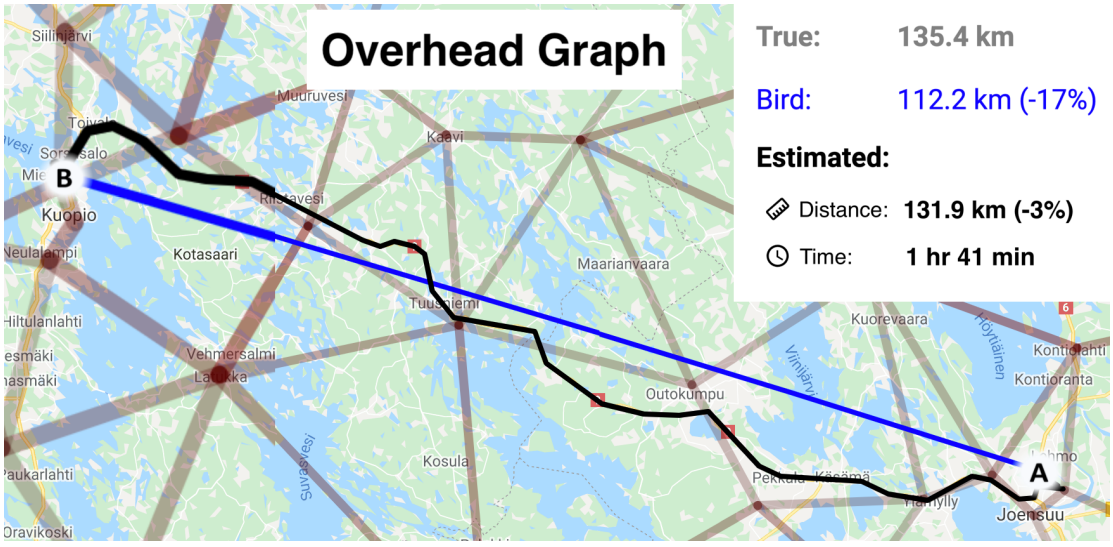

Figure 7. Travel distance and time estimated using the overhead graph (road path given for reference)

In figure 7, the route is generated using an overhead graph from Joensuu to Kuopio. We can see that the results are much similar and contain a very small error rate. The black route is retrieved by an overhead graph that contains navigation instructions. The blue route is the bird distance from Joensuu to Kuopio.

2.4 Initialization

Random swap with K-means requires initial centroids to perform clustering. The initial centroids are used to calculate the TSE and then to perform a random swap on the centroids and keep checking if a lesser TSE value is found.

K-means can be improved significantly in two ways. First, providing a good initialization technique can help in finding a better set of centroids. Secondly, repeating (restarting) the algorithm can also help in improving K-means. Following these two ways required to study the dataset in detail first. These two tricks were significantly helpful when cluster overlapping is found [Sieranoja and Fränti, 2019].

K-means is run on random initialization and is considered powerful enough to tune the locations. When there are many numbers of clusters and the data is separated widely fine-tuning the centroid locations is hard for K-means. Another important thing that we describe in K-means is the repeats. If on every single random swap, K-means is running 5 times, we have 5 chances to get a better set of centroids. Using a greater number of repeats is costly in terms of time complexity. The algorithm will run for a longer duration, but the set of centroid locations are always meaningful.

2.4.1 Random

A fixed random seed is used to generate a set of initial centroids randomly. The benefit of using a fixed random seed is that the locations are randomized in a predictable pattern. The drawback of using random locations as centroid is that some centroids may form over a lake or somewhere in the forest where no proper road network is established. Also, it is very unrealistic to create a hospital there in the future. Random locations are random and expecting a good set of centroids every time is not promised. By chance or luck, we may have a good start of centroids that lowers the risk of patients and TSE, but it all depends on the luck.

2.4.2 Current health care configuration

The STEMI dataset also has the original healthcare location i.e., the real hospitals present in the WGS84 coordinate system. Those locations are used as initial centroids because as we know the current situation of patients such as travel distance, travel time, and risk percentage. The objective of clustering optimization is to search a better location for a hospital so that a maximum number of patients are in reach within 90-minute of time.

Experiments were conducted with both random and original healthcare locations to see how they differ from each other. The original health care locations helped to initialize the centroids in such a way that we already have a good solution of hospital i.e., the original healthcare locations. Our algorithm has a good starting technique in this way and looks for another better set of centroids that lowers the risk of patients as well as the TSE.

2.5 Nearest node lookup

The overhead graph requires the information of the nearest node of locations passed to it. The algorithm iterates over all the patient locations and checks for the nearest node. The previous implementation of finding the nearest node for each location is in linear time. The nearest neighbor (node) trick helps to convert the linear time program to constant time.

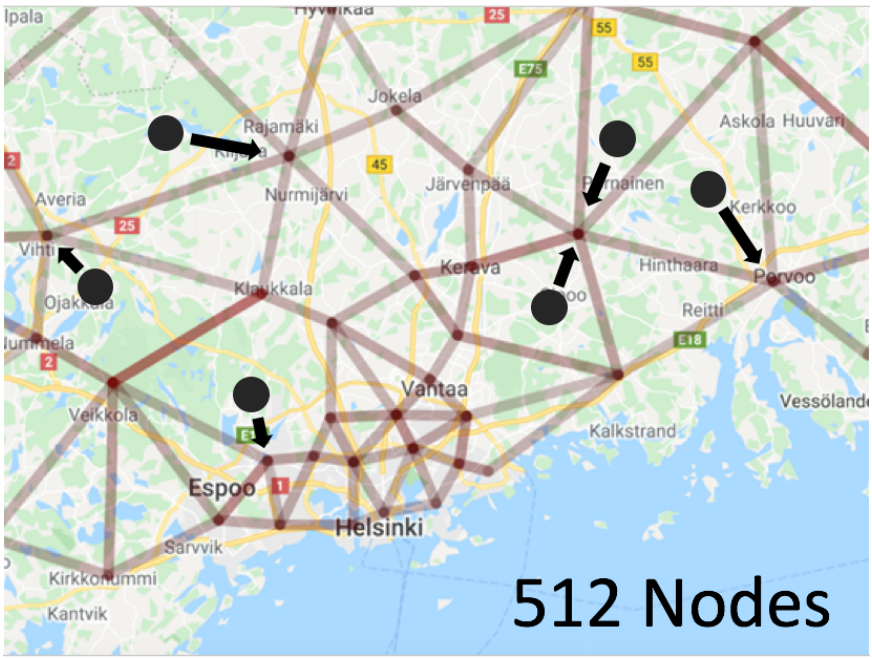

Figure 8. Locations mapped to the nearest Overhead graph node.

