MIXED INTEGER PROGRAMMING AND HEURISTICS APPROACHES FOR CLUSTERING WITH CLUSTER-BASED FEATURE SELECTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SENA ÖNEN ÖZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

JULY 2019

Approval of the thesis:

MIXED INTEGER PROGRAMMING AND HEURISTICS APPROACHES FOR CLUSTERING WITH CLUSTER-BASED FEATURE SELECTION

submitted by **SENA ÖNEN ÖZ** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Yaşar Yasemin Serin
Head of Department, **Industrial Engineering** _____

Assoc. Prof. Dr. Cem İyigün
Supervisor, **Industrial Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Sinan Gürel
Industrial Engineering, METU _____

Assoc. Prof. Dr. Cem İyigün
Industrial Engineering, METU _____

Assoc. Prof. Dr. Zeynep Pelin Bayındır
Industrial Engineering, METU _____

Assoc. Prof. Dr. İsmail Serdar Bakal
Industrial Engineering, METU _____

Assist. Prof. Dr. Fatma Yerlikaya Özkurt
Industrial Engineering, Atılım University _____

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Sena Önen Öz

Signature :

ABSTRACT

MIXED INTEGER PROGRAMMING AND HEURISTICS APPROACHES FOR CLUSTERING WITH CLUSTER-BASED FEATURE SELECTION

Önen Öz, Sena
M.S., Department of Industrial Engineering
Supervisor: Assoc. Prof. Dr. Cem İyigün

July 2019, 135 pages

Cluster analysis tries to figure out the hidden similarities between data points in order to place similar data points into the same group and different data points into separate groups using unlabeled data. Understanding the data becomes difficult and the power of obtaining informative clusters for an algorithm decreases as the dimensionality of the data set gets high. Identifying the relevant features of high dimensional data sets is the mostly used technique in order to increase the performance of the algorithm to find the best clusters. However, selecting or deselecting the features comes up with the assumption that all the selected features have the same relevance for all clusters.

In this study, it is assumed that the features to be used in clustering may differ for each cluster. Number of clusters and number of relevant features in each cluster are given in advance. By using a center-based clustering approach, identifying the cluster centers, assigning data points to a cluster and selecting relevant features for each cluster are performed simultaneously. A mixed integer mathematical model is proposed which minimizes the total distance between data points and their cluster center by using the selected features for each cluster. Since the proposed model is not

linear, mathematical models using different linearization methods have been used to solve the problem. In addition to those mathematical models, we propose Benders Decomposition solution method implemented on our problem. Besides, two different heuristic algorithms have been developed by taking into account the nature of the mentioned problem. The proposed mathematical models and heuristic algorithms have been experimented on several data sets in different problem sizes in terms of number of clusters, number of relevant features and number of data points.

Keywords: clustering, feature selection, mathematical model, heuristic approach

ÖZ

KÜME ÖZGÜ ÖZNİTELİK SEÇİMİ İLE KÜMELEME PROBLEMİ İÇİN KARMA TAMSAYILI PROGRAMLAMA VE SEZGİSEL YAKLAŞIMLAR

Önen Öz, Sena
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Cem İyigün

Temmuz 2019 , 135 sayfa

Kümeleme algoritmaları, noktalar arasındaki önceden bilinmeyen gizli ilişkileri belirleyip birbirine benzeyen veri noktalarını aynı gruba, birbirinden farklı veri noktalarını ise ayrı gruplara koymayı hedefleyen gözetimsiz bir öğrenme yöntemidir. Ancak veri setinin boyutu arttıkça verinin anlaşılması zorlaştığından doğru kümelemeyi elde etme ihtimali düşer. En iyi kümelemeyi bulmak için kümeleri tanımlayan öznitelikleri belirlemek kümeleme algoritmalarının performansını arttırmak amacıyla büyük ölçekli veri setlerinde en çok kullanılan ön işleme tekniğidir. Ancak özniteliklerin ayırt edici olarak seçilip seçilmemesi, tüm özniteliklerin her küme için aynı ilgi düzeyine sahip olduğu varsayımıyla ortaya çıkmaktadır.

Bu çalışmada, kümelemede kullanılacak özniteliklerin her bir küme için farklılık gösterebileceği varsayılmaktadır. Küme sayısı ve her küme için ilgili öznitelik sayısı önceden verilmektedir. Küme merkezleri bazlı bir kümeleme yaklaşımı kullanılarak, küme merkezlerinin belirlenmesi, veri noktalarının bir kümeye atanması ve her bir küme için ilgili özniteliklerin seçimi eş zamanlı olarak yapılmaktadır. Bu çalışma kapsamında küme içindeki noktaların ilgili küme merkezine seçilen öznitelikler üze-

rinden uzaklıklarının toplamını enazlayan karma tamsayılı bir matematiksel model önerilmiştir. Önerilen model doğrusal olmadığı için problemin çözümünde farklı doğrusallaştırma yöntemlerinin uygulandığı matematiksel modeller kullanılmıştır. Bunun yanı sıra, problemin çözümü için Benders Ayrıştırma yöntemi uygulanmıştır. Ayrıca, belirtilen problem için iki farklı sezgisel çözüm yöntemi geliştirilmiştir. Önerilen matematiksel modeller ve geliştirilen sezgisel çözüm yöntemleri nokta ve öznitelik sayısı açısından farklı büyüklükteki veri setleri üzerinde denenmiştir.

Anahtar Kelimeler: kümeleme, öznitelik seçimi, matematiksel model, sezgisel yaklaşım

To my family...

ACKNOWLEDGMENTS

First of all, I would like to express my sincere thanks to my supervisor Assoc. Prof. Dr. Cem İyigün for his constant support, academic guidance and patience. I learned a lot from him during my graduate studies both as a student and as a future academician. His wisdom has guided me to new research areas that otherwise I would be unaware of.

I would also like to thank the members of the examining commitee Prof.Dr. Sinan Gürel, Assoc. Prof. Dr. Zeynep Pelin Bayındır, Assoc. Prof. Dr. İsmail Serdar Bakal and Assist. Prof. Dr. Fatma Yerlikaya Özkurt for their valuable feedback and time in reviewing this work.

I would like to thank my beloved friends and my colleagues: to Dilay Aktaş, who makes any struggle a journey for me, for making me feel comfortable in my most difficult times; to Melis Özateş Gürbüz for being a kind companion and a superhero who cares about everything; to Altan Akdoğan for his endless motivational talks; to my roommate, my first co-worker, my partner in courses and whom I have not had enough of her friendship, Cansu Alakuş; to Can Barış Çetin who is always with me with his unconditional support and lastly to Engin Kut for being there for us during our late night studies.

Most importantly, my deepest thanks go to love of my life, Burak Öz, for being with me every step I was afraid to take and his encouraging support. There is no doubt that I would not be where I am today without his support, his caring love, and his patience.

Last but not least, I thank to my parents for always believing in me and expressing their support. I could not think of an acklowledgements section without mentioning my brother who always makes me feel stronger. I cannot thank my family enough to express my gratitude.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

BD	Benders Decomposition
CBFS	Clustering and Cluster based Feature Selection
$CPUs.$	Computational Time
D_{SP}	Dual of the Subproblem
FCM	Fuzzy C-Means
$\%Gap_B$	Percent Gap from Best Available Solution
$\%Gap_M$	Percent Gap from Made Objective
$\%Gap_O$	Percent Gap from Optimal Solution
H_1	Benders like Heuristic Algorithm
H_2	Iterative Heuristic Algorithm
H_X	Center and Feature Problem
LB	Lower Bound on Objective Function
LM_1	Linearized Model 1
LM_2	Linearized Model 2
LM_3	Linearized Model 3
MP	Master Problem
NBest	Number of Times the Best Available Solution is Found
NOpt	Number of Times the known Optimal Solution is Found
NTL	Number of Times the Model Hits to Time Limit
NM	A Non-linear Mixed Integer Model for CBFS
PAM	Partitioning Around Medoids
PD-Clustering	Probabilistic Distance Clustering
PCA	Principal Component Analysis
P_C	Center Selection Problem

P_{CX}	Center-Assignment Update Subroutine
P_Q	Feature Selection Problem
P_{QX}	Feature-Assignment Update Subroutine
P_X	Assignment Problem
P_{XC}	Assignment-Center Update Subroutine
SP	Subproblem
UB	Upper Bound on Objective Function

CHAPTER 1

INTRODUCTION

Enormous amount of data are obtained in various formats such as transaction records on bank accounts, sensor data obtained from devices used to gather climate information, and posts on social media. All of these data are generated very fast, and must be processed as soon as possible to create valuable information. Analyzing big data and finding relevance of data is too difficult, time consuming and costly. Hence, data mining methods were developed to ease the process of obtaining meaningful information (Jain, 2010).

