ALTERNATIVE CLUSTERING

ALGORITHMS IN SENSOR NETWORKS

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Divya Gupta

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

April 2010

Fargo, North Dakota

North Dakota State University
Graduate School

Title

ALTERNATIVE CLUSTERING

ALGORITHMS IN SENSOR NETWORKS

By

DIVYA GUPTA

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

ABSTRACT

Gupta, Divya, M.S., Department of Computer Science, College of Science and Mathematics, North Dakota State University, April 2010. Alternative Clustering Algorithms in Sensor Networks. Major Professor: Dr. Kendall E. Nygard.

A wireless sensor network is composed of a large number of tiny sensor nodes that can be deployed in a variety of environments like battle fields, water, large fields, and the like, and can transmit data to a Base station (BS). In a cluster-based network organization, sensor nodes are organized into clusters and one sensor node is selected as a sensor head (SH) in each cluster. Each SH denotes a facility and sends useful information to the Base Station (BS) through other SHs via the shortest path. In this paper, we study two clustering techniques, namely k-median clustering and k-center clustering for a wireless sensor network. All the sensor nodes are static and homogeneous (having the same specifications) and SHs are assumed to be heterogeneous with respect to other sensor nodes in their respective clusters (but homogeneous to other SHs once they are located). The focus of this paper is to compare the k-median and k-center clustering techniques based on shortest path and total intra-cluster distance. We have implemented the two clustering techniques using the Java language and necessary experimental and statistical results are provided.

Keywords: Wireless sensor network, Sensor node, Homogeneous, Heterogeneous, Clustering, K-median, K-center, Facility location problem, Sensor Head, Base station and Shortest path.

ACKNOWLEDGMENTS

I would like to express sincere thanks to my advisor Dr. Kendall Nygard for his continued support throughout this paper. I am grateful for the enormous ideas and suggestions given by him. Also I would like to express special thanks to my supervisory committee members, Dr. John Martin, Dr. Brian Slator and Dr. Brenda Hall for their valuable inputs which helped me immensely in completing this paper.

I would also like to express thanks and gratitude to my parents who kept on encouraging me to work hard, and also to my dear friends Mr. Pranav Dass, and Mr. Barjesh Arora who have supported and motivated me throughout in writing and completing this paper.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1. Overview

A sensor network is composed of a large number of sensor nodes that are densely deployed. These sensor nodes are small in size and can communicate within short distances. These tiny sensor nodes consist of sensing, data processing and communicating components [9]. Some of the application areas for these nodes are health, agriculture, military and within homes [10]. Sensor nodes are usually scattered in a sensor field, an area where they are deployed. The nodes coordinate among themselves to produce information about the physical environment and send that information to a remote base station. A greater number of sensors allows for sensing over larger geographical regions with greater accuracy.

Figure 1 illustrates a MICA2 wireless sensor node which is tiny in size, has little memory, and a short transmission range [31].

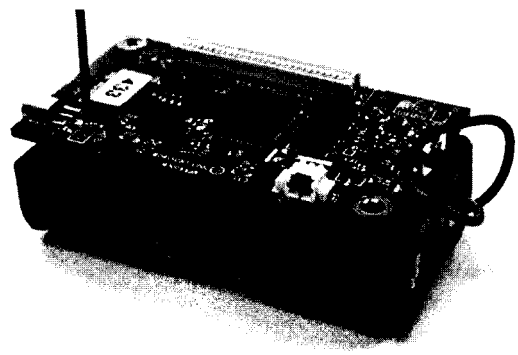

Figure 1. A Wireless Sensor Node "MICA2"

1.1.1. Design Assumptions

The sensor networks that we focus in this study have the following characteristics:

1. All the sensor nodes are deployed *at known locations* in a given region of interest.
2. The sensor nodes possess a *hierarchical* network structure (Cluster-Based Model).
3. All the sensors are *static*.
4. All sensors (Non-SH) *are homogenous* (have the same sensing range), and are deployed at *known locations* and Sensor heads (SHs) are assumed to be *heterogeneous* with respect to other sensor nodes in their respective clusters (but *homogeneous* to other SHs once they are formed).
5. The sensors in the networks do not fail.

1.1.2. Facility location problem

The facility location problem is to open a subset of facilities so as to minimize the sum of distances from each node to its closest open facility [18]. In this paper, we use the term *sensor heads (or SHs)* for "facilities" throughout.

1.1.2.1. Median problem

Here a pre-specified number of facilities must be located so as to *minimize the average distance* to or from the facilities from their affiliated sensor nodes. Median problems arise very often in the context of facility construction for delivery

of *non-emergency services* (post offices, transportation terminals, telephone interchanges, little town halls and, offices for government agencies dealing extensively with the public).

In this paper, we are considering a variant of the median problem, namely the *K-median problem*, in which the goal is to minimize the sum of distances of each node from its closest chosen centroid (or more specifically, sensor head) from a given set of *K* clusters.

1.1.2.2. Center problem

Here a pre-specified number of facilities must be located so as to *minimize the maximum distance* to or from the facilities for the number of their sensor nodes. Center problems (also sometimes referred to as mini-max problems) are more applicable in the context of *emergency urban services* (emergency medical care, fire fighting, and emergency repair services).

In this paper, we are considering a variant of the center problem, namely *K-center problem*, in which the goal is to minimize the maximum distance of each node from its closest chosen centroid (or more specifically, sensor head) from a given set of *K* clusters.

1.1.3. Clustering

Clustering is an important problem in computer science with applications in many problem domains. It is the problem of grouping a set of physical or abstract objects into classes of similar objects.

In a cluster-based network organization, sensor nodes are organized into clusters and one sensor node is selected as a sensor head (SH) in each cluster [8]. Sensor head has a more pronounced role than other sensors such as gathering information from other sensors and transferring it to the base station as compared to a normal sensor node which does not possess this particular feature.

In Figure 2, the cluster-based sensor model used in our paper is being described. We are given a two-dimensional sensor field in which all the sensor nodes are deployed at known locations. These sensor nodes are formed into clusters using either K-Median or K-Center clustering technique and one sensor node is selected as a sensor head (SH) for each cluster. Each SH is shown using a different color in the figure and the sensor nodes associated with each SH are also marked with the same color and shown through small line segments. Then data (information about sensor nodes and their respective SHs) is transmitted via SHs indicated by the BLUE arrows forming a MST (Minimal Spanning Tree) to the Base Station (BS) [8]. Then the data is collected by the base station and the solution is evaluated using Euclidean metric within the clusters and the minimum spanning tree routing among the SHs.

1.1.3.1. K-Median Clustering

In the k-median clustering problem, a set $P \sqsubseteq X$ is provided together with a parameter k. We would like to find k points $C \sqsubseteq P$, such that the sum of distances of points of P to their closest point in C is minimized.

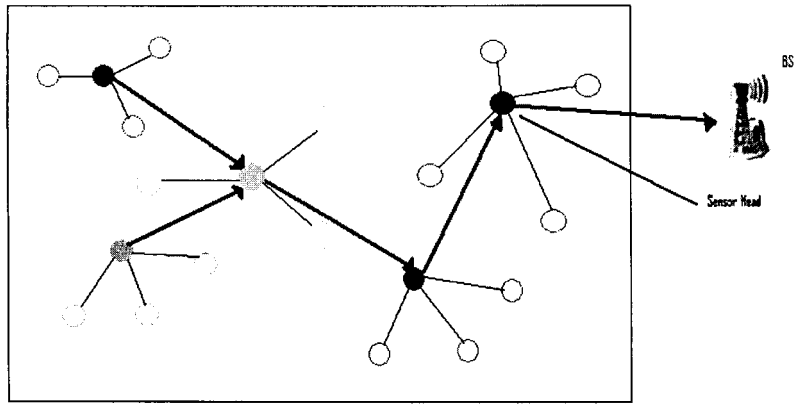

Figure 2. Cluster-Based Sensor Model

1.1.3.2. K-Center Clustering

In the k-center clustering problem, a set P ⊑ X of n points is provided together with a parameter k. We would like to find k points X ⊑ P, such that the maximum distance of a point in P to the closest point in X is minimized.

Now we present a very important result pertaining to our proposed problem known as Hakimi's Theorem [27].

1.1.4. Hakimi's Theorem

Hakimi's theorem states: "At least one set of k-medians exist solely on the nodes of G", where G is an undirected graph (or network) with *N* nodes.

In effect it reduces the burden of searching for an infinite number of points within a large connected network. We only have to search out of *N* nodes for locating *K* facilities because of Hakimi's Theorem [27]. This concept has also been used in this paper for implementing *k-median* problem and the same assumption of

searching out of only *N* nodes for locating *K* facilities is used for implementing *k-center* problem.

After completion of the clustering phase, we will transfer the data to the base station through the shortest route, which can be observed by considering a minimum spanning tree explained below.

