

Fuzzy Clustering with Balance Constraint

by

Siamak Naderi Varandi

Submitted to the Graduate School of Engineering and Natural Sciences

in partial fulfilment of

the requirements for the degree of

Master of Science

August 2014

FUZZY CLUSTERING WITH BALANCE CONSTRAINT

APPROVED BY

Assoc. Prof. Dr. Kemal Kılıç

(Thesis Supervisor)

Assist. Prof. Dr. Gürdal Ertek

Assoc. Prof. Dr. Nihat Kasap

DATE OF APPROVAL: 06.08.2014

©Siamak Naderi Varandi 2014

All Rights Reserved

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Dr. Kemal Kılıç for the continuous support of my study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my study.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Gürdal Ertek and Dr. Nihat Kasap.

One of the joys of completion is to look over the past and remember all the friends and family who have helped and supported me along this short but fulfilling journey. I thank my fellow officemates in FENS 1021 for all the fun we have had in the last one and half years.

Last but not the least, I would like to thank my family specially my beloved one, for two years of support, patient and love.

FUZZY CLUSTERING WITH BALANCE CONSTRAINT

Siamak Naderi Varandi

Industrial Engineering, Master's Thesis 2014

Thesis supervisor: Assoc. Prof. Dr. Kemal Kılıç

Keywords: Fuzzy Clustering, Equality, Districting, Lagrangean Relaxation,
LEACH protocol

Abstract

We study equality in fuzzy clustering algorithms where an equality constraint is added to the existing model. Equality is being used in various areas, such as districting (either zonal or political), industries (distribution companies). We focus on wireless sensor networks problem. Existing protocols do not pay too much attention to the cluster head selection step and equality of workload of the clusters. These two issues have significant effect on the consumption of energy in a network where increasing lifetime of network is critical. A solution approach based on the Lagrangean relaxation is developed. The proposed algorithm is compared with the popular LEACH protocol. Results show that in the same simulated environment, our algorithm works better.

List of Figures

1.1	Clustering of data into four clusters	11
2.1	Example of Agglomerative Hierarchical Clustering	18
2.2	A convex function [45]	20
2.3	A neither convex nor concave function [45]	20
2.4	Global and Local Optima [45]	24
5.1	Effect of g on the Remaining Energy of Network, $n = 100$	53
5.2	Effect of g on the Number of Alive Sensors, $n = 100$	54
5.3	Effect of Repulsion on the Number of Alive Sensors	54
5.4	Effect of Repulsion on the Remaining Energy of Network	55
5.5	Effect of g on the Remaining Energy of Network	55
5.6	Effect of Repulsion on the Number Alive Sensors	56
5.7	Effect of number of clusters on the Total Remaining Energy	56
5.8	Effect of number of clusters on the Number of Alive sensors	57
5.9	CHs' locations over time in a 500×500 m space	57

List of Tables

2.1	Optimality conditions for minimization problem [45]	29
2.2	Optimality conditions for maximization problem [45]	29
5.1	Experiment set up	52

Contents

1	Introduction	10
2	Theoretical Background	14
2.1	Clustering	14
2.1.1	Distance Measure	15
2.1.2	Application Areas of Clustering	16
2.1.3	Hard (i.e., Crisp) vs. Soft (i.e., Fuzzy) Clustering	16
2.1.4	Popular Hard Clustering Algorithms	17
2.2	Optimization	19
2.2.1	Convexity	19
2.2.2	Optimization	23
2.3	Fuzzy C-means Clustering	29
2.3.1	FCM- Necessary and Sufficient conditions	31
3	Literature Review	34
3.1	FCM and Some Extensions	34
3.2	Clustering with Balance Constraint; Zoning/Districting Problems	38
4	Fuzzy Clustering with Balance Constraint	40

<i>CONTENTS</i>	9
4.1 Problem Statement	40
4.2 Repulsion	45
4.3 Optimality	46
5 Application in Wireless Sensor Networks	48
5.1 Computational Experiments	51
6 Conclusion and Future Research	58

Chapter 1

Introduction

Clustering is identification of natural groups (i.e., clusters) in an environment. The “natural grouping” has different definitions in various contexts. In demographics, clustering refers to the gathering of various populations based on factors such as ethnicity, economics or religion. In graph theory, clusters refer to the linked nodes in a network, measured by the clustering coefficient. On the other hand, in the context of Data Mining (DM), clusters refer to groups of objects that are similar to each other and different from others in terms of a similarity metric.

Briefly speaking data mining is extracting knowledge (or *patterns*) from data , i.e., learning from data.

Learning methods can be categorized as *supervised* and *unsupervised* learning. In supervised learning one tries to model data based on labelled training data. Bayes Network, Decision Trees, OneR, IBK and etc are examples of popular supervised learning methods. On the other hand, in unsupervised learning the task is finding hidden structure in unlabelled data. This realized by various clustering methods such as hierarchical clustering, K-means clustering and DSSAN.

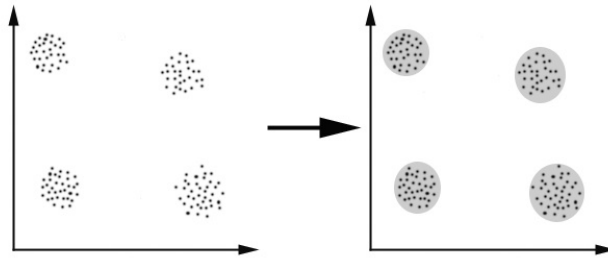


Figure 1.1: Clustering of data into four clusters

The objective of clustering is grouping the objects in such a way that the most similar ones gather in the same group. Furthermore, this similarity should be minimized compared to the objects of the other groups (i.e., dissimilar as much as possible). Figure 1.1 illustrates a simple example of clustering in which given data is grouped to four clusters.

In this example, similarity criterion is distance: two or more objects belong to the same cluster if they are “close” to each other in the plane with respect to a given distance.

Clustering algorithms can be categorized into two approaches, namely hard clustering and soft (i.e., fuzzy) clustering. In the case of hard clustering, in any grouping algorithm a binary assignment variable (u_{jk}) is considered to be 1 if a data point k is assigned to group or cluster j and 0 otherwise. On the other hand, in fuzzy clustering the assignments are no longer crisp (i.e., binary). Instead, the assignments are fuzzy, which means that the data point k is assigned to **cluster** j with a membership degree (u_{jk} , which is usually a function of the distance between data point and the cluster center) which is a number between zero and one. Generally, in fuzzy clustering, the summation of memberships of data point to all clusters is required to be one, ([16], [39] and [40]). A popular exception is possibilistic c-means which we will discuss later.

Fuzzy pattern recognition was introduced by Bill Wee as the first time in his Ph.D. thesis. Then Ruspini [21] discussed fuzzy clustering for the first time, in which constraint

c -partitions of unlabelled data. Dunn [6] and Bezdek [8] published the first paper on fuzzy c -means clustering.

One of the most important issues that in the recent years is attracting a lot of attentions is equality. Equality can be defined in various contexts. For example, governments are trying to assign the equal resources to the different groups of people (budget equality). Another example is in industries where companies are trying to assign equal workload among their employees or assign equal resources to the different regions they cover. Consider a distribution company which is serving a city by use of multiple number of vehicles. A big concern is such companies is assigning an equal workload to each vehicle. In some cases, equality is being done in the purpose of fairness as mentioned in either governmental applications or industries. On the other hand, equality has optimization advantages in some applications. *Wireless Sensor Networks (WSNs)* is one of those applications where equality must be taken into account because of its optimization. Later we will discuss WSNs in details.

Next in Chapter 2, we will introduce the Theoretical Background. We will discuss clustering, optimization and the infamous fuzzy c -means algorithm in this chapter. Later in Chapter 3, we will review the relevant literature on fuzzy clustering, in particular, fuzzy c -means, various problems associated with fuzzy c -means and solution proposals available in the literature will be presented. We will also review the literature on clustering with balance constraint, in particular, zoning and districting. In Chapter 4, we will introduce the mathematical formulation of fuzzy clustering with balance constraint and propose two solution algorithms, namely **FCBC** without *Repulsion* and with *Repulsion*. In Chapter 5, a application area of the proposed algorithms in the context of Wireless Sensor Networks will be presented where selecting *Cluster Heads (CHs)* is based on the the centrality of the sensor and its residual energy. Experimental analysis

and comparison of the proposed algorithm with a popular algorithm, *LEACH* will be provided in chapter 5 as well. The thesis will be concluded with our concluding remarks and various future research problems.

Chapter 2

Theoretical Background

2.1 Clustering

As mentioned earlier, objective of clustering is identification of natural groups in collection of unlabelled data. In the literature abundant number of clustering algorithms are available. These algorithms often result in different clusters. It is hard to decide which set of groups yields a better clustering as it depends to the final aim of the clustering. Therefore, the user who is clustering data must decide the criterion in a way which the result will satisfy his/her needs. Minimizing within distance (i.e., total distance of the cluster members to the centers or each other), or maximizing between distance (i.e., intra-distance, that is to say distance between different clusters) or minimizing within distance and maximizing between distance simultaneously are some examples of popular objectives.

2.1.1 Distance Measure

An important factor in clustering is decision of the distance measure between data points. In mathematics, distance function or metric is a function that defines a distance between elements of a set. If the features of the data points are in the same physical units then it is possible to use *Euclidean* distance metric to cluster similar data instances. In 2-dimensional space, given $a = (x_1, x_2)$ and $b = (x'_1, x'_2)$, following is the equation used to show the Euclidean metric:

$$d = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2}$$

If we consider points as vectors then the equation can be written as:

$$\|d\| = \sqrt{(a - b) \cdot (b - a)}$$

where distance is the *dot product* of the differences of two points.

Another distance measure is *Taxicab* distance or *Manhattan* distance where Euclidean geometry is replaced by a new metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates. Given two vectors, a and b in n -dimensional space, the Taxicab distance equation is as follows:

$$d = \sum_{i=1}^n |a_i - b_i|$$

Generalization of Euclidean and Taxicab distances is called *Minkowski distance* which defined as follows:

$$d = \left(\sum_{i=1}^n |a_i - b_i|^p \right)^{\frac{1}{p}}$$

where a and b are two vectors in n -dimensional space and p is distance order and typically being 1 or 2, where latter results in Euclidean distance and former results Taxicab distance.

Other distance metric such as *Kernel-Induced* distance is also available in the literature (See [30] for the details).

2.1.2 Application Areas of Clustering

Clustering algorithms can be applied in various areas. Some examples are as follows:

- 1- *Marketing*: Finding groups of costumers with the similar behaviour according to the historical data.
- 2- *Biology*: Grouping the plants and animals according to their features.
- 3- *Insurance*: Detecting frauds and abusive behaviour from historical transactions.
- 4- *Earthquake Studies*: Grouping observed earthquake epicenters to identify dangerous zones.
- 5- *WWW*: Clustering web log data to discover groups of similar access patterns.

2.1.3 Hard (i.e., Crisp) vs. Soft (i.e., Fuzzy) Clustering

Clustering algorithms can be classified as *Crisp* and *Fuzzy* algorithms. In crisp algorithms, such as K-means, result of clustering is non-overlapping where a data point belongs to the closest cluster. On the other hand, in fuzzy clustering, a data point can belong to multiple number of clusters with a membership degree. Let us firstly define what do we mean by fuzzy.

Consider two groups of people, namely “Tall” and “Short”. Suppose a person with height of taller than six feet is considered as tall and rest is consider as short. Then number of 1 or 0 can be assigned to each person, demonstrating a person’s belongingness to either short or tall. We call this number as *membership degree* of a person to a specific group. In the crisp case the membership degrees, denoted by $u_{jk} \in \{0, 1\}$, indicates assignment of data point k to cluster j . On the other hand in *fuzzy sets theory* memberships are not limited to 0 or 1 any more. Rather they are a number in the range of $[0, 1]$. Therefore in fuzzy clustering a person is not considered either tall or short but a person can be both short and tall to a degree. For instance a person with height of 5.9 feet belongs to tall set with a membership degree of 0.95 and set of short people with 0.05.