Well-known data mining tasks are grouped into two categories depending on the availability of information, which are *supervised learning* and *unsupervised learning*. In supervised learning, data has the label information and labels are used to train the algorithm. Then, the trained algorithm is used to predict the unknown label of new observations. Classification and regression are considered in this class. On the other hand, there is no known labels used in unsupervised learning. Unsupervised learning algorithms aim to obtain meaningful information using available unlabeled data. Clustering is a widely studied unsupervised learning method. It aims to group similar data points in a data sets into the same cluster by separating them from the data points which are dissimilar.

There are some challenges associated with clustering problems, and choosing the similarity measure is one of them. The measure should be selected by considering the properties of data sets, whether it contains quantitative (continuous or binary) or qualitative (categorical) features. Also, objective function which can be used in grouping data sets may change. As the definition of clustering problem suggest, clusters should be well-separated because we try to put dissimilar data points into different clusters.

This aim is named as *separation* in the literature. Also, similar data points should be in the same cluster, and it is obtained by measuring the *compactness*. Since clustering problem is considered as an unsupervised learning, the number of clusters is not known a priori and the data has no label information. Besides, different approaches used in clustering may come up with different clustering solutions. Therefore, selection of similarity measure and objective function, constructing different numbers of clusters, and using different solution techniques may generate totally different clustering solutions. Hence, proposing a universal quality measure for clustering is not possible.

In the literature, several studies propose solution to clustering problem by using different methods, objective functions and data sets with different properties. In general, we can classify clustering approaches as *Partitional Clustering* and *Hierarchical Clustering*. The former method aims to generate disjoint clusters in such a way that similar data points are gathered around a cluster representative. The latter constructs a hierarchical cluster structure of a data set, and it enables to obtain different clustering solutions if the cluster structure is cut at different levels. There are advantages and disadvantages of these methods, and they will be discussed in Chapter 2. Most of the studies include heuristic approaches to obtain clusters in a reasonable time due to the complexity of the problem. However, in Chapter 2, studies including exact solution methods for clustering will also be delivered.

Data sets may contain redundant and irrelevant features as well as relevant ones. Clustering patterns hidden in the data set may be masked by the redundant or irrelevant features. In order to obtain clusters in a reasonable time, redundant and irrelevant features should be removed from data set. This phenomenon is called as "*dimension reduction*", and there are numerous techniques to do that. In the literature, most of the dimension reduction techniques assume that the same features may be used to identify all clusters. However, global selection of features may not be helpful to obtain all clusters in a data set, since there may be no common subset of features that is relevant for all clusters. In the literature, there are some studies proposed to overcome this problem, and both data points and features are clustered simultaneously. These studies will be called as local clustering algorithms throughout this study.

Our problem in this study is to find clustering solution with feature selection but we focus on identifying clusters via different subsets of features. That means, each cluster is described by a different set of features. Here, it is allowed that the same feature might be relevant for many other clusters. In the first part of this study, a mixed integer mathematical model has been proposed for the solution of clustering problem with cluster based feature selection (CBFS). The model decides (i) location of the cluster centers, (ii) features to be selected for each cluster, and (iii) assignment of data points to a cluster simultaneously. The number of clusters and number of features that will be selected for each cluster are given in advance. The aim is to minimize total distance between the data points and cluster centers through selected features. Here, it should be noted that *Partitional Clustering* is implemented, and cluster centers are selected among data points. Also, data sets only include continuous features, and similarity measure is selected as $L1-norm$. Since the proposed mathematical model include nonlinear term in the objective function, different linearization methods have been applied in order to increase the performance.

In the second part of the study, Benders Decomposition approach is applied on our problem in order to obtain the exact solution. Benders Decomposition method is used to solve large-scale optimization problems, and the nature of our problem is suitable to apply Benders Decomposition. When the data set gets larger in terms of number of features or number of data points, the solution time of the proposed mathematical models gets worse. After obtaining insights about the problem structure by Benders Decomposition, a Benders like heuristic algorithm is constructed. This heuristic algorithm uses a new mathematical model which only decides the cluster centers and relevant features of the clusters. When the cluster centers and selected features are obtained from the mathematical model, each data point will be assigned to the their closest center in order to minimize the total distance between data points and their cluster centers via selected features of that clusters. For the assignments of data points, a simple search procedure is developed. Additionally, a new heuristic algorithm has been introduced in the second part of this thesis. This heuristic algorithm decides each decision variable by iteratively solving smaller problems. That means, at each iteration, two of the decision variables are fixed, and the other is decided by the defined smaller problem specific to problem context.

There may be no common subset of features that can be used to identify all clusters. Therefore, we try to identify relevant subset of features which are specific to clusters. To the best of our knowledge, we are contributing to the literature by using mathematical models and proposing heuristic algorithms to solve clustering problem and cluster based feature selection simultaneously. The mentioned problem may be used in different real life applications. In customer segmentation, customer groups can be identified by clustering with cluster based feature selection, and services which are provided to those customer groups may be arranged depending on the features of groups. Another example might be team formation where teams in a work environment are formed considering the skill(features) of employees in order to assign task accordingly. Last but not least, cluster based feature selection can be used in order to identify the similar regions in an image where the similarity depends on the features of regions.

This thesis is organized as follows. In Chapter 2, background information on clustering and its properties are given. Related studies in the literature specifically for clustering and feature selection in clustering problems are discussed. In Chapter 3, we define the problem with its properties, and proposed mixed integer linear programming models for our problem are delivered. Results of the experimental studies conducted on these models are presented in Chapter 4. Benders Decomposition method implemented to our problem and Benders like heuristic algorithm are explained in Chapter 5. Chapter 6 will introduce a new heuristic algorithm which works in a way that all decision variables are decided by an iterative solution method. Chapter 7 includes experimental results for heuristic algorithms and their comparison with one of the proposed mathematical models. Finally, in Chapter 8, the main findings obtained from this study and future research directions are delivered.

CHAPTER 2

BACKGROUND ON CLUSTERING AND LITERATURE REVIEW

As stated by Xu and Wunsch (2005), in the literature, there are many studies in various disciplines, such as marketing, biology, economy and medicine which use clustering in the literature. In this chapter, basic terminology about clustering, commonly used clustering approaches and studies related to our problem and their contribution to the literature will be discussed. In Section 2.1, definition of the clustering problem, its properties and solution approaches will be introduced. Then, we describe the need for reducing the size of data sets in Section 2.2. In this section, advantages and disadvantages of different dimension reduction methods will be discussed. Next Section 2.3 will introduce the studies on clustering and feature selection in clustering problems. Lastly, simultaneous clustering of data points and features will be discussed in Section 2.4.

2.1 Clustering Problems

Cluster analysis is an unsupervised learning technique used to figure out the hidden similarities of data points in order to designate the relationship between them. Data sets are grouped by considering the similarities of data points according to predefined similarity measure. By this way, similar data points are grouped in a cluster whereas dissimilar ones are assigned to different clusters.

Selection of the similarity measure can be considered as one of the main challenges of clustering problem. Data sets may include quantitative (continuous or binary) or qualitative (categorical) features which represent the properties of the data points. Therefore, choice of the similarity measure highly depends on the type of the features.

- **Quantitative Features:** Similarity is measured with any distance metrics like $L_1 - norm$ or $L_2 - norm$ where data sets contain only continuous features.

- **Qualitative Features:** There are some special similarity measures like Hamming Distance, Rand Index, and Jaccard Coefficient that can be used with categorical features.