1.1.5. Minimum Spanning Tree

A spanning tree of a graph G is a sub graph of G that is a tree containing all the vertices of G. In a weighted graph, a *minimum spanning tree* or *minimum weight spanning tree* is a spanning tree whose sum of edge weights is the smallest. It always provides an optimal solution because the sum of edge weights will always be the smallest [28].

Figure 3 shows an example of a minimum spanning tree where each edge is labeled with its weight. In this case the sum of edge weights forming the minimum spanning tree comes out to be 38 distance units.

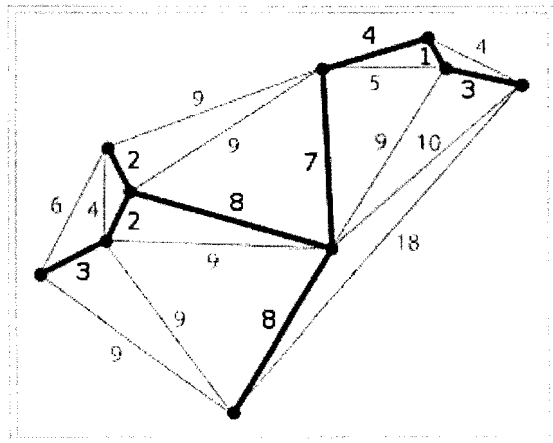

Figure 3. Minimum Spanning Tree

1.2. Outline

The rest of the paper is organized as follows. Section 2 describes the related work with respect to the Facility Location Problem. Section 3 contains the actual problem statement. Section 4 presents the solution approach to the Facility Location Problem. Section 5 shows the experimental results and the necessary analyses. Section 6 provides the conclusion and future work and Section 7 contains references.

2. LITERATURE REVIEW

The problem of clustering a given set of nodes (or points) has been widely studied by the researchers over the past few years. Many researchers have proposed various clustering schemes and the most popular ones happened out to be *K-Means*, *K-Median* and *K-Center* clustering. All of these schemes focus on selecting a cluster head for each cluster in which the given set of nodes are to be clustered.

Extensive research has focused on developing hierarchical routing (or cluster-based algorithms). Hierarchical routing refers to a two-layer routing mechanism where one layer is used to select a sensor head and the other layer is used for routing the information to the base station. Heinzelman et. al. proposed a hierarchical clustering algorithm, called Low-Energy Adaptive Clustering Hierarchy (LEACH), in which random rotation of cluster heads in a cluster-based protocol evenly distributes the work load among the sensors [1]. Bandyopadhyay and Coyle focused on minimizing the communication cost in hierarchically clustered networks, which was, for some time, considered a simple strategy to select random cluster heads with a specified probability [2, 3]. Other popular cluster-based protocols were Linear distance-based scheduling (LDS) and Balanced-energy sleep scheduling (BS), developed by Deng et al for cluster-based high density sensor networks. In LDS the goal is to reduce energy consumption while maintaining adequate sensing coverage capabilities. BS extends the LDS scheme by evenly distributing the sensing and communication tasks among the non-head sensors so that their energy consumption is similar regardless of their distance to

the cluster head [4, 5, 18]. However, in our paper, we are not considering any of these algorithms. We are simply interested in clustering of nodes and routing any desired information via only the cluster (or sensor) heads thus formed to the base station.

Some of the clustering algorithms involving facility location problems focus on organizing sensor nodes into clusters and selecting one sensor out of all sensor nodes as a sensor head in each cluster [8, 19]. This concept has been thoroughly studied in this paper. There are two clustering techniques to focus upon in this paper, namely *K-Median* and *K-Center*, and a thorough comparison of these two techniques has been done in this paper based upon the "shortest route to the base station" and the "Total intra-cluster distance" among the nodes.

Over many years researchers have developed several algorithms to account for *K-Median* clustering. Some of them are being discussed now. A dynamic programming algorithm to implement *K-Median* clustering using a tree metric was proposed by Tamir [11]. It was improved further by Bartal [12] in 1998. The first arbitrary precision approximation for planar k-median and first polynomial time approximation algorithm with a provable performance guarantee was developed by Lin and Vitter [13, 14]. Korupolu provided an analysis of k-median problem solved by a local search heuristic [15]. Charikar et al proposed the first constant factor approximation algorithm for k-median clustering in the metric space settings [6, 7].

Some of the concepts and implementation methods to account for *K-Center* clustering will be discussed now. A 2-approximation algorithm has been proposed

by Hochbaum and Shmoys [23]. Some 2-approximation algorithms have also been proposed by Kleinberg and Tardos [21], and Kanungo et al [22]. In 2006, farthest-point heuristic approximation for the k-center problem proposed by He [24], which also closely related to our approach for solving both the *K-Median* and the *K-Center* problems.

A heuristic always gives a good approximation to the problems which are *NP-Hard* [21] in nature, as is the case with our proposed problem. Thus to solve such problems for an optimal solution is a very complex process. Though we have studied some integer-linear programming techniques for solving such problems [7, 12, 14, 16, 22, 29] but because of their extreme complexities we decided to solve the proposed problem using a heuristic [24].

To transfer data to the base station we have implemented the concept of Prim's Minimum Spanning Tree algorithm using an adjacency matrix representation [30].

For comparing both the techniques we have considered one more aspect of clustering, namely Intra-cluster distance, which can be taken on an average and it interprets the "closeness" of the facilities to the clients in a particular area [20].

The work closest to our research is being done by Charikar et al [16, 17] in which a few very distant clients, called outliers, can exert a disproportionately strong influence over the final solution but to resolve the problem of very distant clients/sensors, we are using the concept of clustering (k-median and k-center) techniques.

Our proposed problem is "unique" in the sense that a comparison between *K-Median* and *K-Center* clustering techniques has been performed based on some parameters. This problem finds several applications in the field of facility location, where the goal is to locate a set of facilities for a given set of client locations such that the sum of distances from any of the client location to its closest facility is minimized [8, 19]. Another application area for our problem is "feature selection", where the objective function of the clustering algorithm is constantly perturbed resulting in the gradual shift of each of the clusters towards a global median of zero for the entire unlabeled dataset [25]. One more application of this technique lies in the fact that clustering a very large volume of data is cumbersome and filled with errors. So, in the data stream model of computation, the points are read in a sequence, and this technique is useful in segregating a set of points (cluster of points) to facilitate computation of the objective function [26].

3. PROBLEM STATEMENT

The formal definition of the proposed problem is given as:

Problem Definition: *Given a set of S sensors (clients) in a 2-D region, cluster them into K subsets, each containing a sensor head SH (facility) such that the data communication takes place via the shortest route through those K sensor heads. The sensor heads form a backbone for the network and transmit all the data to a single base station. The location of these K facilities is done through both K-median and K-center clustering techniques and a performance comparison is done between the two based upon the "Total intra-cluster distance" and the "shortest route" to the base station.*

The term *Total intra-cluster distance* refers to the total Euclidean distance between all the nodes and their respective sensor heads (SHs) for all clusters. The term *shortest route* refers to the total minimum Euclidean distance travelled through all the sensor heads (SHs) visiting each of them exactly once achieved by forming a minimum spanning tree (MST) among the SHs.

Given:

N: set of sensor nodes

K: number of clusters

Mathematically the *K-median* problem can be formulated as:

Find a set *C* of *K* points that minimizes

$D(N) = (\sum_{n \in N} d(n, C))$, where

$d(n, C) = \min_{c \in C} (d(n, c))$ for every $n \in N$

This indicates that the sum of the distances of every node $n \in N$ to its closest SH is minimized.

Mathematically the *K-center* problem can be formulated as:

Find a set C of K points that minimizes

$D(N) = (\max_{N} (d(n, C)))$ for every $n \in N$, where

$d(n, C) = \min_{c \varepsilon C} (d(n, c))$ for every $n \in N$

This indicates that the maximum distance of every node $n \in N$ to its closest SH is minimized.

The term $d(n, C) = \min_{c \varepsilon C} (d(n, c))$ for every $c, n \in N$ indicates the calculation of "distance of every node to its closest centroid, or SH" for all such SHs.

4. SOLUTION APPROACH

We present two detailed algorithms for the Facility Location Problem for finding the cluster set of size *K* using *K-median* and *K-center* clustering techniques. The values representing the numbers in distance matrix for both the techniques are taken by rounding them to the nearest floor function, for example, floor (4.2) = 4.

4.1. K-Median Clustering

We are given a set of *N* sensor nodes deployed at known locations and *K* clusters. There are only *N* candidate points on the network for the placement of the SHs because of the Hakimi's theorem discussed earlier. Figure 4 illustrates a heuristic algorithm for *K-Median* clustering technique.

The complexity of the *K-Median* clustering algorithm is $O(|N|^2 \cdot K)$, where *N* represents the set of initial deployed sensor nodes and *K* represents the number of clusters.

Now we present a step-by-step example which describes the algorithm.

Objective: Cluster the following set of 10 sensor nodes into 3 clusters, i.e. K = 3.