2.1.4 Popular Hard Clustering Algorithms

K-means, which first used by Macqueen [9], is one of the simplest unsupervised learning algorithm to solve the hard clustering problems. The idea is fixing some centroids (say K), one for each cluster. These centroids should be placed in a cunning way because different locations cause different results. Ideally centroids can be located as far as possible from each other. In the next step each data point is assigned to the closest centroid. Once we are done with the initial grouping, we need to recalculate the positions of the centroids of the resulting clusters. Since the new positions of the centroids are different from the previous iteration, a reassignment step is required, which assigns each data point to the closest centroid. This loop continues until there won’t be any change (or negligible, i.e. ϵ) in centroids’ locations. K-means algorithm attempts to minimize

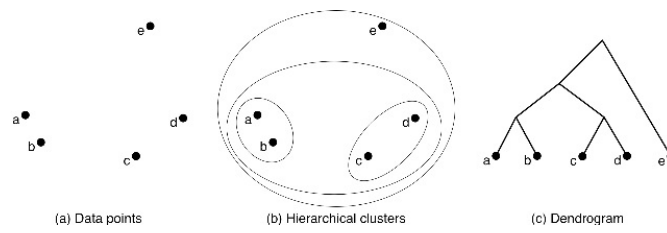


Figure 2.1: Example of Agglomerative Hierarchical Clustering

within distance in objective function:

$$J = \sum_{j=1}^c \sum_{k=1}^n \|x_k - v_j\|^2$$

where $\|\cdot\|$ is a chosen distance measure between a data point, x_k and cluster center v_j (i.e, the centroid of the j^{th} cluster).

Another popular unsupervised clustering is hierarchical clustering which, seeks to build a hierarchy of clusters: Each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy (in the case of agglomerative hierarchical clustering). Hierarchical clustering does not require a *priory* number of clusters. In other words, hierarchical clustering groups data over a variety of scales by creating a cluster tree or dendrogram. The tree is not a single set of clusters, but rather a multilevel hierarchy, where clusters at one level are joined as clusters at the next level. This allows us to decide the level or scale of clustering that is most appropriate for our application.

Figure 2.1 is a simple example of agglomerative hierarchical clustering

2.2 Optimization

2.2.1 Convexity

Convexity is a frequently used concept in the context of optimization, both as convexity in sets and convexity of functions. We will next introduce some basic definitions regarding to convexity, which will be referred to later in this thesis.

Convexity of function of a single variable

A function of a single variable, $f(x)$ is a *convex function* if for any pair of values x', x'' ,

$$f[\lambda x' + (1 - \lambda)x''] \leq \lambda f(x') + (1 - \lambda)f(x'') \quad 0 < \lambda < 1 \quad (2.1)$$

In Equation 2.1, switching to $<$ from \leq makes the function be a *strictly convex* function. Concept of convexity can be interpreted geometrically. As one can see in the Figure 2.2, x' and x'' are two points in the domain of the function and $\lambda f(x') + (1 - \lambda)f(x'')$ is a line that connects x' and x'' , consists of all possible points. In the Figure $f[\lambda x' + (1 - \lambda)x'']$ is located above the function for all intermediate point. If the line is above the function as in the Figure 2.1 then the function is convex. Note that this should be true for any given pair of points in the domain of the function. The other interpretation which indicates convexity of a function $f(x)$ is, bending *upward* if it bends at all.

To be more precise, if $f(x)$ possesses a second derivative everywhere (note that, if it does not, we can not judge about convexity), then $f(x)$ is convex if and only if $d^2 f(x)/dx^2 \geq 0$ for all possible values of x . Again if \leq changes to $<$ we have a *strictly convex* function. On the other hand if $d^2 f(x)/dx^2 \leq 0$, for all x , then $f(x)$ is a concave function. In this case the function should bend downward, if it bends at all. A linear function is

both convex and concave. Also there are some functions which are neither convex nor concave. Figure 2.3 illustrates a function which is neither convex nor concave in the whole domain of the function in the range of $[x_1, x_2]$, however convexity and concavity exists locally in some neighbourhood.

Convexity of a Function of Multiple Variables

We can generalize the discussion on convexity (concavity) of single variable functions, for functions of several variables. Consider a function of multiple variables as $f(x_1, x_2, \dots, x_n)$.

Then $f(x_1, x_2, \dots, x_n)$ is convex **iff**:

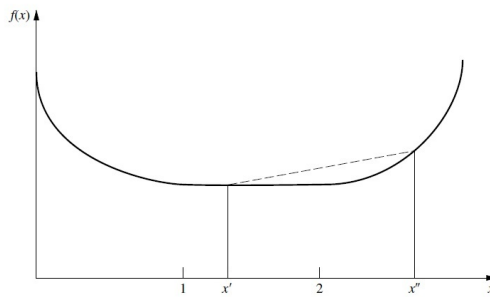


Figure 2.2: A convex function [45]

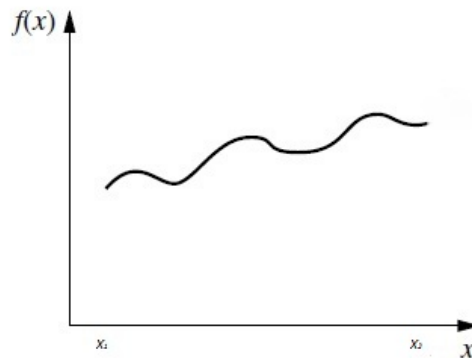


Figure 2.3: A neither convex nor concave function [45]

$$\begin{aligned}
& f(\lambda x'_1 + (1 - \lambda)x''_1, \lambda x'_2 + (1 - \lambda)x''_2, \dots, \lambda x'_n + (1 - \lambda)x''_n) \leq \\
& \lambda f(x'_1, x'_2, \dots, x'_n) + (1 - \lambda)f(x''_1, x''_2, \dots, x''_n) \qquad 0 < \lambda < 1 \quad (2.2)
\end{aligned}$$

Geometrically speaking, if the line segment at the right hand side of the Equation 2.2 lies entirely above the function, the function is convex. The same definitions apply to strictly convex and a concave function as discussed earlier.

Definition *Hessian Matrix* or Hessian is a square matrix of second-order partial derivatives of a function.

Given $f(x_1, x_2, \dots, x_n)$ if all second partial derivatives of f exist and are continuous over the domain of the function, then the Hessian matrix of f is:

$$H(f) = \begin{bmatrix} \frac{\delta^2 f}{\delta x_1^2} & \frac{\delta^2 f}{\delta x_1 \delta x_2} & \cdots & \frac{\delta^2 f}{\delta x_1 \delta x_n} \\ \frac{\delta^2 f}{\delta x_2 \delta x_1} & \frac{\delta^2 f}{\delta x_2^2} & \cdots & \frac{\delta^2 f}{\delta x_2 \delta x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta^2 f}{\delta x_n \delta x_1} & \frac{\delta^2 f}{\delta x_n \delta x_2} & \cdots & \frac{\delta^2 f}{\delta x_n^2} \end{bmatrix}$$

If $H(f)$ be *positive semi-definite* then $f(x)$ is *locally convex*. Note that a real matrix $H(f)$ is positive semi-definite if zHz^T be non-negative for all non-zero column vector z of n real numbers.

Mathematically, $f(x_1, x_2, \dots, x_n)$ is **convex if and only if** its $n * n$ **Hessian** matrix be *positive semi-definite*. If Hessian matrix is positive definite then the function is strictly convex.

Now we can obtain two important results:

1. If $f(x_1, x_2, \dots, x_n)$ is convex then $g(x_1, x_2, \dots, x_n) = -f(x_1, x_2, \dots, x_n)$ is concave.

2. Sum of some convex functions is convex. [45]

Convex Sets

There are two definitions for a convex set. Firstly, collection of the points lie above or on a convex function form a convex set. Similarly, collection of the points lie below or on the concave function form a concave set.

The other way of defining a convex set is a formal mathematical definition. Suppose that two points belong to a set and then if their strict convex combination also belongs to the same set, this set is convex.

$$x_1 \in S, x_2 \in S \Rightarrow \lambda x_1 + (1 - \lambda)x_2 \in S \quad \lambda \in (0, 1)$$

One important concept associated with convex sets in the context of optimization is Extreme Points;

Definition Extreme point of a convex set is a point that does not lie on any segment line. In other words, it can not be represented as strict convex combination of two other points in the same set or

We can also define extreme point as a point c where $f(c) \leq f(x)$ or $f(c) \geq f(x)$ (c is absolute minimum or absolute maximum respectively). A point is local extreme point if it is locally minimum or maximum (in an interval in the domain of the function).

Another concept in this discussion is *critical point*, which is a point c in domain of the function, where $f'(c) = 0$ or $f'(c)$ does not exist. We can conclude that every extreme point is a critical point but every critical point is not local extreme point. For example saddle points are critical points but not extreme points.

Note that point is called *saddle point* if function is convex (concave) immediately before

that point and concave (convex) after that point.

2.2.2 Optimization

In this section we will be reviewing some classical methods of optimization for unconstrained and constrained functions.

Unconstrained optimization of a function of single variable

Suppose that $\frac{df(x)}{dx} = 0$ at $x = x^*$, that is to say x is a critical point, and either locally minimum or maximum. Now this question shows up, is x a global optima? Firstly let us clarify what do we mean by global optima. If a function is greater or equal than a specific point in the domain of function, then that point is global minima. On the other hand, if a function is less than or equal to a point in the domain of function, then that point is global maxima. In order to find out whether this point is globally minimum or not in the domain of the function, we need to check if the function is convex or not. If function is a convex and derivation is zero, then x is a global minimum. Otherwise if function is concave, then x is a global maximum. on the other hand if the function is neither convex nor concave we can't be sure about globality. Figure 2.4 demonstrates these concepts. Note that inflection point (i.e., saddle point) is another extreme point which is neither locally minimum nor maximum.

Constrained optimization with equality constraints

Consider the problem of minimization (or maximization) of $f(x) = f(x_1, x_2, \dots, x_n)$ subject to restrictions that x has to satisfy all equations bellow:

$$g_1(x) = b_1$$

$$g_2(x) = b_2$$

$$\vdots$$

$$g_m(x) = b_m$$

where $m < n$. A classical method of dealing with this problem is *Lagrange multipliers*.

This procedure begins by formulating the Lagrangian function as follow:

$$h(x, \lambda) = f(x) - \sum_{i=1}^m \lambda [g_i(x) - b_i]$$

where $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ are called *Lagrangian multipliers*.

$(x, \lambda) = (x^*, \lambda^*)$ is a local (global if the function is convex) optima for minimization of the unconstrained function $h(x, \lambda)$. According to this approach, to obtain the λ and x derivatives with respect to x_j and λ_i must be set to zero, which results in the following equations:

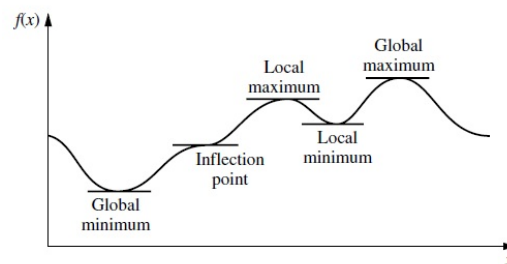


Figure 2.4: Global and Local Optima [45]

$$\frac{\delta h}{\delta x_j} = \frac{\delta f}{\delta x_j} - \sum_{i=1}^m \lambda_j \frac{\delta g_i}{\delta x_j} \quad \forall j = 1, 2, \dots, n$$

$$\frac{\delta h}{\delta \lambda_i} = -g_i(x) + b_i = 0 \quad \forall i = 1, 2, \dots, m$$

As one can observe, the last m equations satisfy feasibility of the solutions. From a practical computational point of view, Lagrange multipliers are not a particularly powerful procedure. It is often not possible to solve the equations to obtain the critical points. Furthermore, even when the points can be obtained, the number of critical points may be so large (often infinite) that it is impractical to attempt to identify a global minimum or maximum. However, for certain types of small problems (where number of decision variables and constraints that are being relaxed is not big [45]), this method can be used successfully.

Constrained optimization with equality constraints and inequality constraints

Consider the problem of minimization or maximization of $f(x)$ subject to restrictions that x has to satisfy all equations below:

$$g_1(x) = b_1$$

$$g_2(x) = b_2$$

$$\vdots$$

$$g_m(x) = b_m$$

$$k_1(x) \leq d_1$$

$$k_2(x) \leq d_2$$

$$\vdots$$

$$k_p(x) \leq d_p$$

$$h(x, \lambda, \gamma) = f(x) - \sum_{i=1}^m \lambda_i [g_i(x) - b_i] + \sum_{j=1}^p \gamma_j [k_j(x) - d_j]$$

where $h(x, \lambda, \gamma)$ is unconstrained problem of the original problem.

The *Karush-Kuhn-Tucker* conditions which will be discussed next, give us candidate optimal solutions x^* . That is, given x^* with λ^* and γ^* satisfying the KKT conditions, and the following ensure solution is optimal ([45]):

If $f(x)$ is a convex function (which can be checked from its hessian matrix), and the feasible region forms a convex set. Note that in order to have a convex feasible region, $g_i(x)$ must be linear and $k_j(x)$ must be convex.