In clustering problems, selection of objective function is also an important issue. The definition of clustering problem states that obtained clusters should be well-separated since dissimilar data points are aimed to be grouped into different clusters. Therefore, the distance between clusters should be maximized. This aim is named as *separation* in the literature. Mostly used separation measures are linkage metrics such as complete-link or single-link when separation is taken as the point-wise distance. Those measures consider data points grouped in different clusters, and calculate the pairwise distance between those data points. Complete-link takes two farthest data points from different clusters and calculates the distance between those data points, whereas single-link takes into account the closest data points of different clusters to measure separation. Also, cluster analysis puts similar data points into the same cluster and those data points should be close to each other. It is named as *compactness* and this measure should be minimized. In the literature, several compactness measures are proposed which can be grouped into two categories as representative point based and individual point-wise compactness measures. Representative point based compactness measures define similarity as the distance between a cluster representative and data points in that cluster. On the other hand, pairwise data point similarities within a cluster are used in individual point-wise compactness measures instead of the similarity between cluster representatives and data points. These measures perform differently depending on the used clustering algorithms and properties of the data types. There are some studies that both of those measures are used as objective functions separately. Also, a combination of those measures is used as a single objective in the literature like the ratio of separation and compactness which will be maximized.

Due to the nature of the data sets, the true cluster structures and number of clusters in a data set cannot be known in advance. Using different similarity measures or ob-

jective functions may result in obtaining different clustering solutions. Evaluating the obtained clustering solutions is not straightforward since there is no label information. Due to these issues, in the literature, there are several clustering algorithms. According to Jain et al. (1999), those approaches can be classified under two main categories namely *partitional clustering* algorithms and *hierarchical clustering* algorithms.

2.1.1 Partitional Clustering

Clusters are formed around the cluster centers which are representatives of the clusters. Due to the definition, partitional clustering methods are also known as *center-based clustering*. They can be classified based on the assignments of the data points:

- **Hard Clustering:** Each data point should only be assigned to one cluster center, and disjoint clusters will be obtained.
- **Soft Clustering:** Assignment of a data point to a cluster will be associated with the membership value. That means, a data point may be assigned to multiple clusters with a probability.

Total distance between data points and their cluster center is the mostly used objective function in partitional clustering algorithms. Due to the type of the objective function, a data point is closer to the cluster center of the assigned cluster, than the other cluster centers. Partitional clustering algorithms require lower memory and time, and they are good at clustering large data sets. The main problem of these algorithms is that the number of clusters should be given in advance to the algorithm. Also, they only work well with data sets with all quantitative features. Besides, clustering solution will differ depending on the initial cluster centers. Therefore, the performance of the algorithms highly depends on the initial selection of cluster centers.

K-means (MacQueen, 1967) and PAM (Kaufman and Rousseeuw, 1990), which is a kind of k-medoid clustering, are the commonly used partitional clustering algorithms, in the literature. Both of these algorithms work in an iterative fashion while trying to minimize the distance between data points and their cluster centers. Starting with randomly selected initial cluster center, assignment of each data point to a cluster

is performed. With these assignments, new cluster centers are calculated, and these steps are repeated until assignments of data points do not change. There are two main differences between those algorithms. K-means tries to minimize the total squared distances, while PAM minimizes the sum of distances between data points and their cluster centers. Hence, the latter is less sensitive to the outliers. Also, cluster centers are selected among data points in PAM, whereas cluster centers are the means of all the data points within clusters in K-means algorithms. In both of these algorithms, hard clustering is performed, that is, each data point is only assigned to one cluster.

Fuzzy C-Means (FCM) (Bezdek et al., 1984) and Probabilistic Distance Clustering (PD-Clustering) (Ben-Israel and Iyigun, 2008) are the approaches for the soft clustering. Those algorithms also work in iterative manner, but there is also a membership value for each data point to each center. In each iteration, centers of the clusters and membership value of data points are computed until convergence. Between these two algorithms, objective function, calculation of membership value and calculation of cluster centers differ. Convergence criteria is also different among those algorithms. The former stops when the objective function value does not change, whereas the latter terminates when there is no change in the center locations anymore. These methods are useful when the boundaries among clusters are not well-separated.

2.1.2 Hierarchical Clustering

Hierarchical clustering creates nested partitions of data sets. There will be a hierarchical relation between the created clusters, and those clusters are represented with dendrograms. Different partitions of the data sets can be found at each level of these tree-like structures. Algorithms of this type can be grouped into two, namely divisive methods and agglomerative methods.

Divisive methods, which are also known as top-down approaches start with a single all-inclusive cluster and split the chosen cluster into two until having only clusters with one data point. Whereas *agglomerative methods* (bottom-up approaches), start with the individual clusters that include single data point and merge clusters at each iteration until getting a single all-inclusive cluster. Agglomerative methods try to minimize the linkage criterion with selected distance metric while merging the clus-

ters. *Single-link* and *complete-link* are the most commonly used linkage criteria. As defined before, in single-link, the proximity of the clusters is defined as the distance between the two closest points of different clusters. At each step, two clusters with the minimum single-link are merged. In complete-link, the distance between the farthest points of two different clusters is considered as a proximity. Algorithm merges two clusters with the smallest complete-link.

Main advantage of hierarchical clustering is that the number of clusters is not necessarily be given in advance. Also, they can be used for qualitative data sets as well as quantitative ones. However, they are sensitive to noise and outliers. Also, they are computationally expensive comparing to partitional clustering. Besides, the mistake made in any iteration cannot be fixed.

CURE (Guha et al., 1998) and BIRCH (Zhang et al., 1996) are the well-known hierarchical clustering algorithms used in many applications.

2.2 Dimension Reduction in Clustering

Data features (attributes) represent the properties of data sets. Each feature stores some information about the data point. As the number of features increases, the dimensionality of the data set increases as well. Therefore, understanding the data becomes difficult and the chance of obtaining useful information decreases. The phenomenon is known as "Curse of Dimensionality" in the literature.

Data sets may include relevant and irrelevant(or redundant) features. Irrelevant and redundant features do not contain useful information to analyze the data (Dy and Brodley, 2004). The difference between redundant and irrelevant features can be seen in Figure 2.1. Figure 2.1 (a) shows that clusters can be identified by using either feature x or y. Therefore, we can eliminate one of them while clustering the data set. However, in Figure 2.1 (b), feature y does not contain any information to separate data points of two clusters. That means, when it is eliminated, we do not lose any information to identify clusters. Even it would be beneficial to eliminate feature y to obtain clusters in less computing time. As in Figure 2.1 (c), both features x and y should be used together to identify clusters. They are both relevant features.

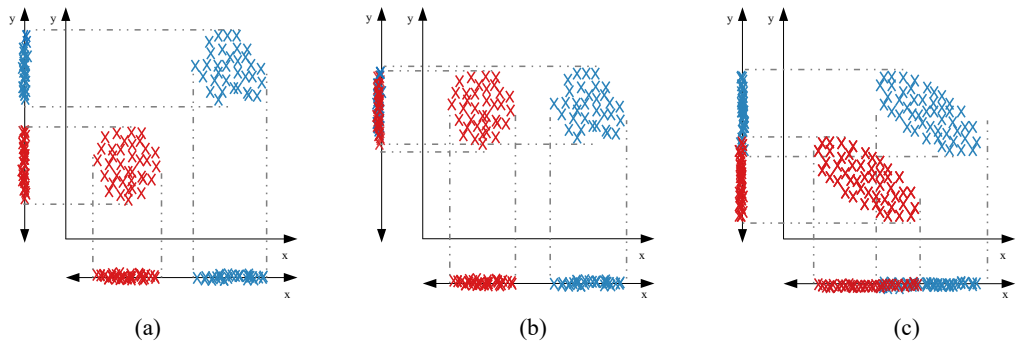

Figure 2.1: Examples for Redundant and Irrelevant Features

In order to extract meaningful information from a data set, dimension reduction is an important step for machine learning problems. In order to decrease the size, feature extraction or feature selection algorithms can be applied (Alelyani et al., 2013). Both of these methods improve prediction power of the clustering, reduce computational requirements and effects caused by the "Curse of Dimensionality".