Consider a data set of 10 sensor nodes as follows:

$S_1 = (2, 4)$ $S_6 = (5, 7)$

$S_2 = (3, 6)$ $S_7 = (6, 9)$

$S_3 = (4, 8)$ $S_8 = (4, 3)$

$S_4 = (3, 9)$ $S_9 = (3, 3)$

K-Median Clustering

1. Input: S = Set of sensor nodes deployed at known locations, $S_i = \{(a, b) \mid a, b \in Z^+\}$ and $S = \{\{S_i\} \mid i \in Z^+\}$; K = Number of Clusters.

2. Initialization: C = set of centroids, $C = \{\{C_i\} \mid C_i \sqsubset S\}$; $Cluster_j$ = cluster set j containing the centroid C_i and the node (s) associated with that centroid, $Cluster_j = \{\{C_i, S_i\} \mid C_i \in C, S_i \subset S\}$; Sum_Dist = 0. $1 \leq i, j \leq K$.

3. Initial Step: (a) Randomly select K centroids from S;

(b) Form set C of K chosen centroids;

(c) Calculate the distance of each node $S_i \in (S - C)$ from each centroid $C_i \in C$ and store all these distances in a matrix;

(d) Compare each of those distances and select the C_i having the lowest distance with S_i.

(e) Form K cluster sets $Cluster_j$'s ($1 \leq j \leq K$) by associating each S_i to its closest C_i such that each cluster set contains the centroid along with all its associated nodes;

(f) Calculate the sum of distance of each S_i to its closest C_i and store it in Sum_Dist.

4. Iteration Step: (a) For each centroid $C_j \in C$

(i) For each node $S_i \in (S - C)$ /*Chosen randomly*/

1. Swap C_i & S_i and repeat steps 3(b) to 3(f).

(ii) End For

(b) End For

5. Select the set C having the lowest sum ofdistance.

6. If C changes then repeat step 4 else go to step 7.

7. Output: C = Set of K centroids;

$Cluster_j$'s = Cluster sets containing sensor nodes associated to their closest centroids. $1 \leq j \leq K$

Figure 4. K-Median Clustering Algorithm

$S_5 = (8, 5)$ $S_{10} = (9, 4)$

Figure 5 shows sensor nodes for calculating K-Median clustering.

Step 1: Initialize K Centroids

Centroids $C = \{C_1, C_2, C_3\} = \{(4, 8), (6, 9), (5, 7)\}$

Let us assume $C_1 = (4, 8)$, $C_2 = (6, 9)$ & $C_3 = (5, 7)$

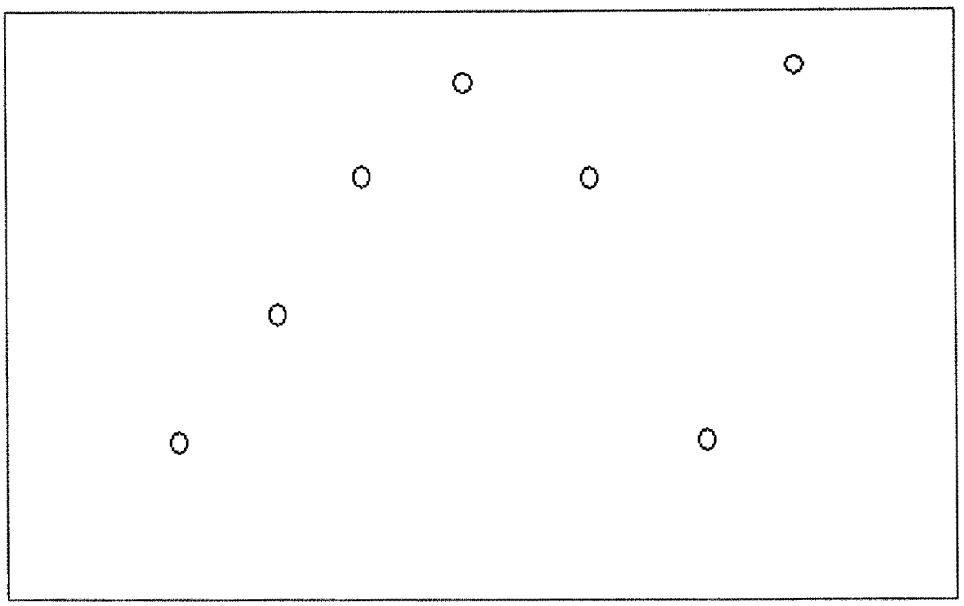

Figure 5. All Sensor Nodes for K-Median Example.

Calculate distance matrix for each centroid using Euclidean distance. Table 1 illustrates the calculated distance matrix below.

Now proceeding row-wise and comparing the distance values in each row we deduce which node should belong to which cluster. So we start with the first row and compare distance values of a node with each centroid, i.e., compare dist ((4, 8), (2, 4)), dist ((6, 9), (2, 4)) and dist ((5, 7), (2, 4)), and find the lowest value among these distances. In this case, it comes out to be 4 (in case of a tie, choose any one). We have chosen (4, 8) as the centroid for (2, 4).

Likewise we compare the distance values for each row and evaluate the minimum distance and on the basis of that assign each node to its closest centroid.

Then Clusters become:

$Cluster_1$ = {(4, 8), (2, 4), (3, 6), (3, 9)}

Table 1. Distance Matrix for Step 1 of K-Median Example.

	Sensor Nodes			Sensor Nodes			Sensor Nodes	
C_1	S_i	Distance	C_2	S_i	Distance	C_3	S_i	Distance
4,8	2,4	4	6,9	2,4	6	5,7	2,4	4
4,8	3,6	2	6,9	3,6	4	5,7	3,6	2
4,8	3,9	1	6,9	3,9	3	5,7	3,9	2
4,8	8,5	5	6,9	8,5	4	5,7	8,5	3
4,8	4,3	5	6,9	4,3	6	5,7	4,3	4
4,8	3,3	5	6,9	3,3	6	5,7	3,3	4
4,8	9,4	6	6,9	9,4	5	5,7	9,4	5

$Cluster_2$ = {(6, 9), (9, 4)}

$Cluster_3$ = {(5, 7), (8, 5), (4, 3), (3, 3)}

Since the nodes that are close to their respective centroids form a cluster

Sum of Distances = {dist ((4, 8), (2, 4)) + dist ((4, 8), (3, 6)) + dist ((4, 8), (3, 9))}

+ {dist (6, 9), (9, 4)}

+ {dist ((5, 7), (8, 5)) + dist ((5, 7), (4,3)) + dist ((5, 7),(3, 3))}

= (4 + 2 + 1) + 5 + (3 + 4 + 4)

= 7 + 5 + 11 = **23**

Step 2: Select a non-centroid C_1' (another sensor node which is not present in the initial set of centroids) and replace it with C_1.

Let us assume C_1' = (3, 3)

Now C = {(3, 3), (6, 9), (5, 7)}

Calculate distance matrix for each centroid again using Euclidean distance. Table 2 illustrates the calculated distance matrix below.

Table 2. Distance Matrix for Step 2 of K-Median Example.

	Sensor Nodes			Sensor Nodes			Sensor Nodes	
C1	**Si**	**Distance**	**C2**	**Si**	**Distance**	**C3**	**Si**	**Distance**
3,3	2,4	1	6,9	2,4	6	5,7	2,4	4
3,3	3,6	3	6,9	3,6	4	5,7	3,6	2
3,3	4,8	5	6,9	4,8	2	5,7	4,8	1
3,3	3,9	6	6,9	3,9	3	5,7	3,9	2
3,3	8,5	5	6,9	8,5	4	5,7	8,5	3
3,3	4,3	1	6,9	4,3	6	5,7	4,3	4
3,3	9,4	6	6,9	9,4	5	5,7	9,4	5

Now new clusters become

$Cluster_1$ = {(3, 3), (2, 4), (4, 3)}

$Cluster_2$ = {(6, 9), (9, 4)}

$Cluster_3$ = {(5, 7), (3, 6), (4, 8), (3, 9), (8, 5)}

Sum of Distances = {dist ((3, 3), (2, 4)) + dist (3, 3), (4, 3))}

+ {dist ((6, 9), (9, 4))}

+ {dist ((5, 7), (3, 6)) + dist ((5, 7), (4, 8) + dist ((5, 7), (3, 9)) + dist ((5, 7), (8, 5))}

= (1 + 1) + 5 + (2 + 1 + 2 + 3)

= **15**

Now 15 < 23, so replacing (4, 8) with (3, 3) was a better choice and likewise we continue to swap every centroid with every non-centroid one by one (for all centroids till we get the lowest distance). In case of a tie with the lowest distance select any node out of the two choices.

4.2. K-Center Clustering

We are given a set of *N* sensor nodes deployed at known locations and *K* clusters. There are only *N* candidate points on the network for the placement of the SHs because of the same assumption as used in the implementation of *K-Median* technique discussed earlier. Figure 6 illustrates a heuristic algorithm for a *K-Center* clustering technique.

The complexity of the *K-Center* clustering algorithm is $O(|N| \cdot K)$, where *N* represents the set of initial deployed sensor nodes and *K* represents the number of clusters.