Karush-Kuhn-Tucker Conditions for Constrained Programming

Assume that $f(x), g_1(x), g_2(x), \dots, g_m(x)$ are equality and inequality constraints and are differentiable functions, the $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ can be an optimal solution for

the nonlinear programming problem only if there exist $\lambda_1, \lambda_2, \dots, \lambda_m$ such that all the following KKT conditions are satisfied:

$$\frac{\delta f}{\delta x_j} - \sum_{i=1}^m \lambda_i \frac{\delta g_i}{\delta x_j} \leq 0 \quad x = x^*, \forall j = 1, 2, \dots, n \quad (2.3)$$

$$x_j^* \left(\frac{\delta f}{\delta x_j} - \sum_{i=1}^m \lambda_i \frac{\delta g_i}{\delta x_j} \right) = 0 \quad x = x^*, \forall j = 1, 2, \dots, n \quad (2.4)$$

$$g_i(x^*) - b_i \leq 0 \quad \forall i = 1, 2, \dots, m \quad (2.5)$$

$$\lambda_i [g_i(x^*) - b_i] = 0 \quad \forall i = 1, 2, \dots, m \quad (2.6)$$

$$x_j^* \geq 0 \quad \forall j = 1, 2, \dots, n \quad (2.7)$$

$$\lambda_i \geq 0 \quad \forall i = 1, 2, \dots, m \quad (2.8)$$

Note that both 2.4 and 2.6 conditions require that the product of two quantities be zero.

That is to say, at least one of these two quantities need to be zero. Also, condition 2.6

can be combined with 2.5 to express in another form as, $g_i(x^*) - b_i = 0$ or equivalently

$g_i(x^*) - b_i = 0$ when $\lambda_i = 0$ (slack variables are zero). Similarly, condition 2.3 and 2.4

can form, $\frac{\delta f}{\delta x_j} - \sum_{i=1}^m \lambda_i \frac{\delta g_i}{\delta x_j} = 0$ and when $x_j = 0$ equality changes to \leq . In conditions

2.4, 2.5, 2.7, and 2.8, the λ_i correspond to the dual variables of linear programming,

and they have a comparable economic interpretation (shadow prices). If right hand side

b_i of constraint i is increased by Δ then the optimal objective value increases by $\lambda_i^* \Delta$.

However, the λ_i actually arose in the mathematical derivation as lagrange multipliers.

Conditions 2.5 and 2.7 keep the feasibility of the solution and the other conditions elim-

inate most of the feasible solutions as possible candidates for an optimal solution.

Note that satisfying these conditions does not guarantee that the solution is optimal.

There are also other assumptions which are needed to guarantee a point is optimal.

These assumptions can be extracted from the following theorem. Note that KKT con-

ditions work where both equality and inequality constraints exist, while lagrangean relaxation approach works when there exist just equality constraints.

Theorem Assume that $f(x)$ is a *convex function* and that $g_1(x), g_2(x), \dots, g_m(x)$ are *convex functions* (i.e., this problem is a convex programming problem). Then $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ is an optimal solution if and only if all the KKT conditions are satisfied. [45]

For complicated problems, it may be difficult, if not impossible, to derive an optimal solution directly from the KKT conditions. Nevertheless, these conditions still provide valuable clues as to the identity of an optimal solution, and they also permit us to check whether a proposed solution is optimal or not.

In a nutshell, these conditions become **sufficient** if the function is convex, and in this case, the solution point is globally minimum in the domain of the problem.

Also it should be mentioned that a convergent algorithm does not necessarily stop at a local minimum point. Also, a point satisfying KKT conditions is not necessarily a local minimum point. To make sure a point is minimum (either local or global), $f(x)$ and $g(x)$ must be convex.

Now we are able to define necessary and sufficient conditions for a point to be optimal. Table 2.1 and 2.2, contains these conditions for the different minimization and maximization problems respectively.

Table 2.1: Optimality conditions for minimization problem [45]

Problem	Necessary Conditions for Optimality	Also Sufficient if:
One-variable unconstrained	$\frac{df}{dx} = 0$	$f(x)$ convex
Multivariable unconstrained	$\frac{\delta}{\delta x_j}$ for all j	$f(x)$ convex
Constrained, non-negativity constraint only	$\frac{\delta}{\delta x_j}$ for all j	$f(x)$ convex
General constrained problem	Karush-Kuhn-Tucker conditions	$f(x)$ and $g_i(x)$ convex

Table 2.2: Optimality conditions for maximization problem [45]

Problem	Necessary Conditions for Optimality	Also Sufficient if:
One-variable unconstrained	$\frac{df}{dx} = 0$	$f(x)$ concave
Multivariable unconstrained	$\frac{\delta}{\delta x_j}$ for all j	$f(x)$ concave
Constrained, non-negativity constraint only	$\frac{\delta}{\delta x_j}$ for all j	$f(x)$ concave
General constrained problem	Karush-Kuhn-Tucker conditions	$f(x)$ concave and $g_i(x)$ convex

2.3 Fuzzy C-means Clustering

As discussed in Chapter 2, fuzzy clustering is an unsupervised learning algorithm, where a data point can be assigned to multiple clusters with a membership degree. General definition of fuzzy logic is used to define memberships. Fuzzy C-Means clustering (FCM) is constraint optimization where KKT conditions are applied. But as the objective function (we will provide the mathematical model next) is not convex, based on the table 2.1, applying KKT conditions do not result in global optima but nearly to local optima. Given n data points, FCM determine membership degrees u_{jk} of the k^{th} data point to the j^{th} cluster center, where there are c clusters. Objective is the minimization of the total between distances of the data points to the center of the clusters where they are assigned to multiply by their m^{th} power of their membership degrees are used as the weights (m is known as the fuzzy exponent or fuzzification index or degree of fuzziness) subject to summation of the memberships over all cluster centers for each data point

summing up to one. Also membership degrees can be in the range of $[0, 1]$. Below you can find the mathematical model of the fuzzy clustering problem discussed by Bezdek .

$$\text{Min} \sum_{k=1}^n \sum_{j=1}^c u_{jk}^m d_{jk}^2 \quad (2.9)$$

subject to:

$$\sum_{j=1}^c u_{jk} = 1 \quad \forall k \in \{1, \dots, n\}, \quad (2.10)$$

$$0 \leq u_{jk} \leq 1 \quad \forall k \in \{1, \dots, n\}, \forall j \in \{1, \dots, c\} \quad (2.11)$$

where u_{jk} is membership degree of data point k^{th} to cluster j^{th} . d_{jk}^2 is Euclidean distance between the k^{th} data point and the j^{th} cluster center ($d_{jk}^2 = \|x_k - v_j\|^2$). Also, as discussed in Chapter 3, m is degree of fuzziness. The above formulation can be rewritten with substitution if $u_{jk}^m = u_{jk}$ as suggested by Stefano Rovetta in [44] in order to ease the steps of the proof. Alternative formulation is as follows:

$$\text{Min} \sum_{k=1}^n \sum_{j=1}^c u_{jk} d_{jk}^2 \quad (2.12)$$

subject to:

$$\sum_{j=1}^c u_{jk}^{\frac{1}{m}} = 1 \quad \forall k \in \{1, \dots, n\}, \quad (2.13)$$

$$0 \leq u_{jk}^{\frac{1}{m}} \leq 1 \quad \forall k \in \{1, \dots, n\}, \forall j \in \{1, \dots, c\} \quad (2.14)$$

Constraint 2.10 guarantees the sum of membership degrees of a data point over all clusters add up to one. Constraint 2.11 ensures that $u_{jk} \in [0, 1]$.

2.3.1 FCM- Necessary and Sufficient conditions

Bezdek [7] derived the the necessary and sufficiently conditions for the minimum objective function of the FCM.

Suppose that L be the Lagrangian of the standard fuzzy c-means problem:

$$L = \sum_{k=1}^n \sum_{j=1}^c u_{jk} d_{jk}^2 + \sum_{k=1}^n \lambda_k \left(\sum_{j=1}^c u_{jk}^{\frac{1}{m}} - 1 \right)$$

The solution $\Delta L = 0$ minimizes the standards formulation of the FCM. From the condition $\frac{\delta L}{\delta v_j} = 0$ we have:

$$\frac{\delta L}{\delta v_j} = 2u_{jk}(v_j - x_k) = 2v_j \sum_{k=1}^n u_{jk} - 2 \sum_{j=1}^c u_{jk} x_k = 0$$

Which results in:

$$v_j = \frac{\sum_{k=1}^n u_{jk} x_k}{\sum_{k=1}^n u_{jk}} \quad (2.15)$$

Please note that the cluster centers obtained by Equation 2.12 are similar to the resulting cluster centers of Hard C-Means clustering.

From the condition $\frac{\delta L}{\delta u_{jk}} = 0$ we have:

$$\begin{aligned} \frac{\delta L}{\delta u_{jk}} &= d_{jk} + \lambda_k \frac{1}{m} u_{jk}^{\frac{1-m}{m}} \\ u_{jk} &= \left(\frac{\lambda_k}{m d_{jk}} \right)^{\frac{m}{1-m}} \end{aligned}$$

Now from condition $\frac{\delta L}{\delta \lambda_k} = 0$ we have:

$$\sum_{j=1}^c u_{jk}^{\frac{1}{m}} = \sum_{j=1}^c \left(\frac{\lambda_k}{m d_{jk}} \right)^{\frac{1}{m-1}} = 1$$

Therefore:

$$\frac{\lambda_k}{m} = \left[\sum_{j=1}^c \left(\frac{1}{d_{jk}} \right)^{\frac{1}{m-1}} \right]^{1-m} \quad (2.16)$$

By substituting last term in u_{jk} he ended up with the following equations as cluster center and membership degrees.

$$v_j = \frac{\sum_{k=1}^n u_{jk}^m x_k}{\sum_{k=1}^n u_{jk}^m} \quad (2.17)$$

$$u_{jk} = \frac{1}{\sum_{l=1}^c \left(\frac{d_{lk}}{d_{jk}} \right)^{\frac{2}{m-1}}} \quad (2.18)$$

Starting with an initialization in the cluster centers and solving the the equation (2.17) and (2.18) iteratively we can obtain a good but not necessarily the best solution. Based on [10] and [11] the solution obtained from the algorithm is not global optimal as objective function of the relaxed problem is **not convex**. In [11] it is shown that the result is *strict local minimum*.

Objective function (2.9) is minimization of the square of distance of the data points to their associated cluster centers multiply by the associated membership degrees. Minimizing distance squares leads to minimizing the errors (i.e., assigning data points to the best and closest cluster). If we used distance in the objective function, we would minimize the *total distance* (consider a case, where there is a salesman who has to start from a data point and stop by all data points once in a cluster), which is not desired. In addition, using distance in objective function, causes the function not to possess a second derivative, and as discussed in section 2.2, if a function does not possess second derivative, we are not able to judge about the convexity of the function and this means we can not keep track of optimality of the result.

The other question is why in the objective function membership is being to the power

m . As objective function is minimization of the distance error, ideal membership shall be calculated from an equation which relates membership to the inverse of distances. Now suppose that there is a point which has distances to three cluster centers as 1, 7 and 12. As the closest cluster center is 1 unit far from the point and the farthest one is 12 units far from that point, we are expecting memberships to be associated with these distances. However equation 2.18, gives membership as 0.5, 0.3 and 0.2 respectively. To mitigate this affect, m shows up. As $m \geq 1$ so affect of distance on the membership degree can be more by increasing the m . Now let us consider sensitivity of m . As we mentioned, $m \geq 1$. If $m = 1$, then we shift to *hard* or *non-fuzzy* clustering from fuzzy clustering. That is to say, u_{jk} is a binary variable which is 1 if data point k is assigned to cluster j and 0 otherwise. Now assume that $m = \infty$. Then any given data point has an equal membership degree to each cluster center, means $u_{jk} = \frac{1}{c}$.

In the following sections we will be discussing the convergence and optimality (if there is) of FCM.

Chapter 3

Literature Review

In this thesis we will focus on fuzzy clustering problems with balance constraint and propose a solution methodology. In this chapter we will review the relevant literature in detail. We will first start with fuzzy clustering and later extend the discussion to clustering with constraints. We will discuss various constraints that are introduced as extensions to fuzzy clustering in order to overcome certain problems associated with FCM. Next we will present the literature on clustering with balanced constraint. In this context we will discuss particularly zoning and districting problems as they are closely related to balance constraint clustering problems and are well studied in operations research.

3.1 FCM and Some Extensions

The most famous fuzzy clustering algorithm is *Fuzzy C-Means* algorithm [8]. Bill Wee is the first researcher who worked on fuzzy pattern recognition in his Ph.D. thesis. Ruspini [21] discusses the first fuzzy clustering. Woodbury and Clive [22] combined fuzziness

and probability in hybrid fuzzy clustering and at the same time, Dunn [41] and Bezdek [11] published the work on FCM model.