Figure 8 explains how patient locations are mapped to the nearest overhead graph node. These nodes then help in calculating travel distance and travel time.

The pseudo-code for implementation of nearest node lookup is:

```
Global lookup

getNearestNode(p, G):

   if lookup contains p then

      return value

   else

      find nearest node in G ← O(N)

      add it to lookup

      return value

   end if
```

Once for every patient

17 346 × 512

Once for every centroid

22 × 512

(every time it moves)

The Nearest neighbor (node) trick is implemented in two ways:

2.5.1 Lookup table for patient locations

The idea of generating a static lookup table is to store all the original geolocations in a hash code format which helps in finding the nearest node (index) in the overhead graph. It contains all the 17,346 patient locations in a hash code format.

2.5.2 Lookup table for health center locations

The dynamic lookup table contains the geolocations in hash code format for all the centroids or medoids which are the health center in reality. The searching for the nearest node (index) for each centroid geolocation becomes fast because of having a constant time complexity. The dynamic lookup table also gets emptied after every k-means iteration.

The nearest neighbor (node) trick with two lookup tables mentioned above is only helpful when running the random swap with K-means. Both lookup tables are used in the K-means making the time complexity to a minimum and making the algorithm more efficient. In the K-medoids case, the static lookup table is useful for storing all the geolocation as hash codes and returning them when estimating the distance from one point to another. All the centroids are the original data points having the smallest

distance to every other in a cluster, so the dynamic lookup table is never used. The centroids hash codes are already present in the static lookup table shifting the idea of using two lookup tables to only one.

2.6 Partitioning and representatives

Partitioning is a very core step in k-means in which all points are labeled to their closest cluster based on distance. The partitions are used when updating the centroids or medoids for a new better combination. They are also used when finding the TSE for a given set of centroids or medoids.

Partitioning requires a distance method to calculate the shortest distance between the relevant point and centroid and then assign points accordingly. Partitioning totally changes the points inside it if the distance method is changed. For example, if Euclidean or Bird distance is used to calculate the shortest distance between two points, then the points in partitioning are assigned by calculating and comparing the distances to every centroid and selecting the shortest distance available and assigning that point to the centroid. Now, what happens if there is a lake between the point and centroid? Patients are unable to cross the lake in real world. We have to rely on real road network results. Here comes the importance of choosing an overhead graph over other distance functions because it's possible to incorporate a real road network in the partitioning step. The points across the lake are assigned to one centroid and others are assigned to the second centroid resulting in no route diversion and meaningful partitioning.

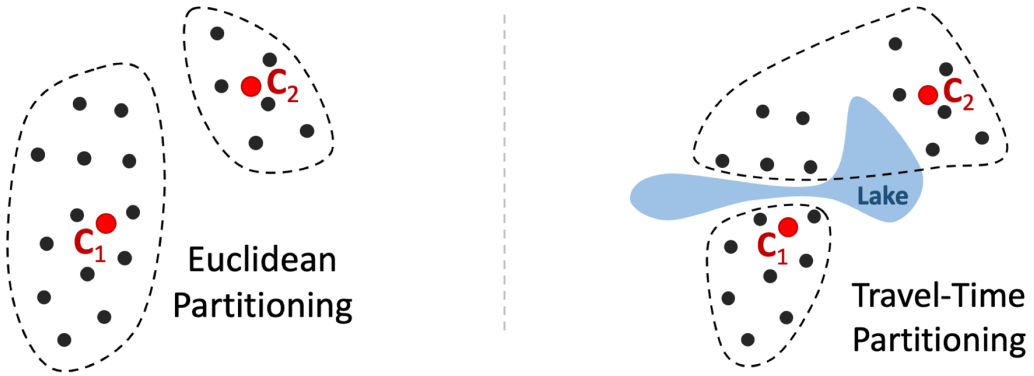

Figure 9. Partitioning behavior when performed with Euclidean and travel time.

Figure 9 explains that the Euclidean distance does not take responsibility for road obstacles and hurdles, such as lakes. On the other hand, using travel time is the right option to use when incorporating a real-road network is required.

The pseudo code for the partitioning is mentioned below:

K-Means (**P**, **K**) → (**C**, **L**)

Input : points **P** = {p_1,...,p_N}

the number of clusters **K**

Output: cluster centers **C** = {c_1,...,c_K}

cluster labels **L** = {label(i), i=1,...,N}

Randomly choose **K** initial centers **C** = {c_1,...,c_K}

REPEAT

$\mathbf{C}_{previous}$ ← **C**

FOR all i ∈ [1, N] DO // Partitioning

label(i) ← arg min d(p_i, c_j)

FOR all j ∈ [1, k] DO // Centroid update

c_j ← Average of p_i, whose label(i) = j

UNTIL **C** = $\mathbf{C}_{previous}$

The centroid relocation step is also very important in the algorithm which decides either to update the given set of centroids or update the set of medoids with every random swap iteration. Random swap uses K-means as a component between swaps. Representatives get updated with every iteration and we observe if they changed their former position resulting in lesser TSE as compare to the original TSE at the very first iteration.

We have used two representatives of K-means in our research:

2.6.1 Centroids

Centroid is the middle point of any cluster. It is formed after taking the mean in the K-means algorithm. A drawback of using K-means is that centroids/representatives may form over a lake, which is unrealistic to create a hospital there. Another limitation is that it is impossible to use other distance methods in K-means.