Now we present a step-by-step example which describes the algorithm.

Objective: Cluster the following set of 10 sensor nodes into 3 clusters, i.e. K = 3.

Consider a data set of 10 sensor nodes as follows:

$S_1 = (2, 4)$ $S_6 = (5, 7)$

$S_2 = (3, 6)$ $S_7 = (6, 9)$

$S_3 = (4, 8)$ $S_8 = (4, 3)$

$S_4 = (3, 9)$ $S_9 = (3, 3)$

K-Center Clustering

1. Input: S = Set of sensor nodes deployed at known locations, $S_i = \{(a, b) \mid a, b \in Z^+\}$ and $S = \{\{S_i\} \mid i \in Z^+\}$; K = Number of Clusters.

2. Initialization: C = set of centroids, $C = \{\{C_i\} \mid C_i \subseteq S\}$; $Cluster_j$ = cluster set j containing the centroid C_i and the node (s) associated with that centroid, $Cluster_i = \{\{C_i, S_i\} \mid C_i \in C, S_i \in S\}$; Max_Dist = 0. $1 \leq i, j \leq K$.

3. Initial Step: (a) Randomly select K centroids from S;

(b) Form set C of K chosen centroids;

(c) Calculate the distance of each node $S_i \in (S - C)$ from each centroid $C_i \subset C$ and store all these distances in a matrix;

(d) Compare each of those distances and select the C_i having the lowest distance with S_i.

(e) Form K cluster sets $Cluster_j$'s $(1 \leq j \leq K)$ by associating each S_i to its closest C_i such that each cluster set contains the centroid along with all its associated nodes;

(f) Calculate the maximum distance of each S_i to its closest C_i and store it in Max_Dist.

4. Iteration Step: (a) For each centroid $C_j \subset C$

(i) For each node $S_i \in (S - C)$ /*Chosen randomly*/

1. Swap C_j & S_i and repeat steps 3(b) to 3(f).

(ii) End For

(b) End For

5. Select the set C having the lowest maximum distance.

6. If C changes then repeat step 4 else go to step 7.

7. Output: C = Set of K centroids;

$Cluster_j$'s = Cluster sets containing sensor nodes associated to their closest centroids. $1 \leq j \leq K$

Figure 6. K-Center Clustering Algorithm

$S_5 = (8, 5)$ $S_{10} = (9, 4)$

Figure 7 shows all sensor nodes for calculating K-Center clustering technique.

Step 1: Initialize K Centroids

Centroids $C = \{C_1, C_2, C_3\} = \{(4, 8), (6, 9), (5, 7)\}$

Let us assume $C_1 = (4, 8)$, $C_2 = (6, 9)$ & $C_3 = (5, 7)$

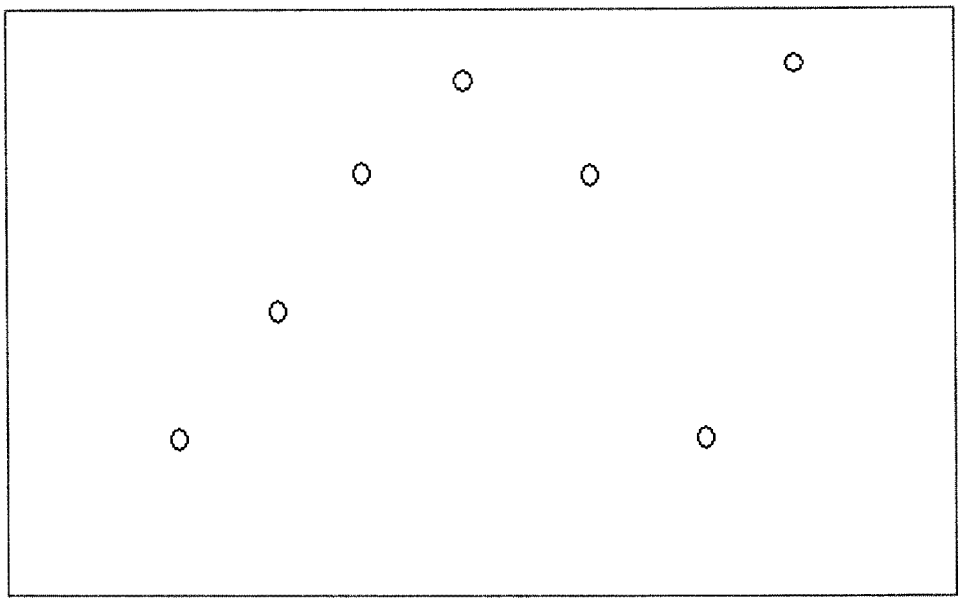

Figure 7. All Sensor Nodes for K-Center Example.

Calculate distance matrix for each centroid using Euclidean distance. Table 3 illustrates the calculated distance matrix below.

Now proceeding row-wise and comparing the distance values in each row we deduce which node should belong to which cluster. So we start with the first row and compare distance values of a node with each centroid, i.e., compare dist ((4, 8), (2, 4)), dist ((6, 9), (2, 4)) and dist ((5, 7), (2, 4)), and find the lowest value among these distances. In this case, it comes out to be 4 (in case of a tie, choose any one). We have chosen (4, 8) as the centroid for (2, 4).

Likewise we compare the distance values for each row and evaluate the minimum distance and on the basis of that assign each node to its closest centroid.

Then Clusters become:

$Cluster_1$= {(4, 8), (2, 4), (3, 6), (3, 9)}

Table 3. Distance Matrix for Step 1 of K-Center Example.

	Sensor Nodes			Sensor Nodes			Sensor Nodes	
C_1	S_i	Distance	C_2	S_i	Distance	C_3	S_i	Distance
4,8	2,4	4	6,9	2,4	6	5,7	2,4	4
4,8	3,6	2	6,9	3,6	4	5,7	3,6	2
4,8	3,9	1	6,9	3,9	3	5,7	3,9	2
4,8	8,5	5	6,9	8,5	4	5,7	8,5	3
4,8	4,3	5	6,9	4,3	6	5,7	4,3	4
4,8	3,3	5	6,9	3,3	6	5,7	3,3	4
4,8	9,4	6	6,9	9,4	5	5,7	9,4	5

$Cluster_2$ = {(6, 9), (9, 4)}

$Cluster_3$ = {(5, 7), (8, 5), (4, 3), (3, 3)}

Since the nodes that are close to their respective centroids form a cluster

Maximum distance = Max (Max {(dist (4, 8), (2, 4))), (dist ((4, 8), (3, 6))), (dist (4, 8), (3, 9)))}

+ Max {dist (6, 9), (9, 4))}

+ Max {(dist ((5, 7), (8, 5))), (dist ((5, 7), (4, 3))), dist ((5, 7), (3, 3)))})

= Max (Max (4, 2, 1), Max (5), Max (3, 4, 4))

= Max (4, 5, 4)

= **5**

Step 2: Select a non-centroid C_1' (another sensor node which is not present in the initial set of centroids) and replace it with C_1.

Let us assume $C_1' = (3, 3)$

Now C = {(3, 3), (6, 9), (5, 7)}

Calculate distance matrix for each centroid again using Euclidean distance. Table 4 illustrates the calculated distance matrix below.

Now new clusters become

$Cluster_1$ = {(3, 3), (2, 4), (4, 3)}

$Cluster_2$ = {(6, 9), (9, 4)}

$Cluster_3$ = {(5, 7), (3, 6), (4, 8), (3, 9), (8, 5)}

Maximum dist = Max (Max {((3, 3), (2, 4)), (dist ((3, 3), (4, 3)))}

+ Max {dist ((6, 9), (9, 4)))}

+ Max {dist ((5, 7), (3, 6)), (dist ((5, 7), (4, 8))), ((dist ((5, 7), (3, 9))), (dist ((5, 7), (8, 5)))})

= Max (Max (1, 1), Max (5), Max (2, 1, 2, 3))

= Max (1, 5, 3)

= **5**

Since 5 <= 5, so we are not replacing (4, 8) by (3, 3) and likewise we continue to swap each centroid with each non-centroid one by one (for all centroids till we get the lowest distance). In case of a tie with the lowest distance select any node out of the two choices.

4.3. Minimum Spanning Tree

After the implementation of clustering techniques we present algorithm for Minimum Spanning Tree (MST) [30] given in Figure 8.

Table 4. Distance Matrix for Step 2 of K-Center Example.

	Sensor Nodes			Sensor Nodes			Sensor Nodes	
C1	SI	Distance	C2	SI	Distance	C3	SI	Distance
3,3	2,4	1	6,9	2,4	6	5,7	2,4	4
3,3	3,6	3	6,9	3,6	4	5,7	3,6	2
3,3	4,8	5	6,9	4,8	2	5,7	4,8	1
3,3	3,9	6	6,9	3,9	3	5,7	3,9	2
3,3	8,5	5	6,9	8,5	4	5,7	8,5	3
3,3	4,3	1	6,9	4,3	6	5,7	4,3	4
3,3	9,4	6	6,9	9,4	5	5,7	9,4	5

1. Initial Step: S = Set of K sensor heads (SHs) and distances among all SH's;

   K = Number of SH's.