In the classical FCM clustering problem, given a set of data points (n data points) and their associated coordinates, one is interested in grouping them in a specific number of clusters (c clusters). Objective is maximizing the compactness of the clusters which leads us to minimize the total within distances of the data points. This objective function is considered as *least squares model* based on [7].

The objective function can be written mathematically as $\sum_{k=1}^n \sum_{j=1}^c \bar{u}_{jk}^m d_{jk}^2$ where d_{jk}^2 is simply the square Euclidean distance between the data point k and centroid of the j^{th} cluster. A *degree of fuzziness* (m) is also included in the objective function that controls the level of the cluster fuzziness. A small m (close to one) leads the clustering to a *non-fuzzy*, clustering, i.e., crispy clustering, while a high degree of fuzziness forces all membership to be equal to $\frac{1}{c}$ (total or extreme fuzziness). Optimal degree of fuzziness is an important research area. More interested readers can refer to [19], [20], [23] and [24] for more details.

Although in the literature $m = 2$ is the mostly used value as degree of fuzziness, there is no guarantee that this value is optimal or efficient for all data sets. For example Veit Schwammle and Jensen in [29] discuss about the degree of fuzziness. In this study the data set which is used for the experimental studies, is a gene data set with $N \times D$ attributes (N: Number of objects to be clustered and D: number of dimensions of an object e.g. 3 for 3-dimensional data set). In this study rather than a prespecified (i.e, a priori) m determining it as a function of N and D is proposed. The following function provides a good fit of the curves for all combinations of N and D:

$$m(D, N) = 1 + \left(\frac{1418}{N} + 22.05 \right) D^{-2} + \left(\frac{12.33}{N} + 0.243 \right) D^{-0.04061 \ln(N) - 0.1134}$$

Although FCM is the most popular clustering algorithm in literature, there are some drawbacks to this algorithm. In order to overcome to these problems, there are several extension to FCM is suggested in the literature.

Possibilistic C-Means (PCM) is another popular method of fuzzy clustering introduced by Krishnapuran and Keller [25]. PCM relax the requirement of memberships to all clusters for each data point summing up to one, which is enforced in FCM ([16], [25]). The mentioned constraint creates various problems in generating the memberships. For example in case a data point is far but equidistant from the two cluster centers (noisy data points), this constraint forces membership degrees to be equal and in some cases as high as 0.5, however assigning a low or even zero membership degrees to such noisy data points in some applications makes more sense.

Pal *et al.* in [16] describe *typicality* as possibility that a data point belongs to a cluster, a value between zero and one, similar to the membership degree, with a difference that it does not necessarily sum up to 1 over any column of typicality matrix (for a specific data point to all clusters). They proposed a possibilistic-fuzzy c-means algorithm in which both membership degree and typicality are included in the objective function. Necessary conditions are determined and the proposed algorithm is tested with several data sets. It turns out that the new algorithm can mitigate the effect of noisy data points.

PCM itself has problem which is argued by Barni *et.al* in [13]. PCM exhibits an undesirable tendency to converge to coincidental cluster. Consider a case where data points form one cluster, but $c = 2$. PCM does recognise two coincidental clusters rather than spiliting data points to two clusters. Krishnapuram and Keller in [25] describe the basic differences of PCM and FCM. According to the authors PCM's strength is it's high robustness in the presence of noisy data. On the other hand its weakness is that it

requires a good initialization. Also it needs an appropriate degree of fuzziness. When the data is not severely contaminated, the FCM can provide a reasonable initialization and then PCM can be used to improve the results of the FCM.

One of other extensions of FCM is proposed in [30] by S.R. Kannan *et al.* where a robust non-euclidean distance measure for the original data space to derive new objective function is introduced. FCM adopts a new kernel-induced distance in the data space to replace the original euclidean distance. Also a regularized entropy term is added to the objective function ($E(u) = - \sum_{j=1}^c \sum_{k=1}^n u_{jk} \log u_{jk}$). In information theory, the Shannon entropy is a measure of the uncertainty associated with the random variable. The entropy term attains its maximum value when all memberships are equal $\frac{1}{c}$. Therefore entropy term is included in clustering algorithms to get the additional information of the data which provides better partition in the clustering result.

There are more researches that considered the entropy in the objective function. Beni and Liu [36] consider the entropy as well. Moreover in order to minimize the bias of centers towards any data point they added $\sum_{k=1}^n (\vec{x}_k - \vec{c}_j) u_{jk} = 0 \forall j$ as a constraint which locates the centers at the middle of the data points in the cluster.

Izakian *et.al* in [37] use a weighted distance rather than regular distance. Given set of data points, whose features are coming from p data sources, each data source describes data point from its point of view. Having $x_k = [x_k(1)|x_k(2)|\dots|x_k(p)]$ distance can be described as:

$$d_{\lambda_1, \dots, p}^2 = \lambda_1 \|v_j(1) - x_k(1)\|^2 + \lambda_2 \|v_j(2) - x_k(2)\|^2 + \dots + \lambda_p \|v_j(p) - x_k(p)\|^2$$

where λ_j denotes the contribution of a data source on the clustering process and $\sum_{j=1}^p \lambda_j = 1, 0 \leq \lambda_j \leq 1$ and $\|\cdot\|$ denotes euclidean distance.

Another issue regarding to FCM which has to be taken into consideration is local solu-

tions with empty clusters or clusters having very few points. It usually happens when either dimension of data or desire number of cluster is big. Bradley *et.al* in [14] add k constraints to the underlying clustering optimization problem requiring that each cluster have at least a minimum number of points in it.

3.2 Clustering with Balance Constraint; Zoning/Districting Problems

Districting or *Zoning* is a relevant problem to clustering with balance constraint. Zoning usually refers to geographical design of an area. Various applications such as distribution (collection) supply chain subsystems, political districting, public services (e.g., police, health services) districting and sales territories are available. Various objectives such as spatial layout (i.e., connectivity of zones), equality (in terms of population or any specific criterion), interaction (important in transport modelling) and proximity (i.e., compactness in terms of a defined distance measure) [26] are considered in zoning. These objectives (or constraints) differentiate zoning problems from traditional location models. In the literature mostly a crisp assignment is being adopted in such problems rather than fuzzy assignments. On the other hand, balancing the zones reduces the applicability of fuzzy clustering techniques available in the literature to this problem.

For example in the distribution problem all zones have equal workload. By adding a *balance constraint* in to the classical fuzzy clustering it is possible to obtain a result similar to zoning. Hojjati in [17] deals with a political districting problem. Political districting problem is dividing a given area into the c districts subject to each district has almost the same population of voters with a given tolerance and is compact and

has the minimum number of split population units. He utilizes a Lagrangean relaxation approach as a solution to the problem. *Warehouse Location Model* in which objective function is minimizing the square euclidean distance from centre of a population unit to the centre of population unit is developed. Hojjati [17] applied the developed algorithm in the city of the Saskatoon, which had to be partitioned to 11 provincial constituencies (districts).

Salazar-Aguilar *et al* in [14] face with a real life problem arose in Mexico in a bottled beverage distribution company. Given customers information, they are interested in clustering them in such a way that number of customers in clusters be equal (balanced). Pavone *et al.* in [48] address another application of zoning. They deal with the problem of dividing a region to a specified number of sub region, and then assigning a responsible employee to each sub region such that the work load for each responsible employee is equal.

Such problems can be found in the literature. However in some problems not balancing but grouping the objects (e.g. customers) is the main purpose. Baron *et al.* in [46] model the problem of locating c facilities on the unit square to minimize the maximal demand faced by each facility such that assignment to the closest facility and coverage constraints are satisfied. Consider locating cellular phone towers (e.g. facilities) in a given region. Towers are considered identical. As tower's capacity is correlated to the demand it satisfies, demand dictates cost. Hence to minimize cost, minimizing the maximal demand is targeted. Nikolakopoulou *et al.* in [43] consider a problem where *Routing* is an important issue. Therefore objective is minimizing the total travel distance by vehicles which indicate roust. In this study also balancing workload is considered.

Chapter 4

Fuzzy Clustering with Balance

Constraint

4.1 Problem Statement

Fuzzy clustering with Balance constraint is an extension version of fuzzy clustering problem which aims to determine fuzzy clusters that are compacted, in which an *Equality Constraint* is added to the classical model. Therefore mathematical model is:

$$\text{Min} \sum_{k=1}^n \sum_{j=1}^c u_{jk}^m d_{jk}^2 \quad (4.1)$$

subject to:

$$\sum_{j=1}^c u_{jk} = 1 \quad \forall k \in \{1, \dots, n\}, \quad (4.2)$$

$$\sum_{k=1}^n p_k u_{jk} = \frac{p}{c} \quad \forall j \in \{1, \dots, c\}, \quad (4.3)$$

$$0 \leq u_{jk} \leq 1 \quad \forall k \in \{1, \dots, n\}, \forall j \in \{1, \dots, c\} \quad (4.4)$$

where p_k is population of each data point and p is total population. Objective 4.1 minimizes total distance square errors to the cluster centers. Constraint 4.2 forces total membership of each data point to all cluster centers sum up to one. Note that in the context of zoning applications summation of membership degrees to 1 is desired for practical purposes. Constraint 4.3, which is balance constraint, guarantees that total population with respect to membership degrees within clusters are equal. Objective function 4.1 is nonlinear as both u_{jk} and d_{jk} are decision variables. In addition based on [11] objective function is not convex. Both sets of constraints (4.2 and 4.3) are linear as p_k is parameter.

In order to solve this problem, there are some possible approaches. We can either solve this problem, with as exact model, or approximation or by utilizing heuristics or meta heuristic methods exist in the literature. We adopt a Lagrangean relaxation approach to deal with this problem. Recall that KKT conditions are a generalization of lagrangean relaxation approach where there exist inequality constraints. Now to solve this non-linear program with linear constraints, we shall use Lagrangian relaxation. As in the relaxed problem, non-negativity of the u_{jk} ensures the feasibility of the solution, Lagrangian relaxation works as KKT condition as we satisfy the feasibility of the solution. Also we use the same conversion Filippone *et al.* did in [56] and replace u_{jk}^m with u_{jk} . The relaxed problem is as follows:

$$MinL = \sum_{k=1}^n \sum_{j=1}^c u_{jk} d_{jk}^2 + \sum_{k=1}^n \lambda_k (1 - \sum_{j=1}^c u_{jk}^{\frac{1}{c}}) + \sum_{j=1}^c \gamma_j (\frac{p}{c} - \sum_{k=1}^n p_k u_{jk}^{\frac{1}{c}}) \quad (4.5)$$

subject to:

$$0 \leq u_{jk} \leq 1 \quad \forall k \in \{1, \dots, n\}, \forall j \in \{1, \dots, c\} \quad (4.6)$$

We assumed $m = 2$ as most of literature allow it. In order to obtain KKT condition we need to take derivative with respect to all variables.