2.6.2 Medoids

Medoid is a point taken from the dataset which has the smallest sum of distance as compared to all other points in the dataset. K-means is not the ideal for STEMI dataset because we need a lower number of people at-risk after the optimization. The definition of at-risk is linked with travel-time not with Euclidean distance so we switched to K-medoids. K-medoids give us the freedom to use different distance functions when performing K-means iterations.

2.6.3 K-medoids instead of K-means

Experiments show that using K-means is beneficial only when minimized distance is required. For minimizing travel time, we switched to K-medoids.

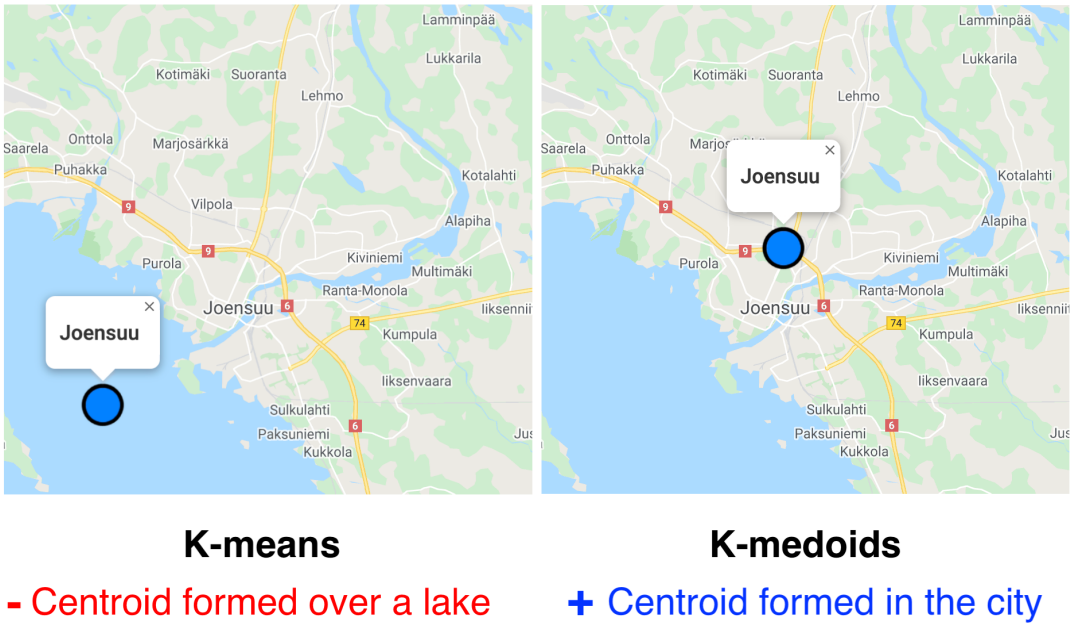

Figure 10. Health center forming over a lake with K-means and in the city with K-medoids.

Figure 10 explains that if we use K-means as a component function in the random swap, healthcare centers are sometimes relocated nearby a lake. This happens because K-means averages the location in the algorithm. On the other hand, K-medoid never creates a hospital or relocates a health center to the lake. The reason is that the medoid is one of the data points taken from the dataset. It is one of the actual patient locations. Patients live in the city or countryside areas so relocating a hospital to a lake is not possible when K-medoids is used.

The pseudo code for the K-medoids algorithm is as follow:

```
K-Mediods (P, K, Pi) → (M)

   Input : points P = {p1,...,pN}

           the number of clusters K

           partitioning Pi

   Output: medoids M = {m1,...,mK}

   FOR all m ∈ [1, K] DO

      minTotalTime ← 0

      FOR all i ∈ [1, N] DO

         totalTime ← 0

         FOR all j ∈ [1, N] DO  // Calculate time

              totalTime += EstimateTravelTime(points[i], points[j])

              IF totalTime < minTotalTime

              minTotalTime ← totalTime

              medoid[m] ← points[i]
```

The medoid is the location from the cluster with the smallest sum of travel times to all other locations. There are plenty of benefits of switching from K-means to K-medoids like clustering with K-means forms a centroid over a lake or forming it in the forest where no road network is present. K-medoids always form the cluster in one of the

original patient locations. The drawback of K-medoids is it's not possible to create a hospital at one of the patient locations.

2.7 Modeling risk

The idea of modeling risk is to make our algorithm understands patients who are at-risk by checking if they are 90-minute away from their nearest or any health facility. The algorithm does not know and is unable to differentiate patients that are 90-minute away from the nearest health center. The algorithm only has the travel time of the patients from their location to the health facility.

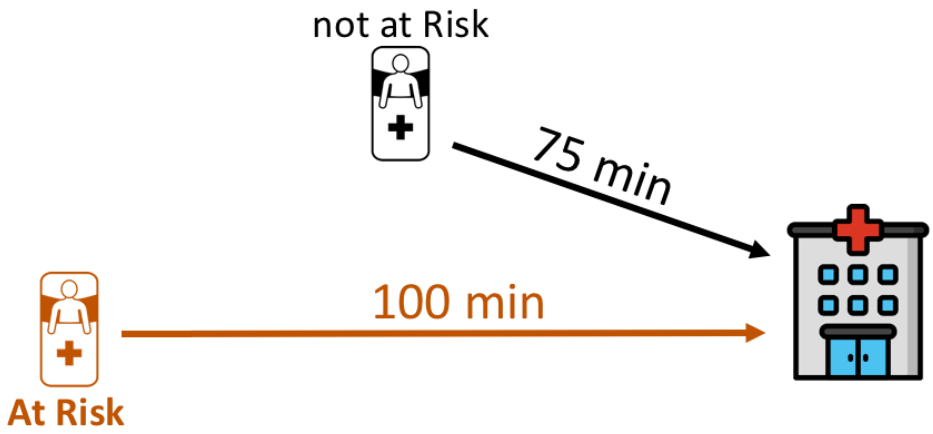

Figure 11. The definition of *at-risk*.

Figure 11 explains how the risk is measured. One patient is 75 min away from the hospital which is considered to be safe or in other words, not at risk. We can also say that in case of emergency this patient is on the safe patient's list, the second patient is 100 min away from the nearest hospital facility indicating that the patient is at risk. The definition of at-risk in our case study also explains that if a patient is 90 min away from the nearest or closest health facility, he is considered to be at risk.