2.2.1 Feature Extraction in Clustering

Feature extraction methods decrease the size of feature set by projecting the original features into a new feature space with a lower dimension. The original features are combined, so new features are created. Since the combination of original features are used, the interpretation of the results will be difficult. Also, it can be noted that noisy data will adversely affect the performance of the feature extraction methods since the transformation of data considers all data.

Principal Component Analysis (PCA) can be given as an example for the most commonly used feature extraction methods. The method uses covariance matrix to generate eigenvalues and eigenvectors. We will obtain *principal components* in order of significance by ranking eigenvectors in order of their corresponding eigenvalues in descending order. *Principal components* are new uncorrelated variables which are the linear combinations or mixtures of the initial variables. Some of the *principal components* are eliminated since they include less information about variability of the data. However, Law et al. (2004) argue that using the features with high variance does not indicate that those features are meaningful for clustering.

2.2.2 Feature Selection in Clustering

Feature selection uses original features throughout the process and selects a subset of original features for analyzing the data. The subset of features is selected based on the underlying distribution pattern in the case of unsupervised learning or the relevance of label information in the case of supervised learning (Alelyani et al., 2013).

Feature selection approaches can be grouped into three main categories as *filter methods*, *wrapper methods*, and *hybrid methods* in data mining. Here, we shortly explain these methods through the clustering problems.

Filter Methods

Filter methods are performed prior to the clustering algorithms. Therefore, quality of the features are not measured with clustering analysis. In these methods, the scores of features by a suitable ranking criterion are computed. Features that have lower scores than the predefined threshold value are removed from further analysis. The resulting feature subset is provided as an input to clustering (Chandrashekar and Sahin, 2014).

In most cases, feature dependencies are ignored and score of each feature is separately considered. These techniques are named as univariate filter methods. They may adversely effect the performance of the clustering algorithms (Saeys et al., 2007). To overcome this drawback, multivariate filter methods are proposed which take into account the dependencies and correlations between features. So, they can handle redundant features as well as irrelevant ones.

Filter methods are performed only once before the clustering since they are independent from the clustering algorithm. Therefore, they are computationally simple and fast, and useful for high-dimensional data sets. However, the ignorance of the interaction between feature selection and clustering algorithm can be considered as the disadvantage of these methods.

Wrapper Methods

Wrapper methods search the feature space to obtain the best subset. In order to find the best subset, wrapper methods start with any subset of features, then evaluate the

performance of the subset by clustering quality. These methods iteratively evaluate the performance of the possible feature subsets until the desired level of quality is obtained. Since considering all possible subsets of feature space is almost impossible, heuristic search algorithms are used mostly.

As opposed to filter methods, wrapper methods are computationally expensive since they interact with the clustering algorithm in evaluating the feature subsets, (Cai et al., 2018). However, wrapper methods provide better clusters since the interaction between clustering algorithm and feature selection is not ignored. But, their performance may change depending on the used clustering algorithm and there is a risk of overfitting.

Hybrid Methods

Hybrid methods try to eliminate the drawbacks of filter and wrapper methods. They can be considered as a combination of those two methods. Hybrid methods are computationally inexpensive than wrapper methods and they can capture relationships between features contrary to filter methods. Filtering method is used to obtain candidate subsets of features. By this way, number of subsets to be evaluated is reduced. Candidate subsets are evaluated with clustering algorithms as in wrapper methods.

So far a brief background for the clustering problem and the feature selection problem are provided. The literature review related to our problem will be covered in the rest of this chapter.

2.3 Literature Review for Clustering and Feature Selection in Clustering

In this study, mathematical models and heuristic algorithms are proposed to obtain a solution for the clustering problem with feature selection. Therefore, this chapter will cover exact solution methods and heuristics approaches used for both clustering and feature selection in clustering. It can be noted that there are not many studies aiming to obtain exact solution for clustering or feature selection in clustering since there is no prior label information.

2.3.1 Exact Solution Methods for Clustering

Apart from traditional algorithms proposed to solve clustering problem, the problem is also handled within the framework of operations research. It is observed that the center-based clustering problem can easily be modeled and solved as an optimization problem due to its similarity to p-median facility problem (Olafsson et al., 2008).

Vinod (1969) provides two mathematical formulations for the grouping problem where one of them uses the principle of facility location problems. The study shows the analogy between the facilities and the cluster centers, and also customers and data points. Predetermined number of cluster centers are selected and data points are allocated to a cluster as in the facility location problems. Therefore, there are two binary decision variables which indicate if a data point is selected as a cluster center and whether a data point is assigned to a cluster or not. The formulation aims to minimize the total cost of assigning all data points to a cluster.

In the years following this study, Rao (1971) extends the study published by the Vinod (1969) by considering two different objective functions, minimizing the maximum distance within clusters which is the farthest distance between data points which are assigned to the same cluster, and minimizing the total within cluster sum of square distances. The study takes definition of two binary decision variables as given in the previous study. Also, Bradley et al. (1996) use mathematical model to decide only cluster centers. They divide the problem into smaller problems, and they assign each data point to its closest cluster after finding the cluster centers. They actually implement the K-medoid algorithm within the concept of optimization. Note that, in all of these studies, cluster centers are selected among the data points.

To obtain optimal solutions for clustering problem, all feasible solutions must be evaluated. Unfortunately, with the increase in the problem size, the number of feasible solutions grows exponentially. For this reason, enumeration of all solutions is computationally infeasible for large problems. Some problems can be solved optimally without explicitly enumerating all feasible solutions by using the branch and bound solution method. In the literature, there are various studies solving clustering problem with branch and bound procedure. Koontz et al. (1975) propose a branch and bound

procedure where nodes of the tree include assignment of data points. Since they do not propose a mathematical model in their article, Klein and Aronson (1991) combine the branch and bound algorithm with modeling the clustering as an optimization problem. In order to find the optimal solution of the problem which has the same objective functions used in Rao (1971), Brusco (2003) uses branch and bound algorithm proposed by Klein and Aronson (1991) but the definition of lower and upper bounds are changed in order to obtain tight bounds. The main similarity between those studies is that they all consider cluster centers as anywhere in the vector space. Therefore, data points are assigned to cluster itself not to the cluster center. Brusco and Stahl (2006) cover branch and bound applications in clustering and also for feature selection with different objective functions.

2.3.2 Heuristic Approaches for Clustering

For the solution of the mathematical model of clustering problem proposed by Vinod (1969) and Rao (1971), a heuristic algorithm which includes Lagrangian relaxation is proposed by (Mulvey and Crowder, 1979). They relax the assignment constraint which is difficult to meet by adding it to objective function. Algorithm starts with finding a good initial clustering solution and cluster centers. By using subgradient method a lower bound on the objective function has been obtained. Since the assignment constraint is relaxed in the subgradient method, the solution may not be feasible. Therefore, in the next step, each data point is assigned to a closest cluster center, and a feasible solution and an upper bound on the objective function are obtained by this way. The heuristic algorithm terminates when the gap between lower and upper bounds is smaller than the predefined threshold or iteration limit has been reached.

Among partitional clustering algorithms, K-means clustering is known to be more sensitive to outliers in comparison with the K-medoid clustering. Here, we will cover some of the studies enhancing K-medoid clustering. PAM (Partitioning Around Medoids) (Kaufman and Rousseeuw, 1990) is a benchmark algorithm for K-medoid clustering. The algorithm consists mainly two steps which are build and swap. Initial solution is generated randomly in the build step, that is, initial cluster centers are

selected among data points (medoid). In the swap step, for each of the non-medoid data points, the change in the objective function is calculated when that point is considered as a center in its cluster instead of the current medoid of that cluster. The data point which causes the most decrease in the objective function is selected as a new medoid. The algorithm is repeated until there is no change in the objective function. In the literature, there are many studies improving the performance of PAM algorithms. However, we will cover the most recent and distinctive ones.