$$\frac{\partial L}{\partial u_{jk}} = \frac{\partial \sum_{k=1}^n \sum_{j=1}^c u_{jk} d_{jk}^2 + \sum_{k=1}^n \lambda_k (1 - \sum_{j=1}^c u_{jk}^{\frac{1}{m}}) + \sum_{j=1}^c \gamma_j (\frac{p}{c} - \sum_{k=1}^n p_k u_{jk}^{\frac{1}{m}})}{\partial u_{jk}} \quad (4.7)$$

$$\begin{aligned} &= \frac{\partial \sum_{k=1}^n \sum_{j=1}^c u_{jk} d_{jk}^2}{\partial u_{jk}} + \frac{\partial \sum_{k=1}^n \lambda_k (1 - \sum_{j=1}^c u_{jk}^{\frac{1}{m}})}{\partial u_{jk}} + \frac{\partial \sum_{j=1}^c \gamma_j (\frac{p}{c} - \sum_{k=1}^n p_k u_{jk}^{\frac{1}{m}})}{\partial u_{jk}} \\ &= d_{jk}^2 - \frac{\lambda_k}{m} u_{jk}^{\frac{1}{m}-1} - p_k \frac{\gamma_j}{m} u_{jk}^{\frac{1}{m}-1} = 0 \end{aligned} \quad (4.8)$$

$$\frac{\partial L}{\partial v_j} = \frac{\partial \sum_{k=1}^n \sum_{j=1}^c u_{jk} d_{jk}^2 + \sum_{k=1}^n \lambda_k (1 - \sum_{j=1}^c u_{jk}^{\frac{1}{m}}) + \sum_{j=1}^c \gamma_j (\frac{p}{c} - \sum_{k=1}^n p_k u_{jk}^{\frac{1}{m}})}{\partial v_j} \quad (4.9)$$

$$\begin{aligned} &= \frac{\partial \sum_{k=1}^n \sum_{j=1}^c u_{jk} d_{jk}^2}{\partial v_j} = \\ &= u_{11}(v_1 - x_1)^2 + \dots + u_{j1}(v_j - x_1)^2 + \dots + u_{c1}(v_c - x_1)^2 \\ &+ u_{12}(v_1 - x_2)^2 + \dots + u_{j2}(v_j - x_2)^2 + \dots + u_{c2}(v_c - x_2)^2 + \dots \\ &+ u_{1k}(v_1 - x_k)^2 + \dots + u_{jk}(v_j - x_k)^2 + \dots + u_{cn}(v_c - x_n)^2 \\ &= 2v_j u_{1j} - 2x_1 u_{1j} + 2v_j u_{2j} - 2x_2 u_{2j} + \dots + 2v_j u_{nj} - 2x_n u_{nj} \end{aligned} \quad (4.10)$$

$$\begin{aligned} &= 2v_j \sum_{k=1}^n u_{jk} - 2 \sum_{k=1}^n x_k u_{jk} = 0 \\ &\Rightarrow v_j = \frac{\sum_{k=1}^n x_k u_{jk}}{\sum_{k=1}^n u_{jk}} \end{aligned} \quad (4.11)$$

$$\begin{aligned} \frac{\partial L}{\partial \lambda_k} &= \frac{\partial \sum_{k=1}^n \sum_{j=1}^c u_{jk} d_{jk}^2 + \sum_{k=1}^n \lambda_k (1 - \sum_{j=1}^c u_{jk}^{\frac{1}{m}}) + \sum_{j=1}^c \gamma_j (\frac{p}{c} - \sum_{k=1}^n p_k u_{jk}^{\frac{1}{m}})}{\partial \lambda_k} \Rightarrow \\ &1 - \sum_{j=1}^c u_{jk}^{\frac{1}{m}} = 0 \Rightarrow \sum_{j=1}^c u_{jk}^{\frac{1}{m}} = 1 \end{aligned} \quad (4.12)$$

$$\begin{aligned} \frac{\partial L}{\partial \gamma_j} &= \frac{\partial \sum_{k=1}^n \sum_{j=1}^c u_{jk} d_{jk}^2 + \sum_{k=1}^n \lambda_k (1 - \sum_{j=1}^c u_{jk}^{\frac{1}{m}}) + \sum_{j=1}^c \gamma_j (\frac{p}{c} - \sum_{k=1}^n p_k u_{jk}^{\frac{1}{m}})}{\partial \gamma_j} \\ &= \sum_{k=1}^n p_k u_{jk}^{\frac{1}{m}} = \frac{p}{c} \end{aligned} \quad (4.13)$$

We can solve the following system of equations:

$$d_{jk}^2 - \frac{\lambda_k}{m} u_{jk}^{\frac{1}{m}-1} - p_k \frac{\gamma_j}{m} u_{jk}^{\frac{1}{m}-1} = 0 \quad (4.14)$$

$$\frac{\sum_{k=1}^n x_k u_{jk}}{\sum_{k=1}^n u_{jk}} - v_j = 0 \quad (4.15)$$

$$\sum_{k=1}^n p_k u_{jk}^{\frac{1}{m}} - \frac{p}{c} = 0 \quad (4.16)$$

$$\sum_{j=1}^c u_{jk}^{\frac{1}{m}} - 1 = 0 \quad (4.17)$$

Now from 4.14

$$d_{jk}^2 - u_{jk}^{\frac{1-m}{m}} \left[\frac{\lambda_k}{m} + \frac{\gamma_j p_k}{m} \right] = 0 \Rightarrow u_{jk}^{\frac{1-m}{m}} = \frac{d_{jk}^2}{\frac{\lambda_k}{m} + \frac{\gamma_j p_k}{m}} \Rightarrow$$

$$u_{jk}^{\frac{1}{m}} = \frac{d_{jk}^{\frac{2}{1-m}}}{\left(\frac{\lambda_k}{m} + \frac{\gamma_j p_k}{m} \right)^{\frac{1}{1-m}}} \quad (4.18)$$

From 4.17 we have:

$$\sum_{j=1}^c \frac{d_{jk}^{\frac{2}{1-m}}}{\left(\frac{\lambda_k}{m} + \frac{\gamma_j p_k}{m} \right)^{\frac{1}{1-m}}} = 1 \quad (4.19)$$

Again from 4.14

$$\left[\frac{d_{jk}^2}{\left(u_{jk}^{\frac{1}{m}} \right)^{1-m}} - \frac{\lambda_k}{m} \right] \frac{m}{p_k} = \gamma_j \Rightarrow$$

$$\gamma_j = \left[\frac{d_{jk}^2 \cdot m}{\left(u_{jk}^{\frac{1}{m}} \right)^{1-m} \cdot p_k} - \frac{\lambda_k}{p_k} \right] \quad (4.20)$$

$$\lambda_k = \left[\frac{d_{jk}^2 \cdot m}{\left(u_{jk}^{\frac{1}{m}} \right)^{1-m} - \gamma_j \cdot p_k} \right] \quad (4.21)$$

From 4.16

$$\sum_{k=1}^n p_k \frac{d_{jk}^{\frac{2}{1-m}}}{\left(\frac{\lambda_k}{m} + \frac{\gamma_j p_k}{m} \right)^{\frac{1}{1-m}}} = \frac{p}{c} \quad (4.22)$$

4.19 can be rewritten explicitly as follows:

$$\begin{aligned}
\frac{d_{11}^{\frac{2}{1-m}}}{\left(\frac{\lambda_1}{m} + \frac{\gamma_1 p_1}{m}\right)^{\frac{1}{1-m}}} + \frac{d_{21}^{\frac{2}{1-m}}}{\left(\frac{\lambda_1}{m} + \frac{\gamma_2 p_1}{m}\right)^{\frac{1}{1-m}}} + \dots + \frac{d_{c1}^{\frac{2}{1-m}}}{\left(\frac{\lambda_1}{m} + \frac{\gamma_c p_1}{m}\right)^{\frac{1}{1-m}}} &= 1 \\
\frac{d_{12}^{\frac{2}{1-m}}}{\left(\frac{\lambda_2}{m} + \frac{\gamma_1 p_2}{m}\right)^{\frac{1}{1-m}}} + \frac{d_{22}^{\frac{2}{1-m}}}{\left(\frac{\lambda_2}{m} + \frac{\gamma_2 p_2}{m}\right)^{\frac{1}{1-m}}} + \dots + \frac{d_{c2}^{\frac{2}{1-m}}}{\left(\frac{\lambda_2}{m} + \frac{\gamma_c p_2}{m}\right)^{\frac{1}{1-m}}} &= 1 \\
\frac{d_{1n}^{\frac{2}{1-m}}}{\left(\frac{\lambda_n}{m} + \frac{\gamma_1 p_n}{m}\right)^{\frac{1}{1-m}}} + \frac{d_{2n}^{\frac{2}{1-m}}}{\left(\frac{\lambda_n}{m} + \frac{\gamma_2 p_n}{m}\right)^{\frac{1}{1-m}}} + \dots + \frac{d_{cn}^{\frac{2}{1-m}}}{\left(\frac{\lambda_n}{m} + \frac{\gamma_c p_n}{m}\right)^{\frac{1}{1-m}}} &= 1
\end{aligned} \tag{4.23}$$

In which there are n equations. 4.22 can be rewritten as follows:

$$\begin{aligned}
p_1 \frac{d_{11}^{\frac{2}{1-m}}}{\left(\frac{\lambda_1}{m} + \frac{\gamma_1 p_1}{m}\right)^{\frac{1}{1-m}}} + p_2 \frac{d_{12}^{\frac{2}{1-m}}}{\left(\frac{\lambda_2}{m} + \frac{\gamma_1 p_2}{m}\right)^{\frac{1}{1-m}}} + \dots + p_n \frac{d_{1n}^{\frac{2}{1-m}}}{\left(\frac{\lambda_n}{m} + \frac{\gamma_1 p_n}{m}\right)^{\frac{1}{1-m}}} &= \frac{p}{c} \\
p_1 \frac{d_{21}^{\frac{2}{1-m}}}{\left(\frac{\lambda_1}{m} + \frac{\gamma_2 p_1}{m}\right)^{\frac{1}{1-m}}} + p_2 \frac{d_{22}^{\frac{2}{1-m}}}{\left(\frac{\lambda_2}{m} + \frac{\gamma_2 p_2}{m}\right)^{\frac{1}{1-m}}} + \dots + p_n \frac{d_{2n}^{\frac{2}{1-m}}}{\left(\frac{\lambda_n}{m} + \frac{\gamma_2 p_n}{m}\right)^{\frac{1}{1-m}}} &= \frac{p}{c} \\
p_1 \frac{d_{c1}^{\frac{2}{1-m}}}{\left(\frac{\lambda_1}{m} + \frac{\gamma_c p_1}{m}\right)^{\frac{1}{1-m}}} + p_2 \frac{d_{c2}^{\frac{2}{1-m}}}{\left(\frac{\lambda_2}{m} + \frac{\gamma_c p_2}{m}\right)^{\frac{1}{1-m}}} + \dots + p_n \frac{d_{cn}^{\frac{2}{1-m}}}{\left(\frac{\lambda_n}{m} + \frac{\gamma_c p_n}{m}\right)^{\frac{1}{1-m}}} &= \frac{p}{c}
\end{aligned} \tag{4.24}$$

In which there are c equations. From latest two sets of equations (4.23,4.24) we have a system of equations including $n + c$ equations and $n + c$ unknowns (Since $m = 2$ these systems are linear). By solving this system we can obtain λ_k and γ_j . And then by substituting these values in the main equation obtained for membership, we are able to find the memberships. From there we can find cluster centres.

Algorithm 1 Proposed algorithm to solve fuzzy clustering with equality constraint

- 1: Define c and ϵ
 - 2: Initialize Cluster Centers
 - 3: **while** $\|u_{jk}^{r+1} - u_{jk}^r\| \leq \epsilon$ **do**
 - 4: Calculate Euclidean Distances
 - 5: Solve the system of equations and determine λ_k and γ_j .
 - 6: Use equation 4.18 and find u_{jk}
 - 7: Use equation 4.15 and find v_j
 - 8: **end while**
-

4.2 Repulsion

In order to avoid very close cluster centers as the result of Algorithm 1 (As it makes objective function be minimum however the results are not favourable) we added a penalty to the objective function as Timm *et al* did in [15]. The term is $\sum_{j=1}^c \sum_{l=1, l \neq j}^c \frac{1}{d_{il}^2}$. As the penalty term in objective function is a function of the cluster center, derivative with respect to v_j is the only equation which is affected. The resulting new equation for cluster center is:

$$v_j = \frac{\sum_{k=1}^n x_k u_{jk} - \eta \sum_{l=1, l \neq j}^c v_l \frac{1}{d_{lj}^2}}{\sum_{k=1}^n u_{jk} - \eta \sum_{l=1, l \neq j}^c \frac{1}{d_{lj}^2}} \quad (4.25)$$

where η is weight of the penalty cost regarding to repulsion.

From the sets of equations (4.23,4.24) we have a system of equations including $n + c$ equations and $n + c$ unknowns. By solving this system we can obtain λ_k and γ_j . And then by substituting these value in the main equation obtained for membership, we are able to find the memberships. From there we can find cluster centres.

Algorithm 2 Proposed algorithm to solve fuzzy clustering with equality constraint and

Repulsion

Define c and ϵ

2: Initialize Cluster Center

while $\|u_{jk}^{r+1} - u_{jk}^r\| \leq \epsilon$ **do**

4: Calculate Euclidean Distances

Solve the system of equations and determine λ_k and γ_j .

6: Use equation 4.18 and find u_{jk}

Use equation 4.25 and find v_j

8: **end while**

4.3 Optimality

In FCM with balanced constraint, we deal with the same objective function we face with in FCM, as it is proven in [37] and [46], this is not a convex function, so any algorithm does not obtain a global optimal solution, while it is possible to obtain a local optima in that specific point. To prove the point we obtained is local optimum, we have to check KKT conditions in that point. And as mentioned before Lagrangean multipliers approach is a special case of KKT conditions. Therefore applying Lagrangean multiplier causes finding local optimum. (Note that in a minimization problem, where objective function is also convex, Lagrangean multiplier leads the solution to be global optimum.)