This is the only way to differentiate between safe and at-risk patients. This has been accomplished by experimenting in two different ways:

2.7.1 Step function

Step function uses the conditional statement i.e., IF-ELSE. If the value is greater than 90, which is our risk threshold time the binary time becomes 0. If the value is less than or equals to 90, the binary time becomes 1. So, all the time values are either 0 or 1. This results in good results too but not that good as we have with sigmoid function.

Step function loses the proximity by just having values as 0 or 1. This is a floating-point precision problem and it's very unsure for the algorithm to decide and assign points to clusters. For example, a point may have the distance to all centroids equal to 1 because it's not in reach of any health facility and the algorithm does not know where to assign it.

2.7.2 Sigmoid function

The sigmoid function is implemented to binarize the travel times between 1 and 0. The values are separated on a threshold of 90-minute. The travel time values near to 90-minute are considered to be 0.5, travel time values after 90 are 1, and before 90 are 0. The sigmoid function is used for modeling the risk.

$$\sigma_{Risk}(x) = \frac{1}{1 + e^{-x+90}}$$

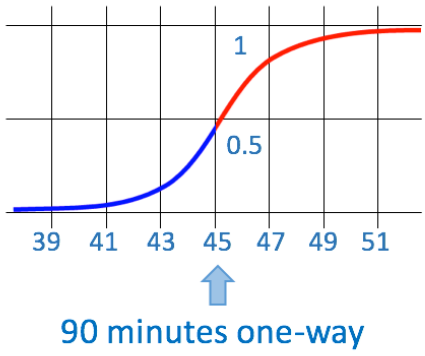

Figure 12. Sigmoid formula and graph for travel-time > 90 min.

Figure 12 shows the sigmoid formula with the visual representation on a graph for modeling the risk with travel time greater than 90-minute.

The implementation of sigmoid in the optimization algorithm looks like this as a pseudo-code:

```
Sigmoid Function (T) → (V)

   Input: time T in minute

   Output: value V in minute

   V = 1 / (1+Math.exp(-T+90))
```

The sigmoid function uses the Math Exponent class which is used in the sigmoid formula and performs processing over the estimated travel time. The result after using sigmoid on travel-time is between 0 and 1 but it contains the value with a lot more precision.

Distance Method	Avg. Bird Distance	Avg. Travel Distance	Avg. Travel Time	Patients At-Risk
Original HC	29.0 km	36.6 km	35.3 min	832 (5 %)
Binary Travel-Time	40.2 km	49.4 km	43.9 min	**138 (1 %)**
Sigmoid Travel-Time	41.3 km	50.6 km	45.0 min	**135 (1 %)**

Table 1. Comparison of sigmoid function with step function

Table 1 shows that, if we change the objective function from sigmoid to step, there is not a big difference in the patients at risk count. Sigmoid has proven to be the best suitable objective function for our research.

One important thing to note here is that we decided not to apply the sigmoid in the partitioning step of K-medoids. In theory, it does not have any effect on partitioning because sigmoid is an increasing function but in practice, it helped us in fixing floating point problems.

3 Experiments

Several experiments were conducted to search and study how clustering of STEMI dataset works when applying different techniques mentioned in the above chapter. All the experiments were performed in a pre-defined order over the same dataset and the same number of random swaps and k-medoids iterations.

1) Algorithm: Random Swap with K-medoids
2) Number of Random Swaps: 1000
3) Initial centroids: Random/Original
4) Distance methods: Bird distance, estimated travel distance, estimated travel time, and sigmoid estimated travel time

The optimization is performed by using 4 different distance methods to minimize the selected distance method when it is used as a distance method while clustering.

3.1 Quantitative

When minimizing total bird distance, the bird distance method is used as a distance method in clustering. Optimization with bird distance results in minimizing the total bird distance of patients to their closest health facility.

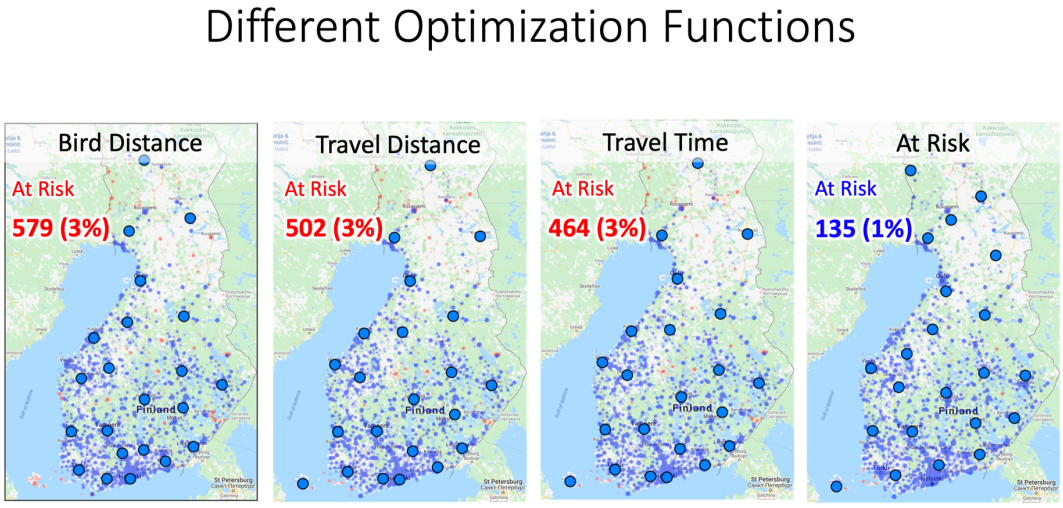

Figure 13. Optimization with different functions and the relocated centroids on the map.

Figure 13 shows healthcare centers on the map after relocation by running optimization with different distance functions. The risk constantly drops, and the best results are achieved by using sigmoid travel time. Risk which is showed by red color on the map can be seen in the first three map images in central, eastern, and northern Finland. Another important observation was that optimized locations were moved to more peripheral areas. The people who are still at risk are in Eastern Finland and Northern Finland. The risk dropped from 832 (5%) to 135(1%).