In classical PAM, all possible swap calculations are made in order to find the highest decrease in the objective function in each step. Reynolds et al. (2006) discuss a way to speed up PAM by decreasing the calculations in swap step. They show that the change in the objective function can be decomposed into two components, where the first component depends only the removal of a medoid, the second component depends only on the selection of a new point as cluster center. Also, in order to decrease the computational time of PAM, Park and Jun (2009) propose a new algorithm for K-medoid clustering where it differs from the traditional K-medoid clustering in terms of initial selection of cluster centers. The new algorithm tends to select the k most middle data points as initial centers. Then, the algorithm continues with assigning all data points to the closest cluster centers. New medoid of each cluster is selected among the data points in that cluster, which is the one that has minimum total distance to other data points in its cluster. Instead of using distances directly, Zadegan et al. (2013) use a rank matrix which is constructed by sorting the distance between each data point and storing the indices of data points starting from the most similar one. New medoids are selected according to a new measure named as *hostility* which simply use the sorted ranking and find the data point that is in the middle of the group. In this algorithm, the convergence is defined by the number of iterations.

Sağlam et al. (2006) try to minimize the maximum cluster diameter as studied by Rao (1971) using mathematical models but they observed that performance of the model is not comparable with the other studies in the literature in terms of computational time. Therefore, they have proposed a heuristic algorithm which starts with fixing the assignment of some of the data points and taking them as a fixed points in their mathematical formulation. In order to eliminate the poor quality clusters, data points are reassigned to clusters after clustering solution is obtained.

The Vertex Substitution Heuristic proposed by Teitz and Bart (1968) performs an iterative search among all possible centers. It replaces selected center with an unselected data point that will decrease the objective function where it is a minimization problem. The process is repeated until the objective function converges during the exchange steps. Since the algorithm starts with random selection of centers, the resulting solution will not be globally optimal solution in general. Hence, it will generate upper bound to the optimal solution of the p-median clustering.

Brusco and Köhn (2008) propose three stages heuristic method to select predefined number of cluster centers and assign data points to a cluster while trying to minimize sum of Euclidean Distance between cluster centers and data points. In the first stage, Vertex Substitution Heuristic proposed by Teitz and Bart (1968) is implemented. The stage is performed multiple times in order to obtain tight upper bounds and eliminate the effect of starting centers. In the second stage, Lagrangian Relaxation method is applied. The relaxed problem is solved by iterative subgradient optimization method, and lower bound on p-median clustering problem is obtained. If the lower bound obtained by Lagrangian Relaxation and the upper bound obtained by Vertex Substitution Heuristics is the same, they state that the solution is optimal. If it is not optimal, then at the third stage, branch and bound algorithm is applied.

Kim et al. (2009) follow the optimization method that has been suggested by Shi and Ólafsson (2000) named as *Nested Partitions* to obtain clusters. At the beginning of the algorithm the feasible region is divided into subregions, and it is assumed that there is a subregion which is promising to have the best solution. The most promising region is divided into predefined number of subregions, and what remains is aggregated into one region called the surrounding region. To evaluate each of these regions, a randomly generated clustering solutions is used, and a promising index is calculated to select the next promising region. If one of the subregions is the best, this region becomes the most promising region. If the surrounding region is the best, the algorithm backtracks to a larger region that contains the old most promising region. It is observed that the idea is actually similar to the branch and bound algorithm.

Fahad et al. (2014) review the traditional clustering algorithms in their study in detail. There are also several types of metaheuristics used in clustering such as Simulated

Annealing, Tabu Search, and Evolutionary Algorithms. Nanda and Panda (2014) provide detailed review of those metaheuristics used in partitional clustering.

2.3.3 Exact Solution Methods for Feature Selection in Clustering

In the classical clustering problem, clusters are identified using all features but as it is mentioned there may be irrelevant variables that may worsen the performance of clustering algorithms. In order to eliminate those irrelevant features, feature selection can be applied in clustering. Hidden clusters in the data sets are identified by only using those features that are not eliminated. This subsection and the following one will review the feature selection methods used in clustering. It should be noted that all of those studies use the same subset of features in order to identify clusters.

As in clustering problems, feature selection is also taken into consideration within the context of operations research. Optimal subset of features can be obtained with exhaustive search which considers all possible combinations, but it is computationally expensive. Therefore, Narendra and Fukunaga (1977) propose a branch and bound procedure for selecting the subset of features. In this procedure, features are eliminated depending on the predetermined criterion function at each node. In order to search the tree effectively and decrease the computational time, there are several different approaches to branch and bound implementation on feature selection. Yu and Yuan (1993) claim that reducing the calculation of criterion function while searching the tree will decrease the computational time. The proposed method says that criterion function can be calculated only at the leaf node of a path which includes a single branch. Keeping information about the previously eliminated feature sets may reduce the search space by eliminating some of the paths without calculating the criterion function. Therefore, Chen (2003) proposes to keep partial paths which have been already eliminated in previous nodes. The author ignores paths containing at least one of those partial paths before evaluating the criterion function. Somol et al. (2004) use a simpler prediction function instead of the actual criterion function. If the node is eliminated according to prediction function, then the actual criterion function is also calculated. Casasent and Chen (2003) and Nakariyakul and Casasent (2007) propose to start searching the tree from different levels other than the root node.

However, none of those studies deliver a mathematical model for feature selection problem. Mathematical models of clustering problem (Vinod (1969); Rao (1971)) are modified by Benati and García (2014), and distances between data points are calculated depending on the selected subset of features. They define a new decision variable additional to the decision variables used in clustering problem. The new decision variable represents the selection of a feature. Proposed model minimizes the total distance between data points and cluster centers through selected features. The model decides the best subset of features, cluster centers and partitions of the data set into clusters simultaneously. In this study, two linearization methods are used which are direct linearization and radius formulation proposed by García et al. (2011) as an effective method that can be used in solving p-median problem.

2.3.4 Heuristic Approaches for Feature Selection in Clustering

A heuristic algorithm to select features to increase the performance of the clustering algorithms is proposed by Brusco and Cradit (2001). Suppose that one clustering solution is identified using the already selected features and a second solution is obtained by using only one unselected feature, let say j. At each step of the algorithm two clustering solutions are compared using Adjusted Rand Index. A large Adjusted Rand Index suggests that selection of feature j would not worsen the current clustering solution. On the contrary, if the Adjusted Rand Index is small, it can be concluded that feature j should not be added to the set of selected features since it masks the current clustering solution. At each iteration, the feature with the highest Adjusted Rand Index is added to the subset of features.

Brusco (2004) works on clustering problem in the presence of irrelevant features. In order to eliminate those features, a heuristic approach is proposed. The assumptions of the proposed method are that clusters are known in advance and data set should only contain binary features. The heuristic algorithm starts with selecting subset of features evaluated on subset of data points. In the next step, additional features among the remaining ones are tested, and they are added to the subset one at a time by evaluating the clusters obtained from k-means algorithm.

Ólafsson and Yang (2005) use the *Nested Partitions* method described in the subsection which reviews the heuristic algorithms for clustering problems. The method is implemented to feature selection problem by defining feasible region as it contains all set of feature subsets. Rest of the algorithm is following the same idea. Yang and Olafsson (2006) enhance this study by proposing to use sample of features instead of using all features. The backtracking step of the algorithm enables to fix erroneous decisions. Also, in this study, it is suggested that the sample rate can be adjusted dynamically according to the observed frequency of backtrackings. This adjustment eliminates the need to know the optimal sample size in advance.

An exhaustive search algorithm for selecting the subset of features has been proposed by Steinley and Brusco (2008). For all possible subset of features, the best clustering solution is identified by using K-means algorithm with multiple initialization. Then, the proportion of explained variation from the clustering process is computed for each clustering solution, VAF. For each subsets at same size, the subset with maximum VAF is selected as the best solution. The algorithm selects the best subset size, according to a ratio between the reduction in the VAF when subset size is increased from s to $s+1$ and the reduction in VAF when the subset size is increased from $s-1$ to s. The subset of features that produces the maximum ratio is selected as the best.

In their study, Andrews and McNicholas (2014) aim to find the features which simultaneously minimize the variance within cluster and maximize the variance between clusters. Apart from variance, they also consider the correlation between features while selecting the features. Their algorithm starts with calculating the variance on each feature, then those variances are sorted in ascending order. The first feature which has the minimum variance is selected. The algorithm searches features in the ascending order of variances and select the features which have lower correlation than predefined threshold with previously selected features.