2. Initial Step: (a) Select a SH $s_i \epsilon S$;

   (b) Identify all the neighboring vertices of $s_j \epsilon S$;

   (c) Select the vertex s_j with the lowest distance from s_i;

   (d) If Selection of s_j forms a cycle, i.e., a path direct to s_i, select another vertex s_j, else go to step 2(e).

3. Iteration Step: Repeat steps 2(b) to 2(d).

4. Output: MST = Minimum Spanning Tree containing the shortest route (Sum of the distances of s_i to all s_j's).

Figure 8. Minimum Spanning Tree Algorithm

The complexity of the algorithm is given by $O(|C|^2)$, where C represents the number of centroids (or SHs) for a given network.

Now we present a detailed step-by-step example which describes the algorithm. Figures 9 through 13 illustrate the example.

Figure 9 shows an undirected graph for MST. Let us assume we obtained the following set of Sensor Heads, S = {A, B, C, D} representing the vertices of a graph. The numbers near the edges indicate the distance between the vertices. The graph is shown below as:

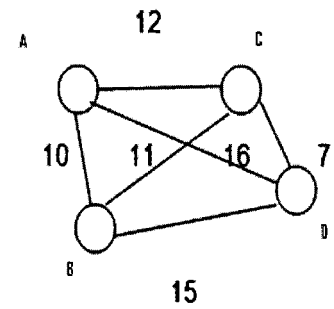

Figure 9. An Undirected Graph for MST.

Step 1: Vertex B has been randomly chosen as the initial vertex for the Minimum Spanning Tree (MST). Figure 10 shows the Initial node B for the MST.

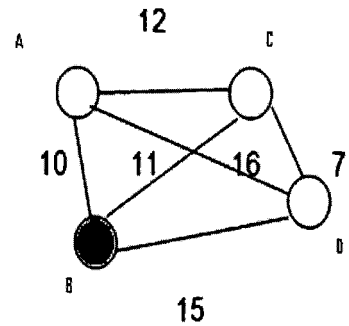

Figure 10. Node B is the Initial Node for the MST.

Step 2: Vertices A, C and D are the neighbors of vertex B. As can be seen vertex A has the minimum distance among all the 3 neighbors, i.e., 10 and hence chosen as the next vertex for the MST. At every step of the iteration we perform a check whether selecting a vertex should not form a cycle, i.e., a vertex should not form a path that directs to itself. Figure 11 shows the initial node for the MST.

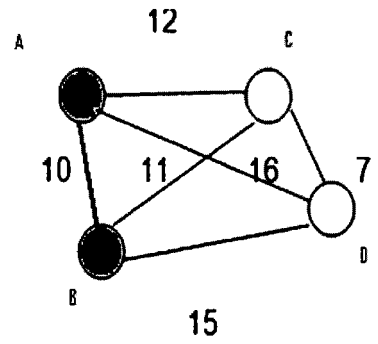

Figure 11. Node A is the Next Node for the MST.

Proceeding in this fashion we obtain the next vertex to be vertex C and finally vertex D. Figure 12 indicates the result of MST.

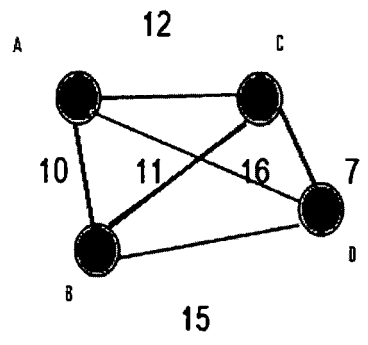

Figure 12. Nodes C & D Complete the MST.

Now all the vertices have been selected and the minimum spanning tree thus obtained has weight 28. The resulting Minimum Spanning Tree (MST) is shown below. Final Minimum spanning tree is indicated in Figure 13.

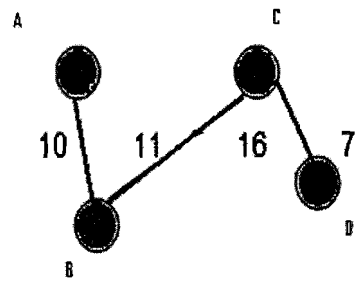

Figure 13. Final MST.

The Euclidean distance between two arbitrary nodes (or points) can be calculated as:

$$d\,(i,\,j) = \sqrt{(y_i - y_j)^2 + (x_i - x_j)^2}\,,\ \text{where}$$

x_i and y_i correspond to the x and y coordinates of the i^{th} node, and

x_j and y_j correspond to the x and y coordinates of the j^{th} node.

Figure 14 illustrates the flow diagram for the entire generalized procedure for the clustering of nodes.

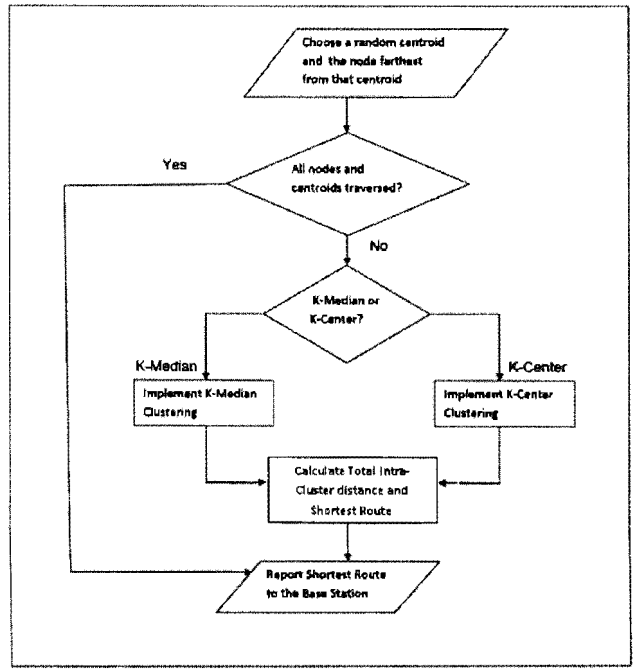

Figure 14. Flow Diagram for the Facility Location Problem

5. EXPERIMENTAL RESULTS AND ANALYSIS

We have implemented the *K-Median* and *K-Center* Clustering in Java using NetBeans IDE 6.7.1. We have designed a Graphical User Interface (GUI) using the Java applet mechanism for choosing a set of criterion to implement both the clustering mechanisms.

All the sensor nodes are deployed in a fixed 500px X 300px region of interest. The values of the number of clusters vary from 3 up to 8 in increments of 1, and the values of the sensor nodes vary from 5 up to 50 in increments of 5.The shortest route is the route that traverses through all the sensor heads to the base station visiting each sensor head exactly once.

Figures 15 through 19 present some of the screen captures indicating the experimental setup in Java.

Figure 15 illustrates the initial setup for our experiments. The user can draw a set of nodes in the region of interest according to his/her choice (s). Once the nodes are being drawn then the buttons for running the chosen algorithm (*K*-Median or *K*-Center) are enabled. The user can choose either of the two clustering schemes to start the running of the algorithm. Also the user can choose among the number of clusters from 3 up to 8. Below the region of interest lies a generalized description of the algorithm to cluster the nodes. During a step-by-step running of the algorithm the control traverses among all the steps before finally converging to the last step, i.e., Step 5 of the description.

Figure 16 shows a complete run the *K-Center Clustering* algorithm for 3 clusters. The colors are RED, BLUE and GREEN indicating the 3 clusters.

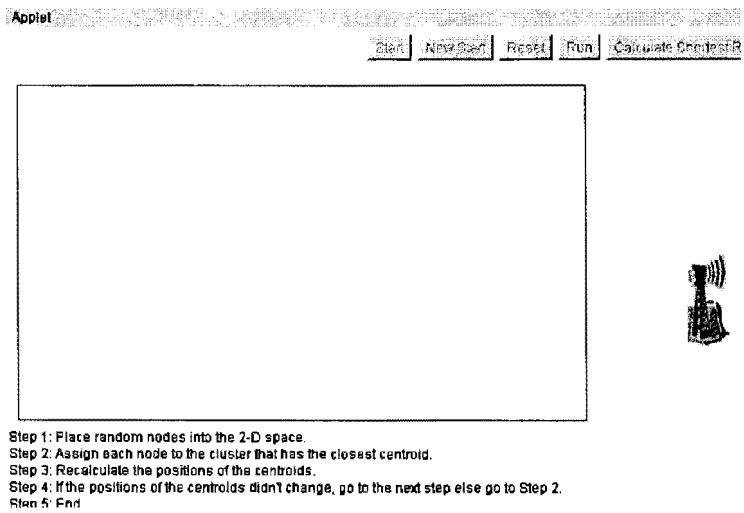

Figure 15. Basic Experimental Setup

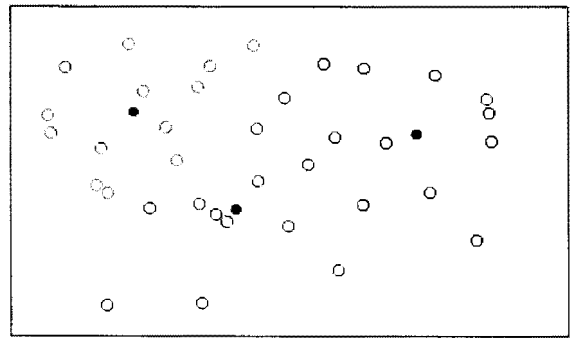

Figure 16. Complete Run of K-Center Clustering for 3 Clusters

Figure 17 illustrates an example showing the clustering of nodes in 3 different clusters forming a pattern or shape.