Hessian matrix of the objective function is as follows:

$$\left[\begin{array}{ccccc|ccccc} 2u_{11}^2 & 0 & 0 & \dots & 0 & 4u_{11}d_{11} & 0 & 0 & \dots & 0 \\ 0 & 2u_{12}^2 & 0 & \dots & 0 & 0 & 4u_{11}d_{12} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 2u_{1n}^2 & 0 & 0 & 0 & \dots & 4u_{cn}d_{cn} \\ \hline 4u_{11}d_{11} & 0 & 0 & \dots & 0 & 2u_{21}^2 & 0 & 0 & \dots & 0 \\ 0 & 4u_{11}d_{12} & 0 & \dots & 0 & 0 & 2u_{22}^2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 4u_{cn}d_{cn} & 0 & 0 & 0 & \dots & 2u_{cn}^2 \end{array} \right]$$

Which is not positive semi-definite (please refer to [11] for the proof) and it is shown hessian matrix associated with this objective function is not positive semi-definite); means objective function is not convex. Therefore the proposed algorithm finds the local optimum.

Chapter 5

Application in Wireless Sensor Networks

Wireless sensor networks (WSNs) have been recognised as an important system in variety of areas in recent years. WSN consists of hundreds of thousands of autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, humidity, light, vibration, etc, equipped with data processing and communication units to pass data to a main location (Base Station). The development of WSNs was motivated by military applications such as battlefield surveillance. Nowadays, WSNs are used in many applications such as environmental monitoring, acoustic detection, seismic detection, inventory tracking, medical monitoring, smart spaces and etc. Advantage of using WSNs is their independence, i.e. they can work without human's interference. In harsh environments where human intervention is risky or infeasible, WSNs can be used as they are extremely small, low cost and need low power (e.g., 1 joule battery).

Since sensor nodes are powered by limited energy source like battery, *energy conservation* is considered to be the most important feature in order to keep the connectivity and

operation of the network and increase the lifetime of the sensor nodes, especially when the implemented field is inaccessible and the battery cannot be replaced or recharged independently; Hence optimization of energy must be taken into consideration.

To this end, grouping sensors to the clusters and assigning a cluster head (CH) to each cluster can save energy as each single sensor connects with the associated CH and after processing data, CH transmits information to Base Station (BS). Since transmitting data to BS by each single sensor is energy costly, clustering helps save energy [31]. There exist a number of cluster-based protocols that have proposed by variant researcher. Among them *low energy adaptive clustering hierarchy* (LEACH), [50] which is a typical cluster-based protocol using a distributed cluster formation algorithm. The CHs are selected with a predefined probability, other nodes select the closest cluster to join, based on the signal strength of the advertisement message they receive from the CHs. The CHs change over time among all the nodes in the network to save energy of the CHs because of high-traffic load in CHs.

Bandyopadhyay and Coyle in [15] use Hierarchical Clustering method to cluster sensors. Ye *et al.* [26] propose a dynamic algorithm which updates cluster heads in each iteration with respect to the residual energy which increases the life time of the system. There are other of protocols which improve the network life time by developing the efficiency of the data transmission however the structure of the clusters is not optimized yet. PEGASIS, TEEN, APTEEN and HEED are examples of such protocols ([51],[52],[53],[54]). Another issue regarding to WSNs is distance of the sensors specially distances of CHs to BS. Since there are always some sensors closer to BS rather than the others and must relay data for a large part of the network, they consume battery energy very quickly. Lou and Hubaux in [3] suggest that BS to be mobile whenever it is possible, therefore close sensors change over time. Having some degrees of overlap among clusters can facilitate

many applications, like inter-cluster routing, topology discovery and node localization, recovery from cluster head failure [5]. As Fuzzy C-Means clustering algorithm always generates clusters in such a way that data points will be assigned not only to one cluster but to all clusters it gives desired results with respect to overlapping. Eventually efficient management of WSNs for extending life time of the system is crucial for system performance. However little attention is paid to the efficiency of energy usage at the CH. Gupta and Younis in [4] address this issue. They claimed if work load of CHs be equal over all clusters, life time of the network extends . This is true, since CHs define the network's life time upper bound (non-functionality of a CH is considered as bottle neck of the network). Intuitively, to form the clusters and assigning a CH to each cluster FCM is used and it is assumed that FCM forms clusters in such a way that the number sensors in clusters be equal. This assumption is true, for small networks. However by increasing network size, not only FCM does not take care of the equality, but also, there can be empty clusters or clusters with small number of sensors. Furthermore even though FCM is used as a way to determine the CH, the available literature defuzzifies at each iteration and assigns each node to a cluster head which reduces the robustness that overlapping clusters might provide to the network. We study clustering of wireless sensor networks where overlapping is a target and simultaneously generates clusters with equal workload for each cluster head. Both objectives lead us towards increasing the life time of the network which cause decreasing cost of the system.

As discussed in chapter 4, we proposed a fuzzy clustering algorithm with balance constraint. Our protocol is similar to the one suggested in [56]. Proposed protocol is as follows:

We firstly initialize energy of the network. Next we use Algorithm 2 to form the cluster. In the next step we shall calculate the centroid of each cluster. Then g closest

sensors will be recognized. The one with highest residual energy will be selected as CH. Then by applying Equation 4.18 to the resulting CHs and sensors we can calculate the membership degrees. As in the proposed algorithm, we do not defuzzify, we use the membership degrees as weights of data packets each sensor sends to CHs. Finally in order to update energy of the network we use the equations provided in [56].

Algorithm 3 Proposed cluster based protocol to solve WSNs problem

Initialize energy of the network

while Number of rounds is less than maximum number of rounds **do**

3: Use Algorithm 2 to form clusters

Find 5 closest sensors to the centroids and choose one with the most residual energy

Update energy of network (both CHs and sensors)

6: **end while**

5.1 Computational Experiments

In order to test the proposed algorithm, we compared the proposed algorithm, with LEACH protocol. In LEACH protocol, each node with a predefined probability can be CH. At the beginning of each round, sensors send an advertisement message to each other. Then based on minimum distance, each sensor sends data packets to a CH and CH after data aggregation, sends data to BS.

The computational study is conducted with data sets generated in MATLAB with different number of data points $n = 100$, $n = 200$, $n = 500$, and their associated coordination. Multiple number of test sets were examined with different characteristics. We consid-

Table 5.1: Experiment set up

Parameters	Value
Network Size	300×300 & 500×500
Number of sensors	100, 200, 500
Number closest sensors to centroid (g)	2, 3, 4, 5
Base Station location	(150, 150) & (250, 250)
Packet Generation rate	1 packet/Sec
Update interval	120 mins
E_{elec}	50 nj/bits
ϵf_s	10 nj/bits/ m^2
Initial Energy	0.5 Jules
Data packet size	500 bytes

ered the area that sensors are distributed is 300×300 , 500×500 and base station is located at (150, 150) and (250, 250). Table 5.1 illustrates all parameters we have used. As mentioned in section 4.2, we have added a penalty cost to the objective function to avoid locating cluster centers too close to each other. To this end, we have considered a constant, η , as penalty cost. To find the best value of η which minimize objective function and simultaneously locating cluster centers far from each other, we have tried different values of η , checked the associated objective function, and selected $\eta = 1700$ in all of our calculations.

We have implemented the proposed algorithm in chapter 5 in MATLAB. We consider total remaining energy of network and total number of alive sensor (with positive residual energy) as two measures for algorithm's performance. We have tested g as well. Recall that g is the number of closest sensors to the centroid. We select CHs from this set of sensors with respect to their residual energy. Figure 5.1 and 5.2 demonstrate

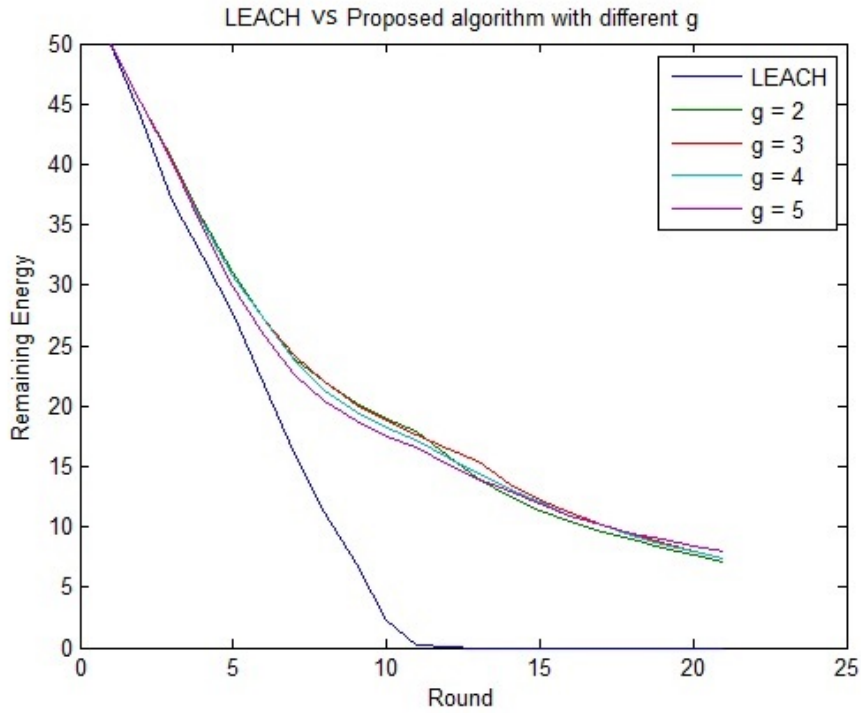


Figure 5.1: Effect of g on the Remaining Energy of Network, $n = 100$

performance of proposed algorithm and LEACH where $n = 100$. Obviously, our algorithm works better than LEACH in both measures. Figure 5.1 and 5.2 also illustrate sensitivity of total remaining energy of network and number of alive sensors to g . We expected the energy consumption decrease by increasing g since increasing g results in selecting a better CH. Figure 5.3 and 5.4 illustrate effect of repulsion. Since repulsion forces the CHs to be far from each other, it decreases distances to CHs therefore energy consumption decreases. Figure 5.5 and 5.6 show affect of g on a network of $n = 200$ sensors. The other parameter which we have examined is number of cluster centers. We expectet the performance of the algorithm gets better when we increase number of clusters but we did not see this from the result. Figure 4.14 represent position of CHs in each round. As you see, after a while when the number of dead sensors is growing up, distribution of the sensors changes, which has effect on the CH selection.

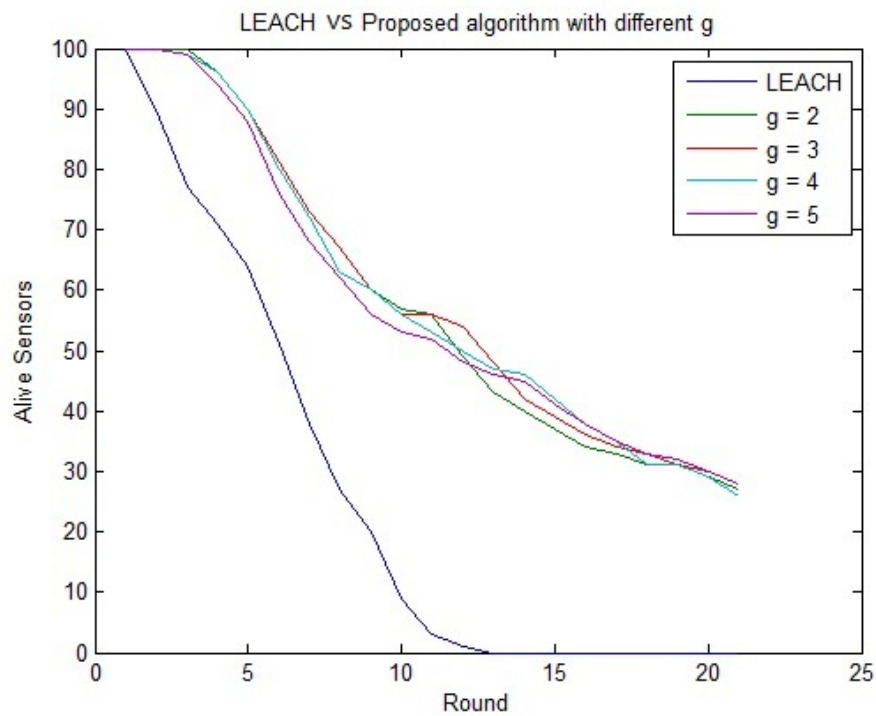
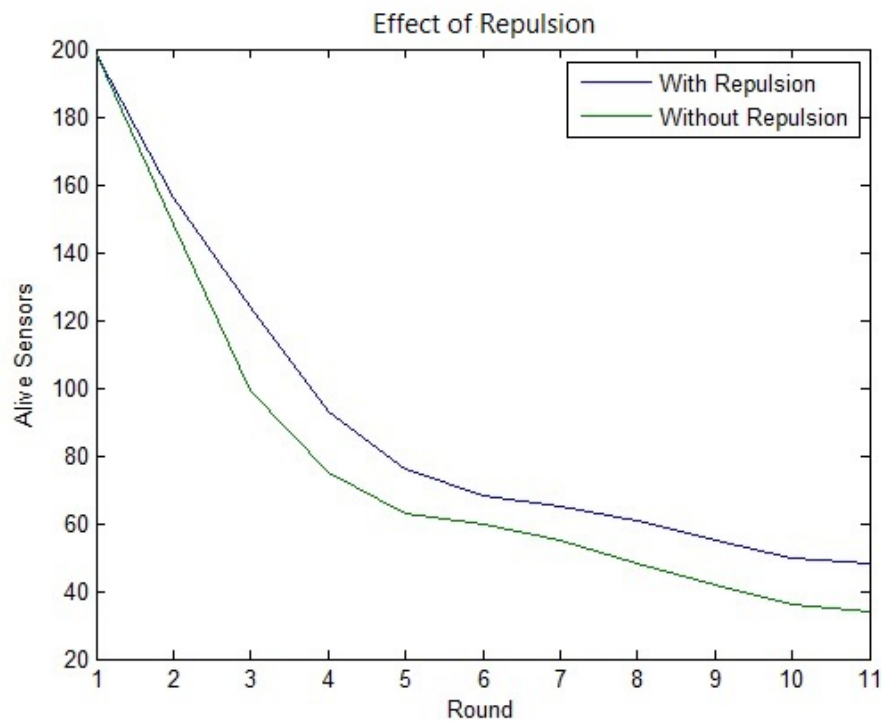
Figure 5.2: Effect of g on the Number of Alive Sensors, $n = 100$ 