Benati et al. (2018) propose two heuristic algorithms for feature selection in clustering. The biggest assumption in their study is that clusters centers are taken as given. Therefore, they aim to find the assignment of each data point to a cluster and relevant features of each cluster. In their first algorithm, they divide the problem into two smaller problems as best-assignment (BA) and best-feature (BF). In the BA, data

points are assigned to the cluster centers through the selected features. BF tried to select best features for each cluster. For finding those features, the total distances over features are sorted in increasing order and the predefined number of features are selected based on this sorted distances. The second algorithm basically adds and drops features from the selected feature set. Algorithm removes a feature, which increases the objective function less, from the set of unselected features and adds it to selected features set. Also, in the following step, algorithm removes a selected feature from the set by considering the maximum decrease in the objective function for a minimization problem.

Xue et al. (2015) provide a detailed review on feature selection used in data mining techniques, and Alelyani et al. (2013) cover feature selection methods specifically used in clustering. Also, nature inspired metaheuristics for feature selection are provided in comprehensive review conducted by Diao and Shen (2015).

Reviewed feature selection methods ignore the fact that selected subset of features may have different significance for each cluster. The study of Frigui and Nasraoui (2004) is the closest study to our problem. They use different subsets of features in order to identify the clusters. But, different than our problem, they assign weights to features instead of selecting the features. Also, in the next section, studies where both data points and features are clustered simultaneously will be delivered.

2.4 Local Clustering and Feature Selection

It is possible that the subset of features relevant to a cluster may not be relevant as well for a different cluster. That is, there may be no common subset of features that can be used to identify all clusters. There are some studies proposed to overcome this problem, and those studies cluster both data points and features simultaneously, which we will call them as local clustering algorithms. Following subsections will summarize these algorithms which are relatively new topics in data mining concept. As a matter of fact, the name of the local clustering and feature selection methods are used interchangeably in the literature which leads to confusion about the types. In this study, we tried to grouped them into two titles, subspace clustering and bi-clustering.

2.4.1 Subspace Clustering

Subspace clustering is a technique which aims to find all clusters within all possible subspaces. There are two major types of subspace clustering depending on the search strategy. Top-down algorithms start with finding an initial clustering using all dimensions and then evaluate the subspaces of each cluster. With the top-down algorithms, partitions of data sets are obtained, that is, each data point is assigned to only one cluster. PROCLUS (Aggarwal et al., 1999) and COSA (Friedman and Meulman, 2004) are the well-known examples of top-down subspace clustering algorithms. Bottom-up algorithms aim to find dense regions in low dimensional spaces and then combine those dense regions to form clusters. In the resulting clusters, one data point can be assigned to different clusters simultaneously. CLIQUE (Agrawal et al., 1998) is one of the bottom-up approaches, and it is also the pioneering study that attempts to find subspace clustering. Both algorithms define similarity of data points as a distance, and data points in the same cluster should be near to each other considering only the subset of features.

2.4.2 Bi-clustering

Contrary to subspace clustering, bi-clustering collects data points which follow a similar behavior into a cluster. Hence, bi-clustering is also named as pattern-based clustering in the literature. Here, the pattern between data points generally relates to the correlations among the features. Bi-clustering allows that a data point or a feature should be able to belong to more than one cluster, to only one cluster, or to no cluster at all. There can be overlapping clusters as in subspace clustering. Bi-clustering algorithms are originally used in the analysis of microarray gene expression data, and Cheng and Church (2000) is the pioneering study of this type. In the literature, there are different approaches on bi-clustering, and they are reviewed on articles written by Pontes et al. (2015) and Padilha and Campello (2017). Kriegel et al. (2009) provide a brief review of both subspace clustering and bi-clustering with their applications.

Consider a data set represented by a matrix, D, where rows denote the data points and columns denote the features. Matrix D is defined by the set of data points, $X =$

$\{x_1, x_2, ..., x_n\}$, and set of features, $Y = \{y_1, y_2, ..., y_n\}$. $D_{IJ} = (I, J)$ represents a submatrix of D which includes only the entries in I and J, where I is a subset of data points X, $I \subseteq X$, and J is a subset of features Y, $J \subseteq Y$. In classical clustering, either cluster of data points, D_{IY}, or cluster of features, D_{XJ}, is identified. However, it is possible that there are multiple D_{IJ} submatrices that can be identified where each cluster meets the predefined similarity criterion, and local clustering algorithms can be used to obtain those submatrices.

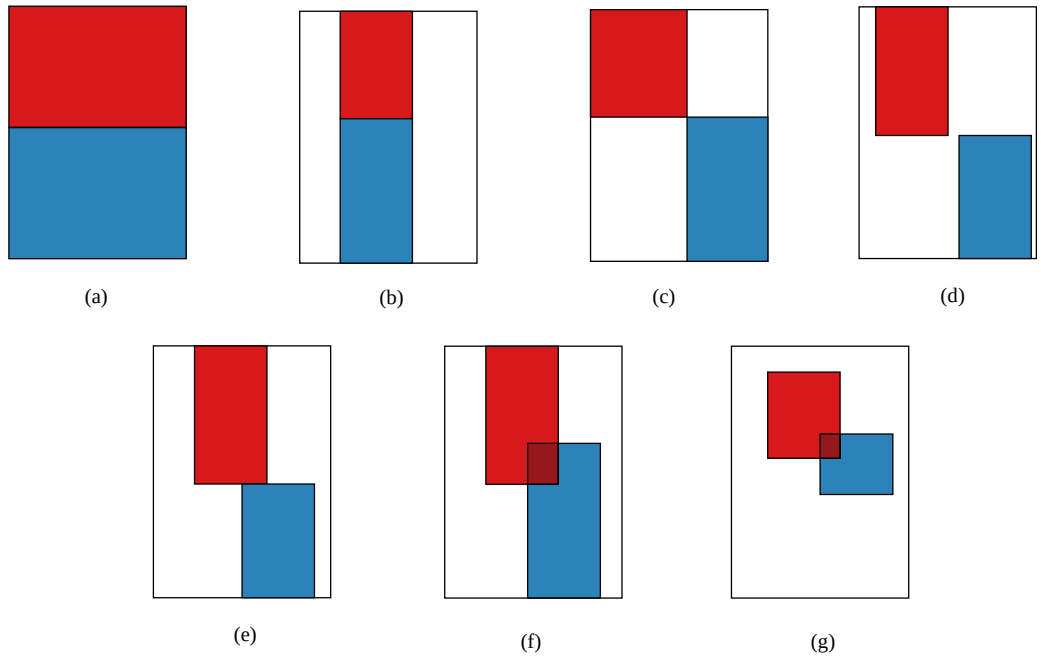

Figure 2.2: Clustering Problems with Different Properties

Figure 2.2 covers the clustering problems with different properties. Classical clustering problem with hard assignment and no feature selection is shown in Figure 2.2 (a). Its extended version with feature selection is in Figure 2.2 (b). In this type, selected features are considered as relevant for both of the clusters. Figure 2.2 (c) represents the local feature selection for hard clustering problem. All features are used by only one of the clusters in this type. However, in Figure 2.2 (d), some of the features are not used. The clustering problem shown in Figure 2.2 (e) allows that features may be used by more than one cluster. The difference between Figure 2.2 (e) and Figure 2.2 (f) is that as well as features data points may be assigned to more than one cluster in the latter. In the last case, Figure 2.2 (g), a data point or a feature can belong to more than one cluster, to only one cluster, or to no cluster at all.

Our problem in this study to feature selection in clustering are similar to subspace clustering and bi-clustering in the sense that we are trying to group data sets into clusters using different subset of features. However, we are not trying to find all cluster in all subspaces. Also, different from the bi-clustering, we construct disjoint clusters where each data point are assigned to only one clusters. Bi-clustering is a type of clustering shown in Figure 2.2 (g), whereas the clustering solution identified by our approaches may be as in Figure 2.2 (b), Figure 2.2 (d), and Figure 2.2 (e).

To sum up, we work on clustering problem with feature selection but it is considered that clusters may lay in different subset of features. Therefore, we will select features specific to the clusters. To the best of our knowledge, we are contributing to the literature by using optimization methods and proposing structured heuristic algorithms to solve clustering problem and cluster based feature selection simultaneously.