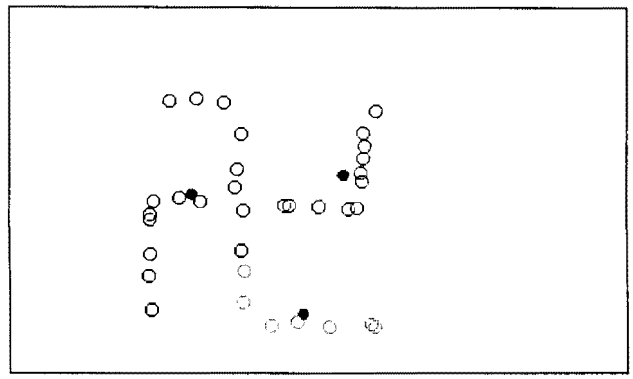

Figure 17. Clustering of Nodes in 3 Different Clusters

Figure 18 shows another clustering pattern for a large number of deployed sensor nodes for K-median into 8 clusters each with a different color.

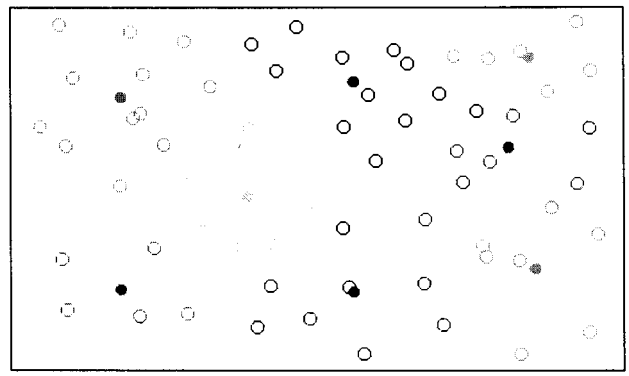

Figure 18. Clustering Pattern for Nodes with 8 SH

Figure 19 represents a 500 sensor node deployment in the experimental setup.

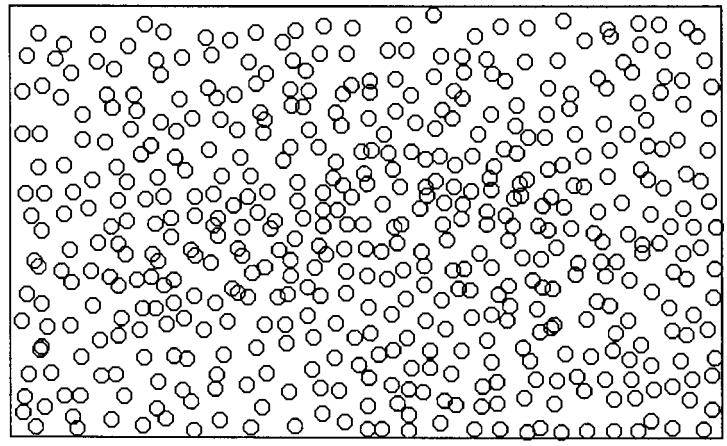

Figure 19. Deployed 500 Sensor Nodes

For testing our implemented procedure we have made use of the following test criterion conducted in *MS Excel 2007*:

5.1. Test Criteria 1.1

We choose the number of clusters as 3. We vary the number of nodes from 20 up to 100 in increments of 10 and run our simulation. The output is the distance in distance units. We then plot a graph for these observed values with number of nodes on the x-axis and the distance on the y-axis illustrated in Figure 20. Table 5 illustrates the shortest route for 3 clusters.

Table 5. Distances for Test criteria 1.1.

	Shortest Route		Total intra-cluster Distance		
Number of Nodes	**K-Median**	**K-Center**	**K-Median**	**K-Center**	**% increases for TICD**
20	371.03	368.18	1411.17	1490.24	5.65
30	321.82	372.45	2232.11	2377.42	6.54
40	319.92	362.33	2987.49	3014.01	0.88
50	333.25	328.32	3866.98	3947.15	2.07
60	335.31	385.64	4821.79	4884.87	1.33
70	365.83	319.26	5364.92	5656.82	5.44
80	320.69	322.52	6306.47	6371.83	1.03
90	315.76	313.64	7058.52	7061.14	0.03
100	311.59	373.19	7843.27	7948.65	1.34

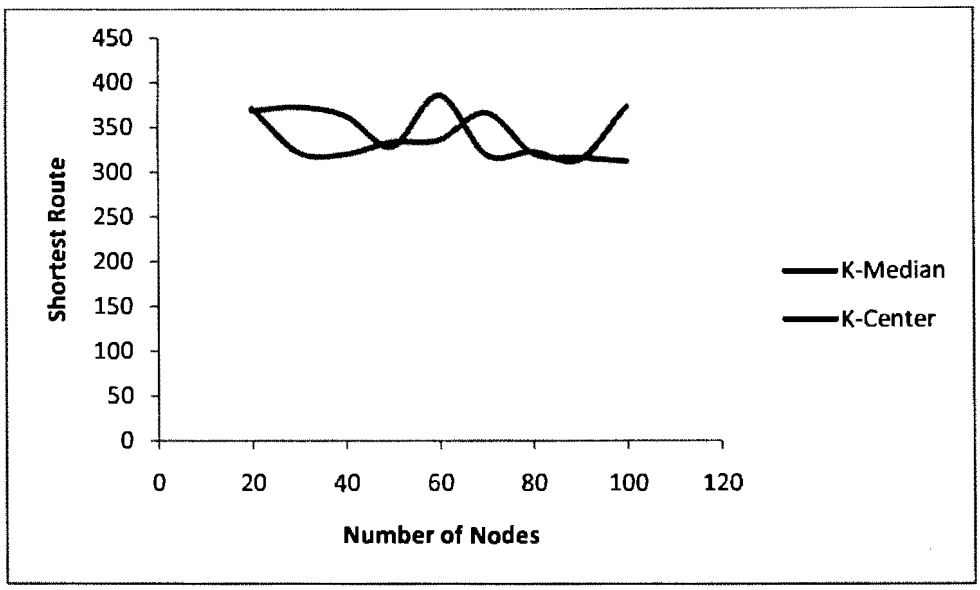

Figure 20. Shortest Route for Test Criteria 1.1.

The data to be transferred to the base station consists of 17 nodes (20 sensor nodes – 3 SHs) carrying about 1411.18 distance units of data via the shortest route of 368.18 distance units when the clustering technique is *K-Median*, and 17 nodes (20 sensor nodes – 3 SHs) carrying about 1490.24 distance units of data via the shortest route of 371.03 distance units when the clustering technique is *K-Center*. Likewise, when the number of nodes is increased the corresponding data gets transferred to the base station.

% increases for TICD means the distance units of k-center is greater than the distance units of k-median. In Table 5, all the values are positive which reflects K-median technique performs better than K-center technique.

Figure 21 shows the corresponding plot for the values illustrated in Table 5. It can be easily interpreted from the plot that both k-median and k-center algorithms show little variations when the number of nodes are increased.

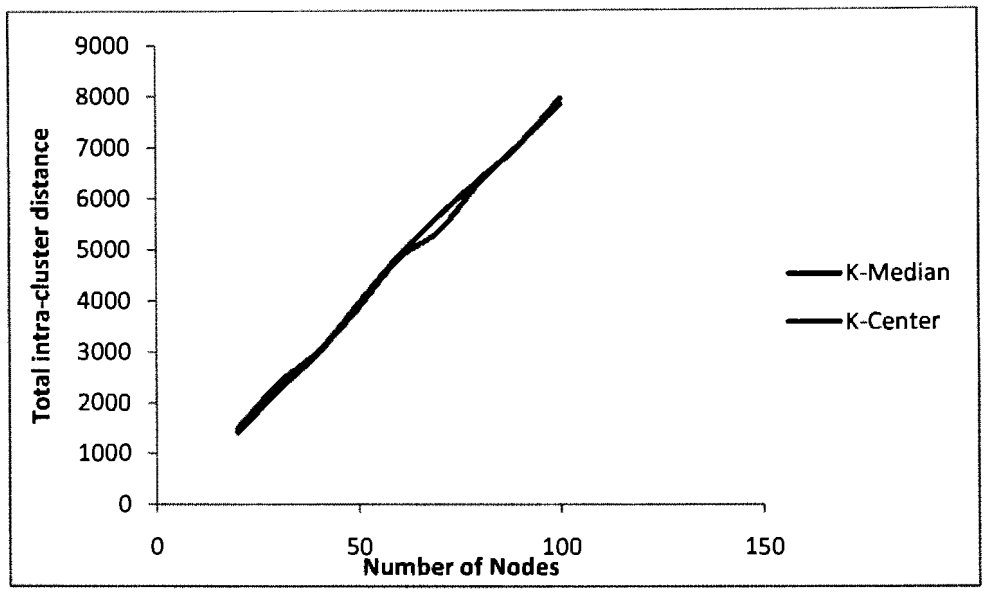

Figure 21. Total Intra-Cluster Distance for Test Criteria 1.1.