Figure 5.3: Effect of Repulsion on the Number of Alive Sensors

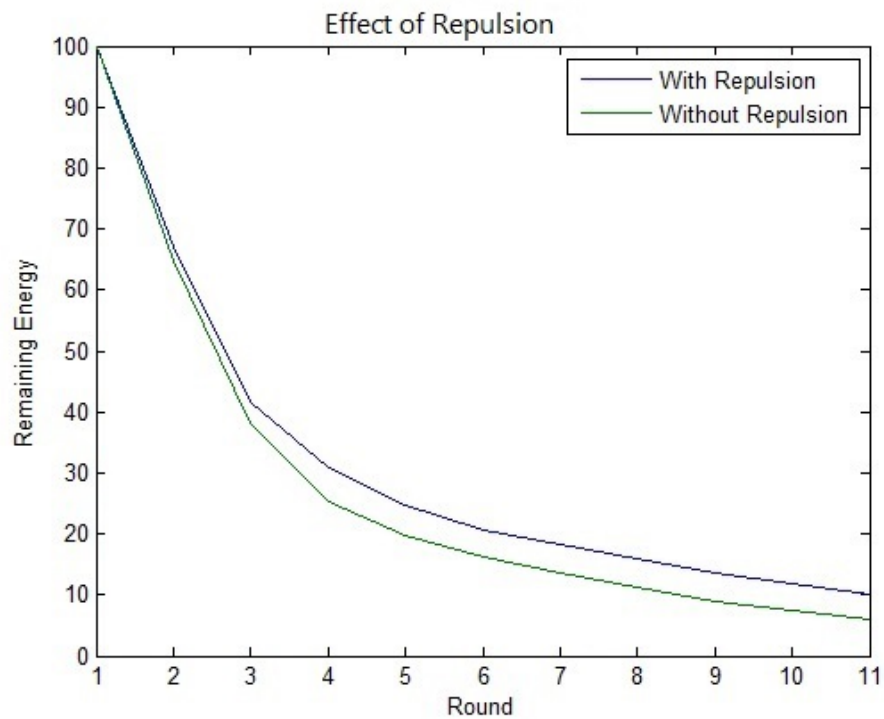
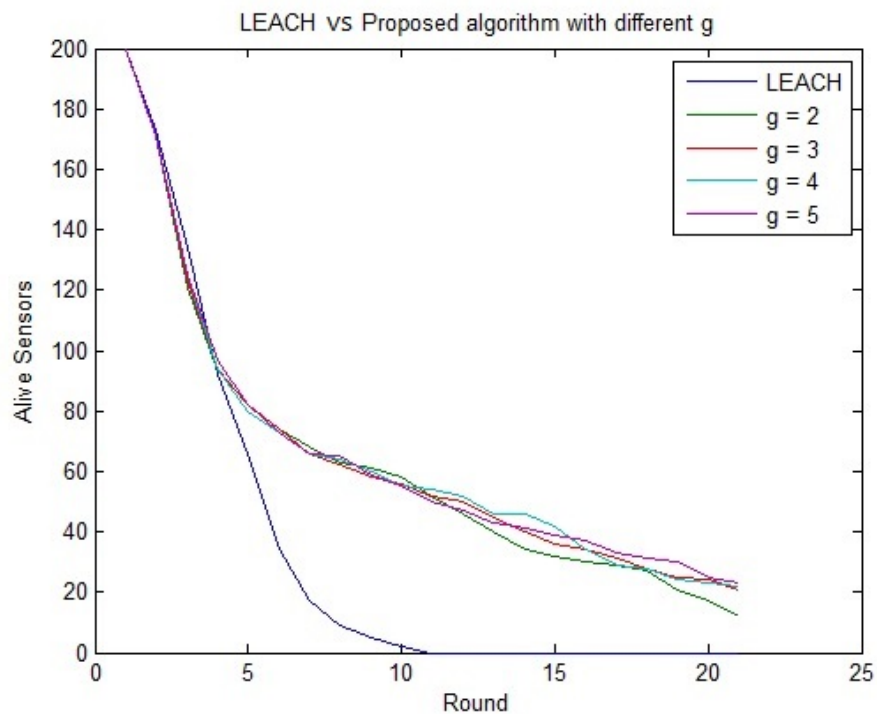


Figure 5.4: Effect of Repulsion on the Remaining Energy of Network

Figure 5.5: Effect of g on the Remaining Energy of Network

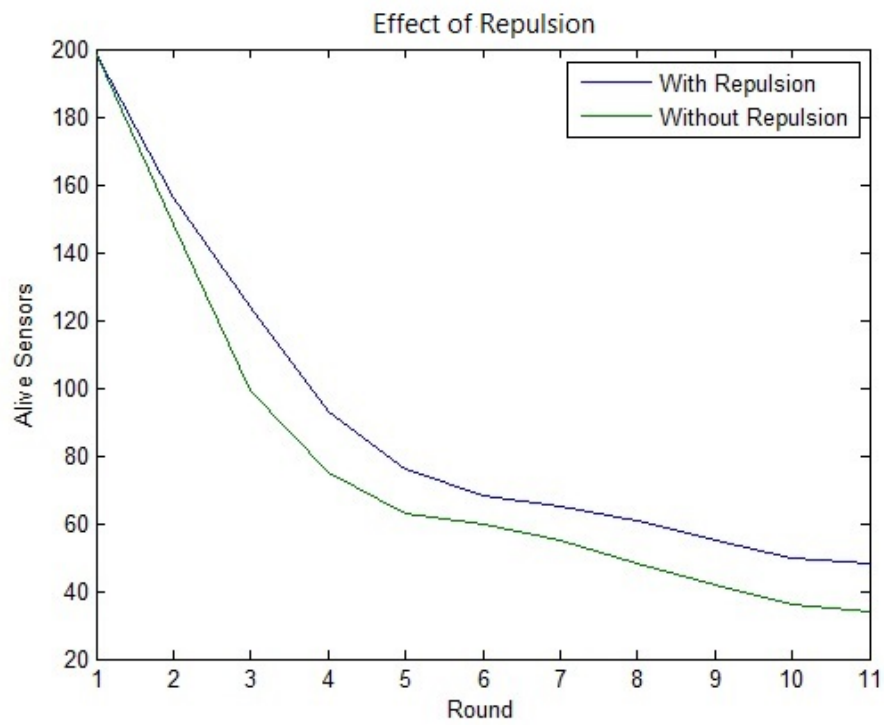


Figure 5.6: Effect of Repulsion on the Number Alive Sensors

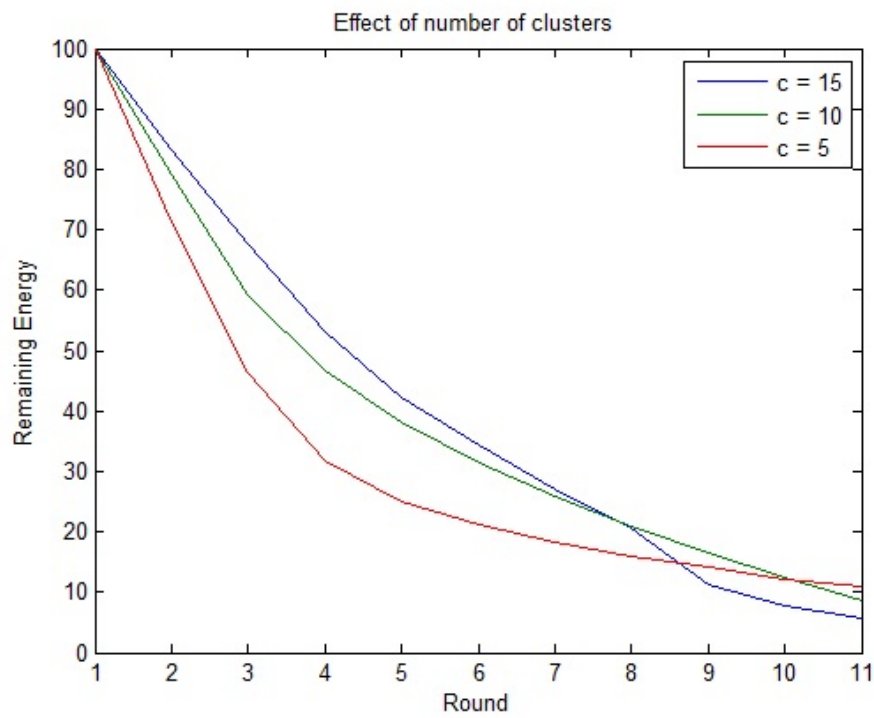


Figure 5.7: Effect of number of clusters on the Total Remaining Energy

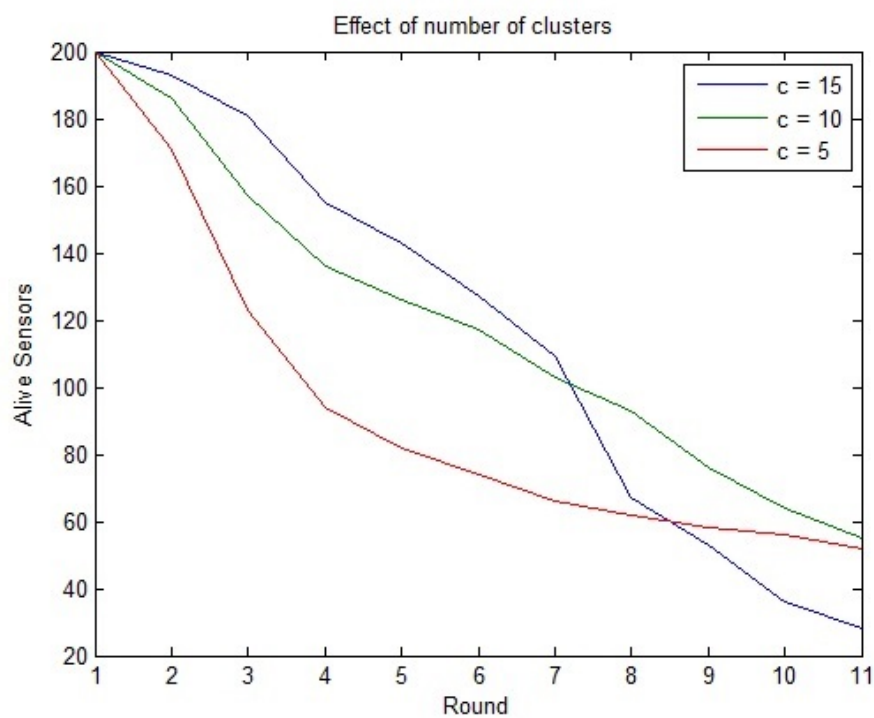
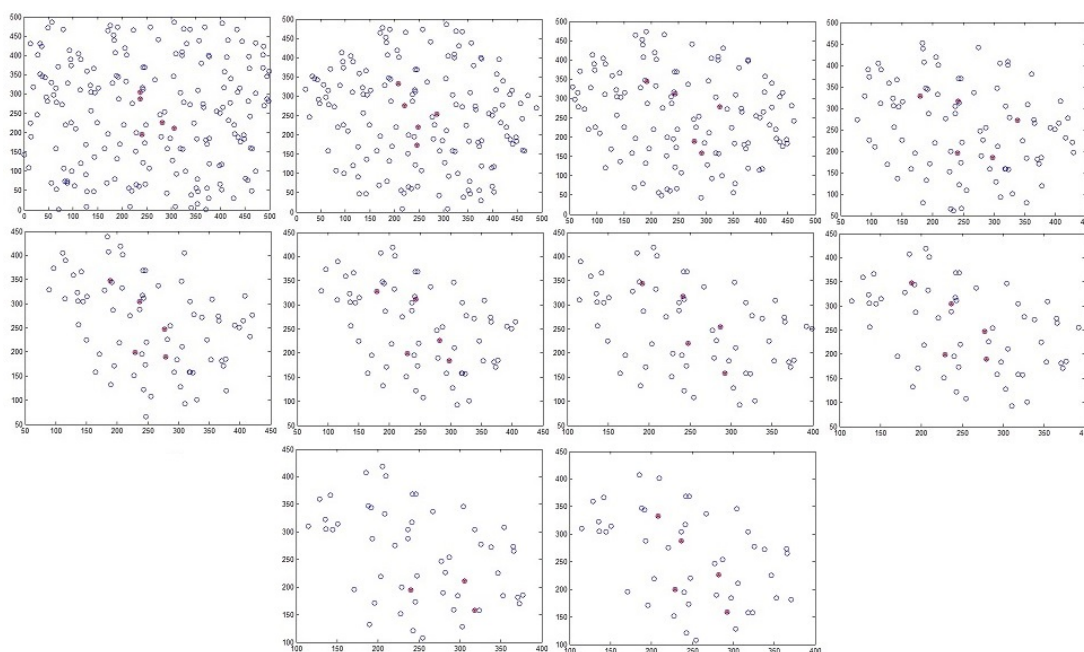