The next chapter mainly provides the characteristics of the problem and mathematical model formulation proposed for the solution of the defined problem.

CHAPTER 3

PROBLEM DEFINITION

In Chapter 2, background information about clustering and its variations were introduced, and the related literature both for clustering and feature selection in clustering problems were reviewed. In this chapter, characteristics of the problem considered in this study will be discussed. After defining the problem, proposed mathematical models will be provided.

3.1 Problem Statement

Clustering algorithms try to group alike data points into the same cluster and assign dissimilar ones to different clusters. In this study, each cluster will be represented by a center, and that center will be one of the data points assigned to that cluster. The center is named as *medoid*. Also, each data point will be assigned to only one cluster.

Similarity measure used in clustering can be defined differently depending on the features of the data set. In this study, we only use data sets which contain continuous features. So, the similarity between data points and cluster centers are calculated by using $L_1 - norm$. Let data points v_i and v_j be defined in $\mathbb{R}^m$, here m is equal to $|M|$ where M is the set of features and $|.|$ is used to denote the cardinality of a set. Then, L_1 distance between data point i and data point j will be as follows:

$$d_{ij} = \sum_{k=1}^{m} d_{ijk}, \tag{3.1}$$

$$where \qquad d_{ijk} = \mid v_{ik} - v_{jk} \mid .$$

The main objective will be minimizing the sum of distances between the data points and their cluster centers. Consider a data set where the set of data points and features are shown by N and M, respectively. By assuming all features to be used in clustering, the objective function of the problem will be as follows:

$$minimize \qquad \sum_{p=1}^{P} \sum_{k=1}^{|M|} \sum_{\substack{i=1 \\ i \in \mathscr{C}_p}}^{|N|} d_{ic_pk} \tag{3.2}$$

Here, P is the number of clusters in the problem, and $|M|$ and $|N|$ stand for the number of features and number of data points. $\mathscr{C}_p$ denotes the data points assigned to cluster p and c_p represents the medoid of that cluster. Then, d_{ic_pk} shows the L_1 distance between data point i and cluster center c_p through feature k. Notice that, this objective function tries to minimize the $compactness$ of the clusters.

In this study, the number of clusters is given in advance. If it is not given, the objective function given in (3.2) will be minimized where the number of clusters is equal to the number of data points, $P = |N|$.

Data sets may include irrelevant or redundant features which do not contain useful information to form clusters besides the ones that are relevant. In order to obtain meaningful clusters, relevant features should be selected. Hence, in this study, we consider a clustering problem with feature selection. Selecting the features comes up with the assumption that selected features have the same relevance for all clusters. But, the feature set used to define each cluster might be different. That means, we are constraining feature selection based on the clusters. Assume that Q features are used to define each cluster, and a feature may be relevant to more than one cluster. The objective function of the problem will be as follows:

$$minimize \qquad \sum_{p=1}^{P} \sum_{\substack{q=1 \\ q \in Q_p}}^{|M|} \sum_{\substack{i=1 \\ i \in \mathscr{C}_p}}^{|N|} d_{ic_pq} \tag{3.3}$$

where Q_p denotes the features selected for cluster p. Rest of the notation used in (3.3) are the same with the ones in (3.2).

It is assumed that number of features used to define each cluster will be given in advance, and it is same for all clusters. Note that, the objective function uses the nonnegative terms. Value of the objective function stays the same or increases with an increase in the number of features to be selected. Therefore, without constraining the number of features to be selected, the model will only use one feature for each cluster to minimize the $compactness$ of clusters.

To sum up, our problem is center-based clustering where each data point is assigned to only one of the clusters, and each cluster is represented by one of the data points assigned to that cluster. The number of clusters is predefined in advance. Each cluster is described by a different set of features, and the number of features to be selected is given. Here, it is allowed that the same feature might be relevant for many other clusters. The aim is to minimize the total distance between data points and their cluster centers via selected features. The defined problem is named as *clustering and cluster based feature selection (CBFS)*.

The total number of possible solutions to the described problem with $|N| = n$ data points, $|M| = m$ features, P clusters, and Q relevant features is computed as follows:

$$\left[\frac{1}{P!}\sum_{p=0}^{P}(-1)^p\binom{P}{p}(P-p)^n\right]\left[\binom{m}{Q}^P\right] \tag{3.4}$$

The first term of (3.4) shows the number of ways to partition n data points into P clusters, whereas the second term is the number of ways to select Q features out of m features for P number of clusters. Table 3.1 provides the number of possible solutions for various combinations of n, m, P, and Q. The table reveals that, even for very small-sized problems, the solution space for CBFS is enormous. It shows that a complete enumeration search over all possible cluster centers and relevant features is computationally impractical for large problems.

Table 3.1: The Number of Possible Solutions for Various Combinations of n, m, P, and Q.

n	m	P	Q	possible number of solutions
20	4	2	2	1.887×10^{7}
20	6	2	2	1.180×10^{8}
20	6	2	3	2.097×10^{8}
20	4	3	2	1.254×10^{11}
20	6	3	2	1.960×10^{12}
20	6	3	3	4.645×10^{12}
40	4	2	2	1.979×10^{13}
40	6	2	2	1.237×10^{14}
40	6	2	3	2.199×10^{14}
40	4	3	2	4.377×10^{20}
40	6	3	2	6.839×10^{21}
40	6	3	3	1.621×10^{22}

Consider the well-known facility location problem, namely *p-median* problem. If one thinks that the centers in the clustering problem denote the facilities and data points refer to the customers, then the clustering problem can be seen as a *p-median* problem. But, the customers are defined on the plane, and they have only two features in the facility location problem. However, data points generally have more than two features in the clustering problem. Also, in the facility location problem, there is no discussion on feature selection, both x and y coordinates are important for the customers. In our problem, feature selection is also crucial. If we do not select features for each cluster and we only have two features, then our problem will be a *p-median* problem. Because of the similarity between two problems, mathematical model for the *p-median* problem may serve as a base model for our clustering problem.

In our problem, by using a center-based clustering approach, (i) locations of cluster centers (ii) assignments of the data points to a cluster and (iii) selection of features for each cluster are decided simultaneously. A mixed integer mathematical model is proposed which minimizes the total distance between data points and cluster centers via selected features. The details of the proposed model is given in the next section.

3.2 Mixed Integer Programming Formulations

In this section, mathematical models proposed for the CBFS are delivered. In 3.2.1, the proposed model is given. Since the model is nonlinear, we have used different linearization techniques proposed in the literature. The following two subsections, subsection 3.2.2 and subsection 3.2.3, give details of those linearized models.

3.2.1 A Nonlinear Mixed Integer Model for CBFS: NM

Consider a data set where N represents the set of data points and M is the set of features where each data point is defined. We try to cluster the data sets into p clusters where each cluster is described by q features. We need to find the centers for each cluster, assign data points to a cluster and select the relevant features of the clusters. Since we are using center-based clustering, each cluster has a center and the centers should be selected from the data points.

The objective of the problem is to minimize the total distance between data points and their cluster centers over the selected features of those clusters. The proposed model **NM** will be:

$$(\mathbf{NM}) \quad \text{Minimize} \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in M} d_{ijk} z_{jk} x_{ij} \tag{3.5}$$

$$\text{subject to:} \quad x_{ij} \leq y_j, \qquad \forall i,j \in N \tag{3.6}$$

$$\sum_{j\in N} x_{ij} = 1, \qquad \forall i \in N \tag{3.7}$$

$$\sum_{j\in N} y_j = p, \tag{3.8}$$

$$\sum_{k\in M} z_{jk} = q\, y_j, \qquad \forall j \in N \tag{3.9}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i,j \in N \tag{3.10}$$

$$z_{jk} \in \{0,1\}, \qquad \forall j \in N, \forall k \in M \tag{3.11}$$

$$y_j \in \{0,1\}. \qquad \forall j \in N \tag{3.12}$$

There are three decision variables in this formulation. A binary variable is defined to decide whether a data point j is selected as a cluster center or not, y_j. We have a binary variable which takes 1 if a data point i is assigned to a cluster where data point j is the center of that cluster, x_{ij}. Lastly, z_{jk} is the binary decision variable stating whether feature k is selected for the cluster where data point j is the center.