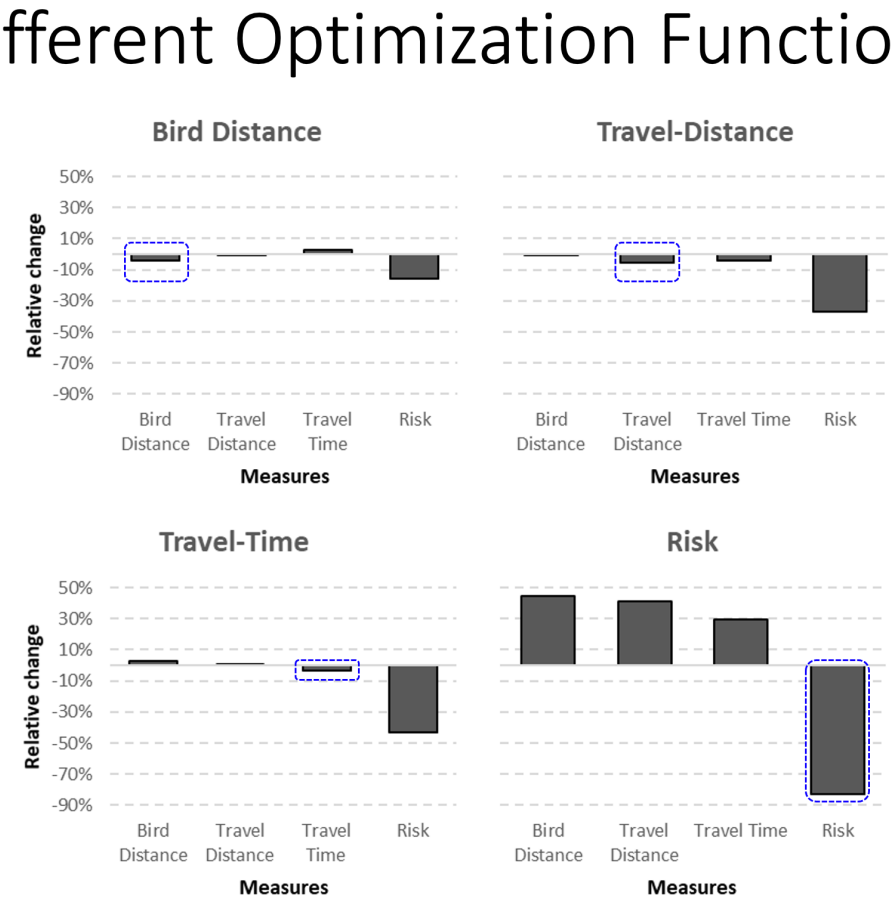

Figure 14. Optimization with different functions and their effect on distance functions.

Figure 14 shows the effect on patient risk relative change. The top-left graph shows optimization with Bird distance, the top right graph shows optimization with travel-distance, the bottom left graph shows optimization with travel time and the bottom right graph shows optimization with sigmoid travel time.

An important observation to be made here is no matter what we try to optimize, the risk always decreases. This indicates that patients at risk are linked to bird distance, travel distance, and travel time in some way. Minimizing any of them also shows the decrease in patients at risk but using the sigmoid estimated travel time is the most efficient and accurate way for our optimization. The bird distance, travel distance, and travel time are increased by a rate of 30% when optimized by using sigmoid travel time.

If we compare different distance functions that are used while performing optimization, we can see that risk always decreases. But the most important distance function which helped us in minimizing the patient risk count is the sigmoid travel time shown in the last graph. Using sigmoid estimated travel time is our end goal and proposed solution for our research. Sigmoid helps in decreasing the patients at risk which is also the risk modeling in our research.

3.2 Qualitative

The below statistics are retrieved from running the optimization over 1000 random swaps and using the same dataset. Random swap with K-medoids was the algorithm that was used here.

Distance method	**Avg. Bird distance**	**Avg. Travel distance**	**Avg. Travel dime**	**Patients at risk**
Original HC locations	29.0 km	36.6 km	35.3 min	**832 (5 %)**
Bird distance	28.2 km	36.9 km	36.0 min	579 (3 %)
Travel distance	28.4 km	34.9 km	33.9 min	502 (3 %)
Travel time	29.4 km	36.4 km	34.1 min	464 (3 %)
Sigmoid travel time	41.3 km	50.6 km	45.0 min	**135 (1 %)**

Table 2. Distance properties statistics when running with 1000 random swaps.

Table 2 shows the distance method that is used while doing clustering minimizes that distance property. If we optimize using bird distance, the average bird distance

decreases. If we optimize using travel distance, travel distance decreases. If we optimize using travel time, travel time decreases.

The smallest risk in the table came when we optimized using sigmoid estimated travel-time, dropping the risk by 4% as compared to the original(current) risk.

4 Optimization App

Centroids after successful optimization need to be displayed and analyzed on a map. We created a website tool for this to make things faster and much clearer for making results. The interface of this tool is designed in simple Front-end technologies i.e., HTML, JavaScript, and CSS. The Back-end is written in PHP. The algorithm itself is implemented in Java language.

4.1 Map application interface

Some images from the optimization app tool are explained below to use the tool in the best way possible and in making meaningful results and analyses.

4.1.1 Optimization legend UI

The web app has a UI for enabling disabling the markers drawn on Google maps. There are two kinds of markers that are health center locations and relocated optimized locations. There is another option to enable/disable clustering on the map.