5.2. Test Criteria 1.2

We choose the number of nodes as 50. We vary the number of clusters from 3 up to 8 and run our simulation. The output is the distance in distance units. We then plot a graph for these observed values with number of clusters on the x-axis and the distance on the y-axis. Table 6 illustrates the total intra-cluster distances for 3 clusters.

Figures 22 and 23 show the corresponding plot for the values illustrated in Table 6. It can be easily interpreted from the plot (Figure 22) that as the number of clusters is increased the distance in both k median and k center algorithms is also increased and from the plot (Figure 23) when the number of clusters is increased the distance in both k median and k center algorithms gets decreased.

Table 6. Distances for Test criteria 1.2.

	Shortest route		Total intra-cluster distance		
Number of Clusters	K-Median	K-Center	K-Median	K-Center	% increases for TICD
3	226.61	221.14	2795.41	2706.56	-3.28
4	332.05	336.92	2350.1	2564.32	8.35
5	409.09	385.96	2066.62	2146.75	3.75
6	421.57	458.43	1917.13	1896.08	-1.11
7	497.68	492.74	1789.48	1864.8	4.03
8	520.79	569.13	1680.04	1667.86	-0.73

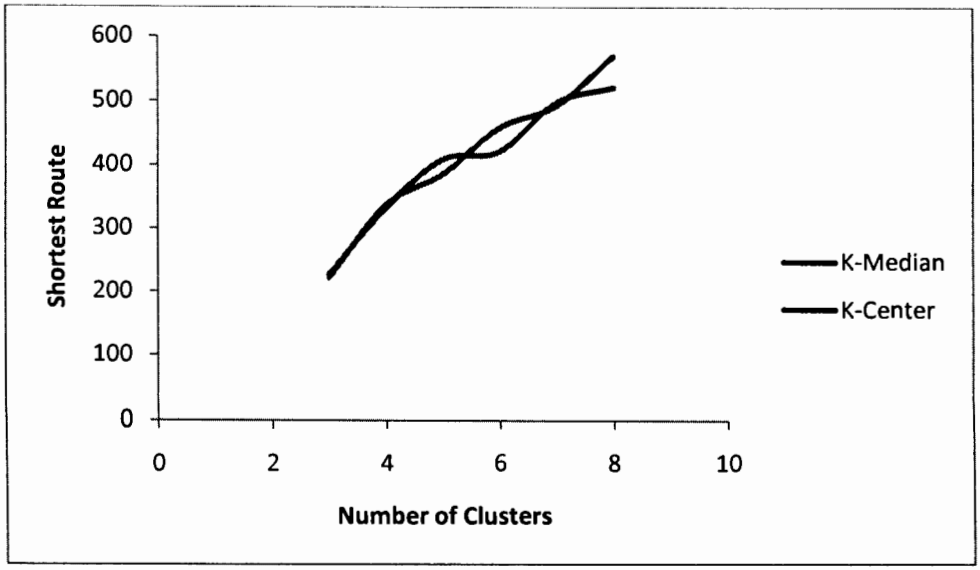

Figure 22. Shortest Route for Test Criteria 1.2.

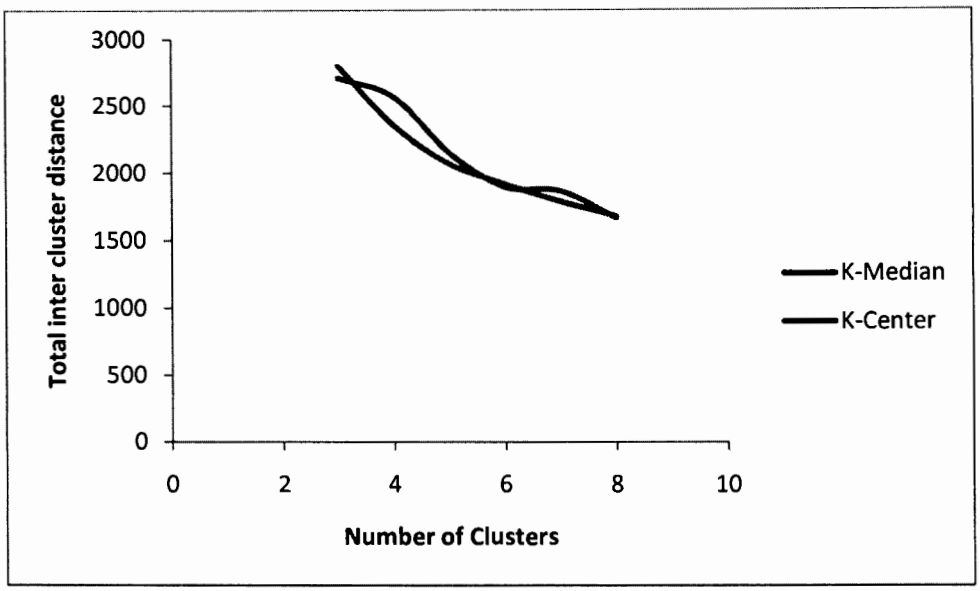

Figure 23. Total Intra-Cluster Distance for Test Criteria 1.2.

% increases for TICD means the distance units of k-center is greater than the distance units of k-median. In Table 6, some of the values are positive and some are negative which could occur because of the variation in the placement of nodes and/or formation of clusters. But overall we can infer that K-median performs relatively better than K-center technique.

5.3. Statistical Analysis

To further test the accuracy of our experimental results we did some statistical analyses of some set of values. All statistical analyses are conducted in MS Excel. A confidence interval gives an estimated range of values from a set of sample data. In our experimental results the sample data is the range of the *shortest route* and the *Total intra-cluster distance between the nodes and the sensor heads (SHs)*. The objective is to achieve a 95% confidence interval for the mean value (s) which gives us the upper and lower bounds for the *shortest route*

and the *Total intra-cluster distance between the nodes and the sensor heads (SHs).*

We ran our simulator 9 times for a specified set of parameters. For each run we recorded the *shortest route* and *Total intra-cluster distance* as our observed values. Both the starting and ending ranges are included. The formulae for calculating the starting and ending ranges for a 95% confidence interval around the average value are given as:

Starting Range = Average value – (1.96 * Standard Deviation) / $\sqrt{\text{Number of Runs}}$

Ending Range = Average value + (1.96 * Standard Deviation) / $\sqrt{\text{Number of Runs}}$

1.96 stands for factor corresponding to 95% confidence level.

5.3.1. Criteria 1

Figure 24 outlines the specifications for evaluating the first criteria for conducting the statistical analysis.

SR = Shortest Route (in units)

TICD = Total intra-cluster distance (in units)

95% confidence interval for Criteria 1 marked in BLUE for K-median and ORANGE for K-center.

It can be inferred from the results that with 95% confidence, the shortest route will lie between 302 and 380 for K-median clustering, and between 305 and 378 for K-center clustering for 25 sensor nodes and 3 clusters.

Number of Runs	Clustering Type			
	SR	TICD	SR	TICD
1	255.44	1299.63	251.21	1422.87
2	336.52	1729.47	313.39	1691.22
3	409.13	1911.51	380.99	1928.43
4	340.54	1772.64	332.68	1738.53
5	354.72	1203.99	352.71	1188.33
6	251.53	1687.89	312.09	1240.02
7	396.45	1939.56	418.92	1946.19
8	312.71	1841.67	298.58	1828.95
9	410.44	1945.77	414.23	2071.62
Number of Nodes	25			
Number of Clusters	3			
Average	340.83	1703.57	341.64	1672.9
Standard Deviation	60.07	272.74	55.56	318.72

Figure 24. Statistical Result for Criteria 1.

5.3.2. Criteria 2

Figure 25 outlines the specifications for evaluating the second criteria for conducting the statistical analysis.

Number of Runs	Clustering Type			
	SR	TICD	SR	TICD
1	485.85	1346.12	444.96	1391.95
2	530.59	1763.93	523.46	1542.75
3	467.49	1428.99	470.29	1166.45
4	516.48	1454.71	515.61	1377.77
5	529.11	1396.75	550.17	1343.89
6	497.65	1081.45	460.07	1272.75
7	501.34	1341.75	523.75	1028.25
8	593.17	1341.75	614.21	1296.55
9	512.81	1289.15	571.85	1161.81
Number of Nodes	25			
Number of Clusters	5			
Average	514.94	1382.73	519.37	1286.9
Standard Deviation	35.62	179.25	55.01	152.21

Figure 25. Statistical Result for Criteria 2.