Figure 5.8: Effect of number of clusters on the Number of Alive sensors

Figure 5.9: CHs' locations over time in a 500×500 m space

Chapter 6

Conclusion and Future Research

Equality is a critical issue in various companies and organizations. We study equality in terms of a balance constraint added to infamous fuzzy c-means algorithm and call new problem as FCBC. The mentioned constraint forces clusters to have equal population with respect to their membership degrees.

We developed a heuristic method to solve FCBC problem. The proposed algorithm was applied to WSNs problem. Next the famous LEACH protocol was developed and proposed algorithm and LEACH protocol were compared. Results were examined based on two criteria, namely, remaining energy of network and also number of alive sensors in each round. Results show that proposed algorithm works better than exist LEACH algorithm in the simulated environment.

For the future research, we consider a simulated environment where, rather than a small number of rounds, number of rounds increased and in each round sensors according to their membership selects a single CH to communicate with. From methodological point of view, the other cluster based protocols to solve WSNs problem can be investigated.

Bibliography

- [1] Bandyopadhyay, Seema, and Edward J. Coyle. "An energy efficient hierarchical clustering algorithm for wireless sensor networks." INFOCOM 2003. Twenty Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies. Vol. 3. IEEE, 2003.
- [2] Ye, Mao, et al. "EECS: an energy efficient clustering scheme in wireless sensor networks." Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International. IEEE, 2005.
- [3] Luo, Jun, and J-P. Hubaux. "Joint mobility and routing for lifetime elongation in wireless sensor networks." INFOCOM 2005. 24th annual joint conference of the IEEE computer and communications societies. Proceedings IEEE. Vol. 3. IEEE, 2005.
- [4] Gupta, Gaurav, and Mohamed Younis. "Load-balanced clustering of wireless sensor networks." Communications, 2003. ICC'03. IEEE International Conference on. Vol. 3. IEEE, 2003.
- [5] Youssef, Adel M., et al. "Distributed Formation of Overlapping Multi-hop Clusters in Wireless Sensor Networks." GLOBECOM. 2006.

- [6] Dunn, Joseph C. "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters." (1973): 32-57.
- [7] Bezdek, James C. "Pattern recognition with fuzzy objective function algorithms." Kluwer Academic Publishers, 1981.
- [8] Bezdek, James C. "Pattern recognition with fuzzy objective function algorithms." Kluwer Academic Publishers, 1981.
- [9] MacQueen, James. "Some methods for classification and analysis of multivariate observations." Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1. No. 14. 1967.
- [10] Selim, Shokri Z., and Mohamed A. Ismail. "K-means-type algorithms: a generalized convergence theorem and characterization of local optimality." Pattern Analysis and Machine Intelligence, IEEE Transactions on 1 (1984): 81-87.
- [11] Bezdek, James C. "A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms." IEEE Transactions on Pattern Analysis and Machine Intelligence 2.1 (1980): 1-8.
- [12] Selim, Shokri Z., and M. A. Ismail. "On the local optimality of the fuzzy isodata clustering algorithm." Pattern Analysis and Machine Intelligence, IEEE Transactions on 2 (1986): 284-288.
- [13] Barni, M. Cappellini, and A. V Mecocci. "Comments on A possibilistic approach to clustering." (1996).
- [14] Salazar-Aguilar, M. Angelica, Roger Z. Rios-Mercado, and Jose Luis Gonzalez-Velarde. "A bi-objective programming model for designing compact and balanced

- territories in commercial districting." *Transportation Research Part C: Emerging Technologies* 19.5 (2011): 885-895.
- [15] Timm, Heiko, et al. "Fuzzy cluster analysis with cluster repulsion." *Euro. Symp. Intelligent Technologies (EUNITE)*, Tenerife, Spain. 2001.
- [16] Pal, Nikhil R., et al. "A possibilistic fuzzy c-means clustering algorithm." *Fuzzy Systems, IEEE Transactions on* 13.4 (2005): 517-530.
- [17] Hojati, Mehran. "Optimal political districting." *Computers & Operations Research* 23.12 (1996): 1147-1161.
- [18] Selim, Shokri Z., and Mohamed A. Ismail. "K-means-type algorithms: a generalized convergence theorem and characterization of local optimality." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 1 (1984): 81-87.
- [19] Dembele, Doulaye, and Philippe Kastner. "Fuzzy C-means method for clustering microarray data." *Bioinformatics* 19.8 (2003): 973-980.
- [20] Babuska, Robert. *Fuzzy modeling for control*. Kluwer Academic Publishers, 1998.
- [21] Bradley, P. S., K. P. Bennett, and Ayhan Demiriz. "Constrained k-means clustering." *Microsoft Research, Redmond* (2000): 1-8.
- [22] Woodbury, Max A., and Jonathan Clive. "Clinical pure types as a fuzzy partition." (1974): 111-121.
- [23] Hppner, Frank, ed. *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley & Sons, 1999.
- [24] Pal, Nikhil R., and James C. Bezdek. "On cluster validity for the fuzzy c-means model." *Fuzzy Systems, IEEE Transactions on* 3.3 (1995): 370-379.

- [25] Krishnapuram, Raghuram, and James M. Keller. "The possibilistic c-means algorithm: insights and recommendations." *Fuzzy Systems, IEEE Transactions on* 4.3 (1996): 385-393.
- [26] Guo, J., G. Trinidad, and N. Smith. "MOZART: A multi-objective zoning and aggregation tool." *Proceedings of the Philippine Computing Science Congress (PCSC)*. 2000.
- [27] Bandyopadhyay, Sanghamitra, Anirban Mukhopadhyay, and Ujjwal Maulik. "An improved algorithm for clustering gene expression data." *Bioinformatics* 23.21 (2007): 2859-2865.
- [28] Xie, Xuanli Lisa, and Gerardo Beni. "A validity measure for fuzzy clustering." *IEEE Transactions on pattern analysis and machine intelligence* 13.8 (1991): 841-847.
- [29] Schwmmle, Veit, and Ole Nrregaard Jensen. "A simple and fast method to determine the parameters for fuzzy cmeans cluster analysis." *Bioinformatics* 26.22 (2010): 2841-2848.
- [30] Kannan, S. R., et al. "Effective FCM noise clustering algorithms in medical images." *Computers in biology and medicine* 43.2 (2013): 73-83.
- [31] Hoang, Duc Chinh, Rasjesh Kumar, and Sanjib Kumar Panda. "Realisation of a cluster-based protocol using fuzzy C-means algorithm for wireless sensor networks." *IET Wireless Sensor Systems* 3.3 (2013): 163-171.
- [32] Julrode, Phichete, and Siriporn Supratid. "Improved Fuzzy Ant-Based Clustering: A Nonparametric Balance Between Exploitation and Exploration." *Research Journal of Applied Sciences* 8.9 (2013): 425-434.

- [33] Reinisch, Christian, et al. "Wireless technologies in home and building automation." *Industrial Informatics, 2007 5th IEEE International Conference on*. Vol. 1. IEEE, 2007.
- [34] Martnez, Diego, et al. "Wireless sensor and actuator networks: Charecterization and case study for confined spaces healthcare applications." *Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on*. IEEE, 2008.
- [35] Krishnapuram, Raghuram, and James M. Keller. "A possibilistic approach to clustering." *Fuzzy Systems, IEEE Transactions on* 1.2 (1993): 98-110.
- [36] Beni, Gerardo, and Xiaomin Liu. "A least biased fuzzy clustering method." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 16.9 (1994): 954-960.
- [37] Izakian, Hesam, and Witold Pedrycz. "Agreement-based fuzzy C-means for clustering data with blocks of features." *Neurocomputing* 127 (2014): 266-280.
- [38] Zarinbal, Marzie, M. H. Fazel Zarandi, and I. B. Turksen. "Interval Type-2 Relative Entropy Fuzzy C-Means clustering." *Information Sciences* 272 (2014): 49-72.
- [39] Cannon, Robert L., Jitendra V. Dave, and James C. Bezdek. "Efficient implementation of the fuzzy c-means clustering algorithms." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 2 (1986): 248-255.
- [40] Cai, Weiling, Songcan Chen, and Daoqiang Zhang. "Fast and robust fuzzy i_c c_j/i_c -means clustering algorithms incorporating local information for image segmentation." *Pattern Recognition* 40.3 (2007): 825-838.

- [41] Dunn, Joseph C. "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters." (1973): 32-57.
- [42] Selim, Shokri Z., and M. A. Ismail. "On the local optimality of the fuzzy isodata clustering algorithm." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 2 (1986): 284-288.
- [43] Nikolakopoulou, George, et al. "Solving a vehicle routing problem by balancing the vehicles time utilization." *European Journal of Operational Research* 152.2 (2004): 520-527.
- [44] Rovetta, Stefano. "Notes about some unsupervised learning methods." (2005).
- [45] Hillier, Frederick S. *Introduction to operations research*. Tata McGraw-Hill Education, 1995.
- [46] Baron, Opher, et al. "The equitable location problem on the plane." *European Journal of Operational Research* 183.2 (2007): 578-590.
- [47] Venkata Narasimha, Koushik, et al. "An ant colony optimization technique for solving minmax Multi-Depot Vehicle Routing Problem." *Swarm and Evolutionary Computation* 13 (2013): 63-73.
- [48] Pavone, Marco, et al. "Distributed algorithms for environment partitioning in mobile robotic networks." *Automatic Control, IEEE Transactions on* 56.8 (2011): 1834-1848.
- [49] Carlsson, John Gunnar. "Dividing a territory among several vehicles." *INFORMS Journal on Computing* 24.4 (2012): 565-577.

- [50] Heinzelman, Wendi Rabiner, Anantha Chandrakasan, and Hari Balakrishnan. "Energy-efficient communication protocol for wireless microsensor networks." *System Sciences*, 2000. Proceedings of the 33rd Annual Hawaii International Conference on. IEEE, 2000.
- [51] Lindsey, Stephanie, and Cauligi S. Raghavendra. "PEGASIS: Power-efficient gathering in sensor information systems." *Aerospace conference proceedings*, 2002. IEEE. Vol. 3. IEEE, 2002.
- [52] Manjeshwar, Arati, and Dharma P. Agrawal. "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks." *Parallel and Distributed Processing Symposium, International*. Vol. 3. IEEE Computer Society, 2001.
- [53] Manjeshwar, Arati, and Dharma P. Agrawal. "APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks." *Parallel and Distributed Processing Symposium, International*. Vol. 2. IEEE Computer Society, 2002.
- [54] Younis, Ossama, and Sonia Fahmy. "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks." *Mobile Computing, IEEE Transactions on* 3.4 (2004): 366-379.
- [55] Dasgupta, Sudakshina, and Paramartha Dutta. "An improved Leach approach for Head selection Strategy in a Fuzzy-C Means induced clustering of a Wireless Sensor Network." *Proc. IEMCON* (2011): 203-208.
- [56] Filippone, Maurizio, et al. "A survey of kernel and spectral methods for clustering." *Pattern recognition* 41.1 (2008): 176-190.