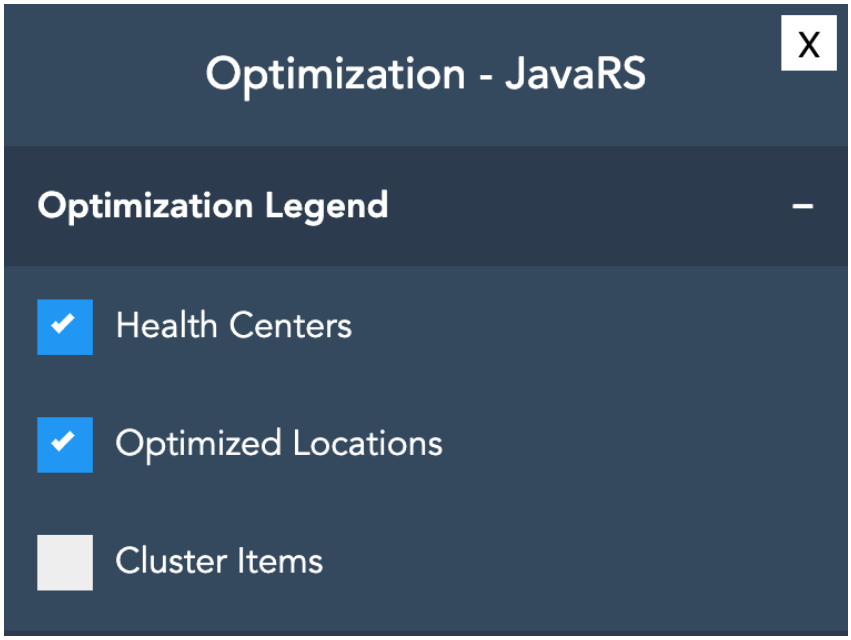

Figure 15. Optimization legend user interface.

Figure 15 shows the option to enable and disable different options on the map. The healthcare centers can be turned off on the map for better analysis of how centroids

are relocated. Optimized locations are centroids that are relocated after the optimization. The option for clustering them on the map is also available and can be turned on or off depending on our needs.

4.1.2 Optimization statistics UI

The web also has UI for displaying the statistic to analyze different factors when optimization is completed. Properties like Average bird distance, average travel distance, average travel time, and patients at risk details are displayed for both original health care locations and optimized locations.

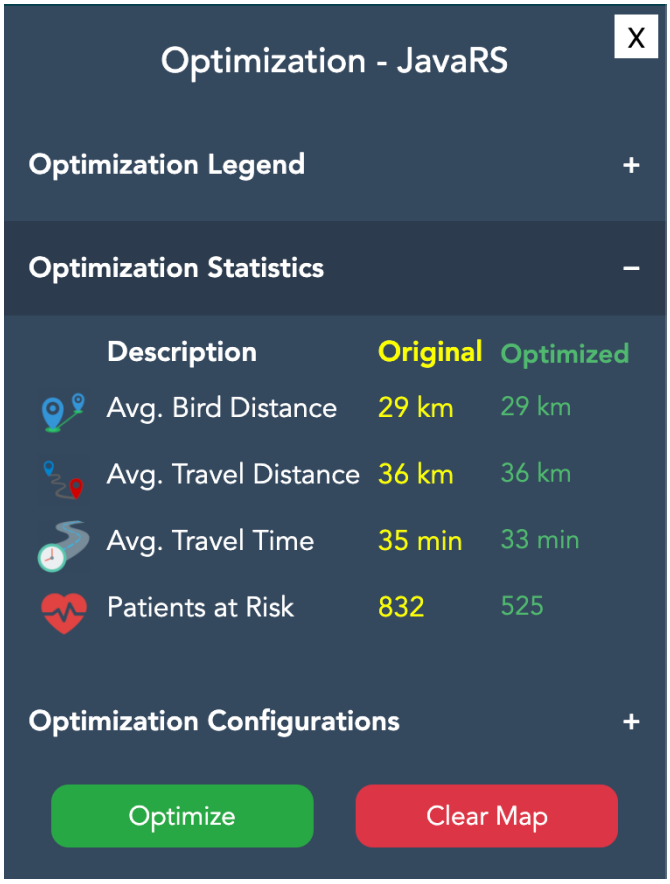

Figure 16. Optimization statistics user interface.

Figure 16 explains different distance measures in a table layout so that they can be compared easily. Important things to note while making observations like Bird distance, travel distance, travel time, and patients at risk count are compared before and after optimization.

4.1.3 Optimization configuration UI

Several configurations are required when running the optimization. The optimization algorithm should know which dataset to run for optimization, the algorithm and distance function should be selected, and initial centroids should be chosen. If travel time is set as a distance function, then the sigmoid dropdown appears otherwise it gets hidden on the user interface. Lastly, a total number of swaps are needed.

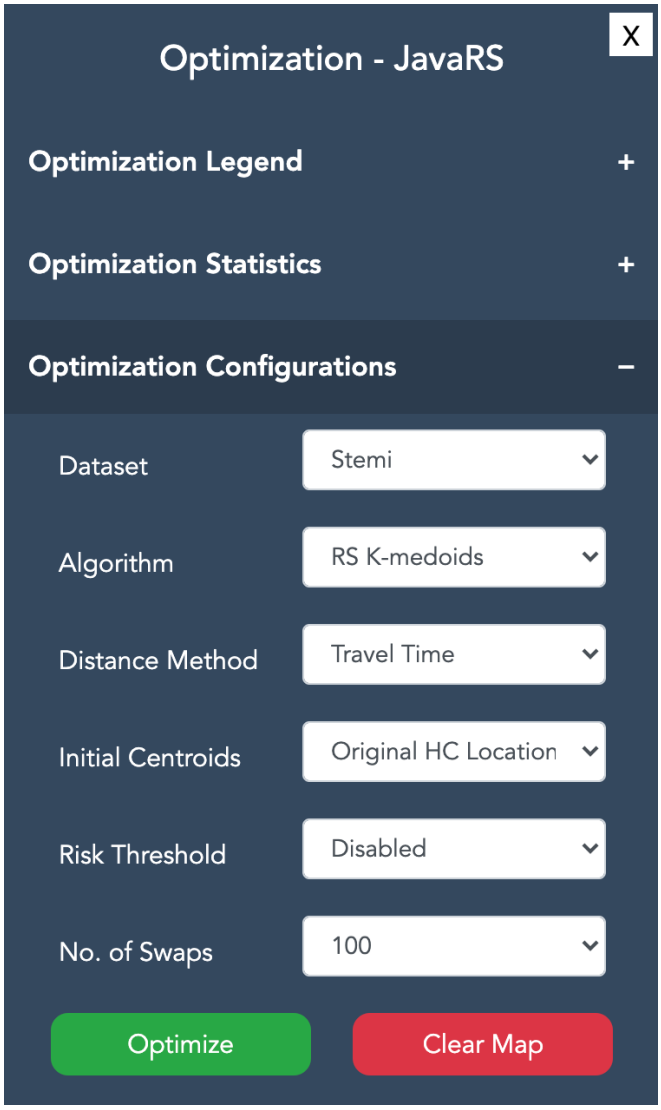

Figure 17. Optimization configuration user interface with buttons and dropdowns.