SR = Shortest Route (in units)

TICD = Total intra-cluster distance (in units)

95% confidence interval for Criteria 2 marked in BLUE for K-median and ORANGE for K-center.

It can be inferred from the results that with 95% confidence, the shortest route will lie between 492 and 538 for K-median clustering, and between 483 and 555 for K-center clustering for 25 sensor nodes and 5 clusters.

5.3.3. Criteria 3

Figure 26 outlines the specifications for evaluating the third criteria for conducting the statistical analysis.

Number of Runs	Clustering Type			
	SR	TICD	SR	TICD
1	339.78	1353.61	331.54	1236.57
2	352.64	1050.82	359.32	1120.18
3	387.79	1269.04	390.17	1429.71
4	359.38	1228.94	344.48	1276.89
5	356.13	1110.97	392.44	1197.47
6	373.78	1350.25	318.64	1335.46
7	299.65	1204.58	317.13	1215.99
8	314.62	1066.81	313.22	1062.19
9	282.45	1117.51	317.39	1199.5
Number of Nodes	50			
Number of Clusters	3			
Average	340.69	1194.72	342.7	1230.44
Standard Deviation	35	115.25	31.37	109.34

Figure 26. Statistical Result for Criteria 3.

SR = Shortest Route (in units)

TICD = Total intra-cluster distance (in units)

95% confidence interval for Criteria 3 marked in BLUE for K-median and ORANGE for K-center.

It can be inferred from the results that with 95% confidence, the shortest route will come between 318 and 364 for K-median clustering, and between 322 and 363 for K-center clustering for 50 sensor nodes and 3 clusters.

5.3.4. Criteria 4

Figure 27 outlines the specifications for evaluating the fourth criteria for conducting the statistical analysis.

Number of Runs	Clustering Type			
	SR	TICD	SR	TICD
1	429.18	2328.9	434.25	2257.75
2	528.04	2644.3	456.43	2780.7
3	496.11	2603.6	507.21	2567.05
4	531.25	2763.2	494.51	2562.1
5	503.47	2457.25	418.57	2488.35
6	418.53	1912.15	401.01	1877.75
7	441.85	1781.9	417.75	1854.1
8	421.71	2611.3	433.31	2559.65
9	388.86	2037.4	380.22	2242.25
Number of Nodes	50			
Number of Clusters	5			
Average	462.11	2348.88	438.14	2354.41
Standard Deviation	52.89	356.02	41.61	322.02

Figure 27. Statistical Result for Criteria 4.

SR = Shortest Route (in units)

TICD = Total intra-cluster distance (in units)

95% confidence interval for Criteria 4 marked in BLUE for K-median and ORANGE for K-center.

It can be inferred from the results that with 95% confidence, the shortest route will come between 427 and 496 for K-median clustering, and between 410 and 465 for K-center clustering for 50 sensor nodes and 5 clusters.

The experimental and statistical analyses' results are summarized as follows:

1. Both the clustering techniques have performed relatively better when the network is sufficiently dense.
2. Larger cluster sizes have led to larger routes and even larger Intra-cluster distances.
3. Generally *K-Median* technique has performed better (Total % increases in Test Criteria 1 for TICD = 24.35% and in Test Criteria 2 for TICD = 10.99%) than *K-Center* technique in terms of routing useful data to the base station and in determining the "closeness" of nodes to their respective sensor heads, when the network is sufficiently dense.
4. Both the procedures have performed relatively well in both sparse and dense topologies.
5. The statistical analysis has revealed robust results for both the clustering techniques when the network is sufficiently dense.

6. CONCLUSION AND FUTURE WORK

In this paper we have addressed the problem of clustering a given set of nodes into *K* known clusters using two clustering mechanisms, namely *K-Median* and *K-Center* clustering. After data clustering, we transmit the data to the base station via only the cluster (or sensor) heads formed during clustering for every cluster. Then a performance analysis is done comparing both the clustering techniques on the basis of the "shortest route" traversed and the "Total intra-cluster distance" among all the nodes indicating the "closeness" of these nodes to their respective sensor heads. From the experimental experiences, we have learnt that k-median technique has performed relatively better than k-center technique.

This technique finds various applications in the fields of facility location, feature selection and minimizing core sets of data points.

In future we plan to extend our work to comparing more clustering techniques based on various parameters. Another area could be the experimentation with mobile sensor nodes instead of considering static sensor nodes. Another aspect could be considering solving the proposed problem with the help of integer linear programming techniques in place of considering a heuristic solution.

7. REFERENCES

1. W. R. Heizelman, A. Chandrakasam and H. Balakrishnan. Energy Efficient communication protocols for wireless micro sensor networks. *In proceedings of Hawaiian International Conference on Systems Science*. 2000.
2. S. Bandyopadhyay and E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. 2003.
3. S. Bandyopadhyay and E. J. Coyle. Minimizing communication costs in hierarchically-clustered networks of wireless sensors. 2004
4. J. Deng, Y. S. Han, W. B. Heinzelman and P. K. Varshney. Scheduling sleeping nodes in high density cluster based sensor networks. *In ACM/Kluwer Mobile Networks and Applications*, Springer, Netherlands. 2005.
5. J. Deng, Y. S. Han, W. B. Heinzelman and P. K. Varshney. Balanced-energy sleep scheduling scheme for high density cluster-based sensor networks. *In Proc. Of the 4th Workshop on Application and Services in Wireless Networks (ASWN '04)*, Boston, Massachusetts. 2004.
6. K. Chen. On k-median clustering in high dimensions. *In Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*. 2006.
7. M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the *k*-median problem. In *Proc. 31st Annu. ACM Sympos. Theory Comput.* 1999.
8. T.Furuta, M.Sasaki, F.Ishizaki, A.Suzuki and H.Miyazawa. A new clustering algorithm using facility location theory for wireless sensor networks. *Technical Report of the Nanzan Academic Society Mathematical Sciences and*

Information Engineering (NANZAN-TR-2006-04), submitted to European Journal of operational Research, 2007.

9. J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*. 2004.
10. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. A survey on Sensor Networks. 2002.
11. A. Tamir. An $O(pn^2)$ algorithm for the p-median and related problems on tree graphs. 1996.
12. S. Arora, P. Raghavan and S. Rao. Approximation schemes for Euclidean k-median and related problems. *In Proceedings of the 30th Annual ACM Symposium on Theory of* Computing. 1998.
13. J. Lin and J. S. Vitter. ε-approximation with minimum packing constraint violation. *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*. 1992.
14. J. Lin and J. S. Vitter. Approximation Algorithms for Geometric Median Problems. *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*. 1992.
15. M. Korupolu, C. Plaxton and R. Rajaraman. Analysis of a local search heuristic for faculty location problems. *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*. 1998.
16. M. Charikar, S. Guha, E. Terdos and D. Shmoys. A Constant Factor Approximation Algorithm for the k-median Problem. *Journal of Computer and System Sciences*. 2002.

17. M. Charikar, S. Khuller, D. Mount and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proc. SODA*. 2001.

18. L. Wang and Y. Xiao. A survey of energy saving mechanisms in sensor networks. *In Proc. of 2nd Int'l Conference on Broadband Networks*. 2005.

19. V. Arya, N. Garg, R. Khandekar, K. Munagala, and V. Pandit. Local search heuristic for k-median and facility location problems. *In Proc. 33rd Annu. ACM Sympos. Theory Comput.* 2001.

20. T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.* 1985.

21. J. Kleinberg and E. Tardos. Algorithm design. Addison-Wesley, 2006.

22. T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom. Theory Appl.* 2004.

23. D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the *k*-center problem. *Mathematics of Operations Research.* 1985.

24. Z. He. Farthest-point heuristic based initialization methods for k-modes clustering. 2006.

25. O. L. Mangasarian and E. W. Wild. Feature selection in k-median clustering. *Proceedings of SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data and its Applications*. 2004.

26. S. Har-Peled and S. Mazumdar. Coresets for k-means and k-median clustering and their applications. *In Proc. 36th Annu. ACM Sympos. Theory Comput.* 2004.

27. S. Nickel and J. Puerto. A unified approach to network location problems. *Networks 34*, 1999.

28. L. Gouveia, A. Paias and D. Sharma. Modeling and solving the rooted distance-constrained minimum spanning tree problem. *Computers and Operations Research*, 2008.

29. D. Shmoys, E. Tardos and K. Aardal. Approximation algorithms for facility location problems. *In Proc. Of the 29th ACM Symposium on Theory of Computing*, 1997.

30. E. Gonina and L. Kale. Parallel Prim's algorithm on dense graphs with a novel extension. *Technical Report*, 2007.

31. http://www.eecs.harvard.edu/~konrad/projects/motetrack/manual/figs/mica2.jpg last retrieved on 09/07/2010.