Figure 17 explains the configurations that are used for minimizing travel time for patients and the dataset is used as STEMI.

4.1.4 Running the optimization

To run the optimization, the configuration should be chosen by the user. There are two buttons available on the side panel. The green Optimize buttons run the optimization by sending the selected configuration from Front-end to the Back-end of the application. The Back-end runs the optimization algorithm and once the optimization is completed the Back-end notifies the Front-end and we can see relocated optimized locations on the map along with the original health center locations.

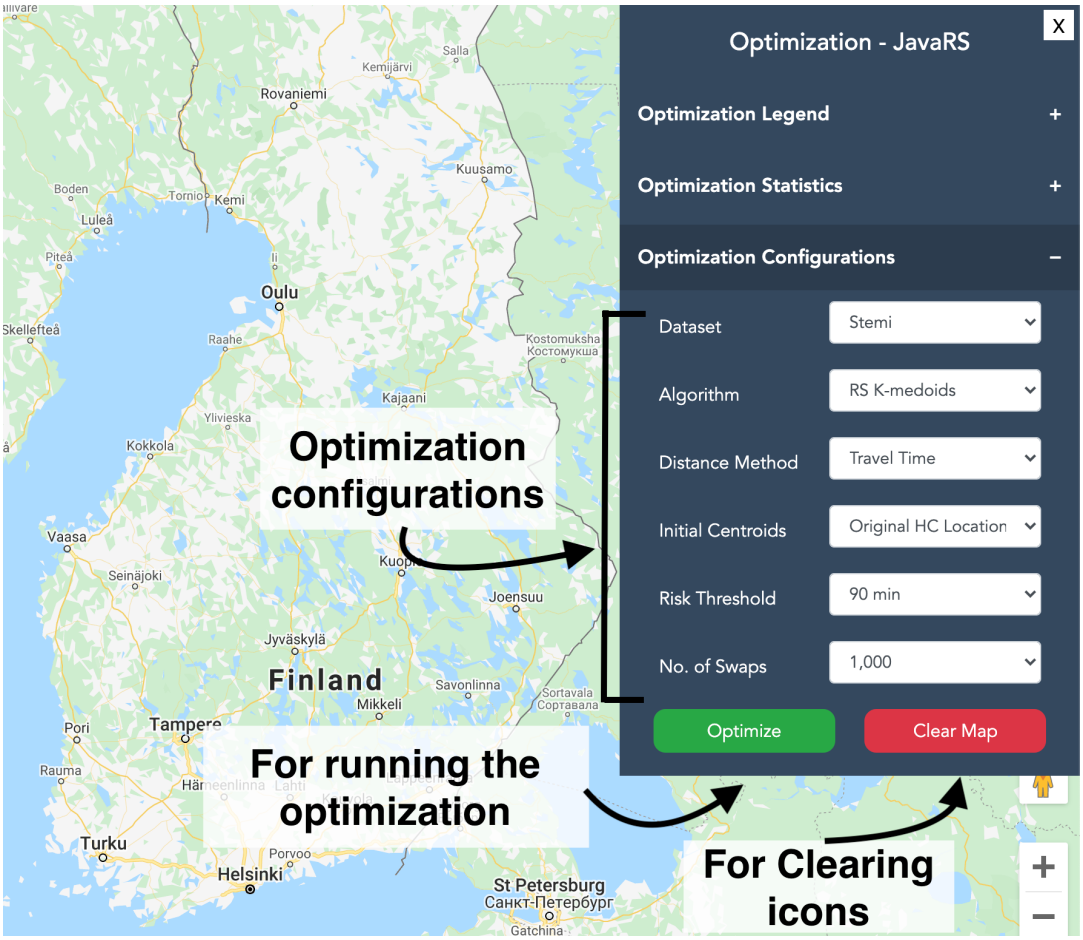

Figure 18. Original health centers on left and new optimized locations on right.

Figure 18 explains the visual screenshot taken from the optimization app. Pressing the green ‘Optimize’ button runs the optimization, whereas pressing the red ‘Clear’ Map button resets everything on the map and the user can again start from the beginning. Various number of available configurations are also shown, and they should be carefully chosen by the end-user to get accurate results.

4.1.5 Optimization results visualization

Once the optimization process is completed, multiple markers are drawn on a map. The markers with the H icon are the original location of health centers. The blue marker icons are the optimized locations that are relocated.

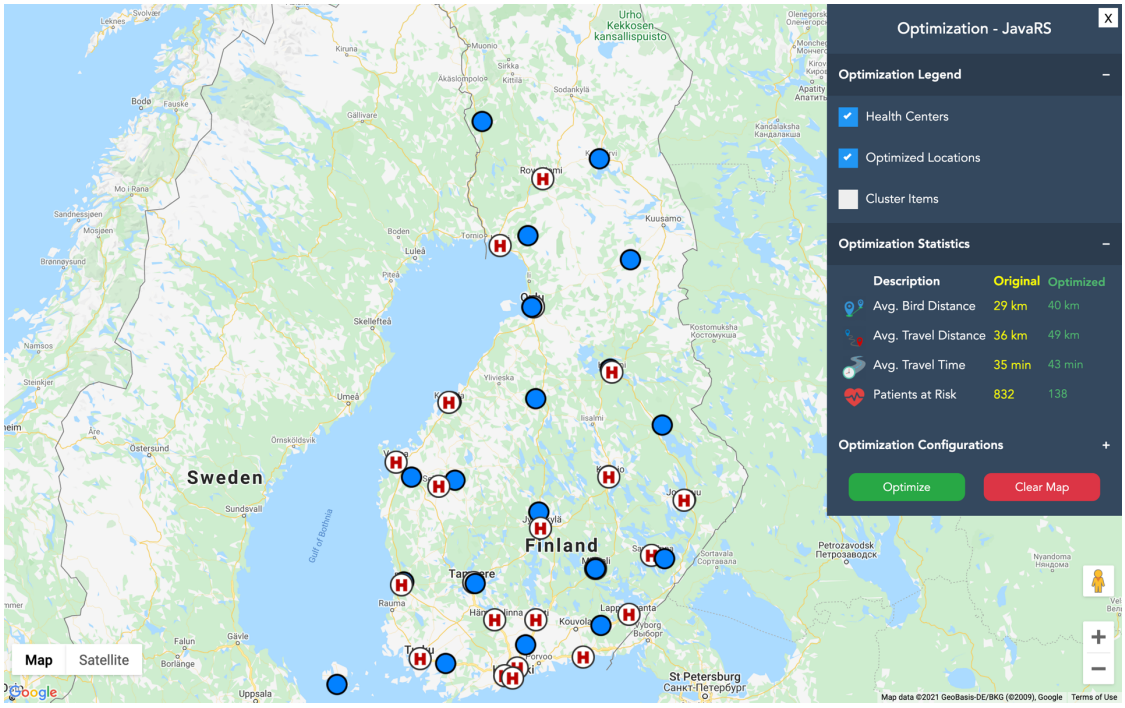

Figure 19. Marker icons are drawn on the map after optimization.

Figure 19 displays the final view once the optimization is completed. The markers are drawn on the map along with the original health center locations. Optimization statistics are also available for comparing it later by changing the optimization configuration and other things.

5 Conclusion

The research was based on trying multiple clustering techniques and observing what helps in achieving our objective and goals. The algorithm was implemented in such a way that the user input is very important for it to get the best results. We cannot let the computer decide everything. There are too many variables like removing some hospitals or shifting hospitals from a highly populated area and the cost of moving should also be kept in mind. There are several cases where computer decision is not that useful. The final decision is left to the decision-makers. We tried to create the system as flexible as possible for getting the best results.

Random Swap algorithm best suited our research goal plus performing random swaps and the search for the best random relocation position was not possible without it. K-medoid helped in giving us the option to engage different distance functions. For example, in the coming future, if the concept of helicopter ambulances is introduced, Euclidean or Bird distance would be the perfect distance function to use in that case. Euclidean and Bird distances are straight line distance or angled distance respectively, in other words, crow-fly distance.

The current need was to incorporate real real-time and accurate road network results in our research, which were successfully done by the overhead graph. Results were almost equal to the true road network with a little error rate of 0-2%. In terms of performance, it was faster than the true road network. It may take weeks/months to run the optimization on a large dataset but by using this overhead graph optimization is done in minutes/hours depends on the number of random swaps and k-means repeats.

Our definition of risk is reaching the hospital within 90-minute of time. The fast travel time estimator mentioned above does not model the risk. For risk modeling, we introduce the objective function in which sigmoid is used. The idea of modeling risk is to binarize the estimated travel times between 1 and 0. The values are separated on a threshold of 90 minutes. The travel times values near to 90 minutes are considered to be 0.5, travel time values after 90 minutes are 1, and before 90 minutes are 0.

References

Fränti, P. (2018). Efficiency of random swap clustering. Journal of big data, 5(1), 1-29.

Rezaei, M., & Franti, P. (2018). Real-time clustering of large geo-referenced data for visualizing on map. Advances in Electrical and Computer Engineering, 18(4), 63-74.

Banitaan, S., Nassif, A. B., & Azzeh, M. (2015, December). Class decomposition using k-means and hierarchical clustering. In 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA) (pp. 1263-1267). IEEE.

Omran, M. G., Engelbrecht, A. P., & Salman, A. (2007). An overview of clustering methods. Intelligent Data Analysis, 11(6), 583-605.

Steinbach, M., Ertöz, L., & Kumar, V. (2004). The challenges of clustering high dimensional data. In New directions in statistical physics (pp. 273-309). Springer, Berlin, Heidelberg.

Bezdek, J. C. (1981). Objective function clustering. In Pattern recognition with fuzzy objective function algorithms (pp. 43-93). Springer, Boston, MA.

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. science, 344(6191), 1492-1496.

Yiu, M. L., & Mamoulis, N. (2004, June). Clustering objects on a spatial network. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data (pp. 443-454).

ReVelle, C. S., & Eiselt, H. A. (2005). Location analysis: A synthesis and survey. European journal of operational research, 165(1), 1-19.

Teixeira, J. C., & Antunes, A. P. (2008). A hierarchical location model for public facility planning. European Journal of Operational Research, 185(1), 92-104.

Zhong, C., Miao, D., & Wang, R. (2010). A graph-theoretical clustering method based on two rounds of minimum spanning trees. Pattern Recognition, 43(3), 752-766.

Mestre, A. M., Oliveira, M. D., & Barbosa-Póvoa, A. P. (2015). Location–allocation approaches for hospital network planning under uncertainty. European Journal of Operational Research, 240(3), 791-806.

Zarrinpoor, N., Fallahnezhad, M. S., & Pishvaee, M. S. (2018). The design of a reliable and robust hierarchical health service network using an accelerated Benders decomposition algorithm. European Journal of Operational Research, 265(3), 1013-1032.

Malinen, M. I., Mariescu-Istodor, R., & Fränti, P. (2014). K-means∗: Clustering by gradual data transformation. Pattern Recognition, 47(10), 3376-3386.

Fränti, P., & Sieranoja, S. (2019). How much can k-means be improved by using better initialization and repeats?. Pattern Recognition, 93, 95-112.

Fränti, P., & Sieranoja, S. (2018). K-means properties on six clustering benchmark datasets. Applied Intelligence, 48(12), 4743-4759.

Mariescu-Istodor, R., & Fränti, P. (2021). Fast travel-distance estimation using overhead graph. Journal of Location Based Services, 1-19.

Fränti, P., Mariescu-Istodor, R., & Sengupta, L. (2017). O-Mopsi: Mobile orienteering game for sightseeing, exercising, and education. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 13(4), 1-25.