Constraint (3.6) ensures that data point i can only be assigned to the data point j if data point j is a cluster center. Constraint (3.7) forces data point i to be assigned to exactly one cluster since we are aiming to obtain disjoint clusters. Constraint (3.8) implies that total number of clusters should be equal to p. Constraint (3.9) imposes that if data point j is selected as a cluster center, then there should be q features selected for this cluster. Constraints (3.10), (3.11) and (3.12) state that decision variables x_{ij}, z_{jk} and y_j are binary variables, respectively.

In the Table 3.2, the notation used for mathematical formulations is summarized.

Table 3.2: Notation used for Mathematical Formulations

Sets	
N	Set of data points
M	Set of features
Parameters	
p	Number of clusters that will be constructed
q	Number of features that should be selected for each cluster
d_{ijk}	Distance between data points i and j on feature k, $i \in N$, $j \in N$, $k \in M$
Decision Variables	
y_j	Binary decision variable as 1 if data point j is selected as a cluster center, 0 otherwise, $j \in N$
x_{ij}	Binary decision variable as 1 if data point i is assigned to data point j, 0 otherwise, $i \in N$, $j \in N$
z_{jk}	Binary decision variable as 1 if feature k is selected for cluster center j, 0 otherwise, $j \in N$, $k \in M$

It is observed that when decision variables representing the cluster centers and selected features, y_j and z_{jk}, are binary variables, assignment variables x_{ij} may be relaxed to take continuous values. Due to the constraint (3.6), x_{ij}'s may take the value of at most 1 at the optimal solution. Since we have positive terms in the objective function, it will be minimized when each data point is assigned to only one cluster instead of giving fractional values to x_{ij}'s. Therefore, x_{ij}'s will take integer values at the optimal solution, and constraint (3.10) can be written as $x_{ij} \geq 0, \forall i, j \in N$.

The objective function of the proposed model contains the product of two decision variables, $z_{jk}x_{ij}$. So, the model **NM** has a nonlinear objective function. Following sections introduce linearized models.

3.2.2 Linearized Model 1: LM_1

Li (1994) proposes the following proposition for the product of two decision variables, where one of them is a continuous variable between 0 and 1, and the other is a binary variable. This proposition can be used to reformulate the given nonlinear model **NM**.

Proposition 1. *A polynomial mixed 0-1 term w = zx, where z is a 0-1 variable and x is a continuous variable,* $0 < x \leq 1$*, can be represented by the following linear inequalities: (i)* $w \geq x + z - 1$*; (ii)* $w \leq x$ *; (iii)* $w \leq z$*; (iv)* $w \geq 0$*.*

Proof.
Case 1. Suppose $w = zx$. All inequalities will be satisfied for $z = 0$ or 1 since x variables are continuous variables such that $0 < x \leq 1$.
Case 2. Suppose all inequalities are true. If $z = 0$, then we have $w = 0$ from inequalities (iii) and (iv). If $z = 1$, it forces $w = x$ from inequalities (i) and (ii). It can be concluded that $w = zx$.
Therefore, if and only if $w = zx$, $z = 0$ or 1, and $0 < x \leq 1$, then (i)–(iv) are satisfied. □

By following the Proposition 1, the term $z_{jk}x_{ij}$ in (3.5) can be rewritten as w_{ijk}. The following constraints should be added to the model **NM**.

$$
\begin{aligned}
& w_{ijk} \geq x_{ij} + z_{jk} - 1, && \forall i, j \in N, \forall k \in M \\
& x_{ij} \geq w_{ijk}, && \forall i, j \in N, \forall k \in M \\
& z_{jk} \geq w_{ijk}, && \forall i, j \in N, \forall k \in M \\
& w_{ijk} \geq 0, && \forall i, j \in N, \forall k \in M
\end{aligned}
$$

Then the resulting linearized model (**LM$_1$**) will be as given below.

$$
\begin{aligned}
(\mathbf{LM_1}) \quad \text{Minimize} \quad & \sum_{i \in N} \sum_{j \in N} \sum_{k \in M} d_{ijk} w_{ijk} && (3.13) \\
\text{subject to:} & \\
& (3.6) - (3.9) && \\
& w_{ijk} \geq x_{ij} + z_{jk} - 1, && \forall i, j \in N, \forall k \in M \quad (3.14) \\
& x_{ij} \geq w_{ijk}, && \forall i, j \in N, \forall k \in M \quad (3.15) \\
& z_{jk} \geq w_{ijk}, && \forall i, j \in N, \forall k \in M \quad (3.16) \\
& w_{ijk} \geq 0, && \forall i, j \in N, \forall k \in M \quad (3.17) \\
& x_{ij} \geq 0, && \forall i, j \in N \quad (3.18) \\
& z_{jk} \in \{0, 1\}, && \forall j \in N, \forall k \in M \quad (3.19) \\
& y_j \in \{0, 1\}. && \forall j \in N \quad (3.20)
\end{aligned}
$$

Since constraints (3.6)–(3.9) are the same, they are not written again in this formulation.

There are different conditions for the right hand side value of the constraint (3.14). If any one of the variables x_{ij} or z_{jk} is at their upper bounds, then

$$
\begin{aligned}
\text{if } x_{ij} = 1, \text{then } w_{ijk} \geq z_{jk} \\
\text{if } z_{jk} = 1, \text{then } w_{ijk} \geq x_{ij} \\
\text{if } x_{ij} = 1 \text{ and } \ z_{jk} = 1, \text{then } w_{ijk} \geq 1
\end{aligned}
$$

Since it is a minimization problem,

$$\text{if } x_{ij} = 1, \text{then } w_{ijk} = z_{jk}$$
$$\text{if } z_{jk} = 1, \text{then } w_{ijk} = x_{ij}$$
$$\text{if } x_{ij} = 1 \text{ and } z_{jk} = 1, \text{then } w_{ijk} = 1$$

Note that, it is at most 1 in either cases.

If at least any one of the variables x_{ij} or z_{jk} is 0, then the right hand side does not take positive value. Therefore, w_{ijk} will be 0.

$$\text{if } x_{ij} = 1 \text{ and } z_{jk} = 0, \text{then } w_{ijk} = 0$$
$$\text{if } x_{ij} = 0 \text{ and } z_{jk} = 1, \text{then } w_{ijk} = 0$$
$$\text{if } x_{ij} = 0 \text{ and } z_{jk} = 0, \text{then } w_{ijk} = 0$$

That means, constraints (3.15) and (3.16) are satisfied at every optimal solution, they can be dropped from the formulation. The reduced model will be as follows.

$(\mathbf{LM_1})$ Minimize
$$\sum_{i\in N}\sum_{j\in N}\sum_{k\in M} d_{ijk}w_{ijk} \tag{3.21}$$

subject to:
$$x_{ij} \le y_j, \qquad \forall i,j \in N \tag{3.22}$$
$$\sum_{j\in N} x_{ij} = 1, \qquad \forall i \in N \tag{3.23}$$
$$\sum_{j\in N} y_j = p, \tag{3.24}$$
$$\sum_{k\in M} z_{jk} = q\, y_j, \qquad \forall j \in N \tag{3.25}$$
$$w_{ijk} \ge x_{ij} + z_{jk} - 1, \qquad \forall i,j \in N, \forall k \in M \tag{3.26}$$
$$w_{ijk} \ge 0, \qquad \forall i,j \in N, \forall k \in M \tag{3.27}$$
$$x_{ij} \ge 0, \qquad \forall i,j \in N \tag{3.28}$$
$$z_{jk} \in \{0,1\}, \qquad \forall j \in N, \forall k \in M \tag{3.29}$$
$$y_j \in \{0,1\}. \qquad \forall j \in N \tag{3.30}$$

3.2.3 Linearized Model 2: $\mathbf{LM_2}$

By following the linearization method used by Benati et al. (2018), we propose to use the following Proposition 2 in order to linearize the product of two decision variables:

Proposition 2. *A polynomial mixed 0-1 term w = xz, where both x and z are 0-1 variables, can be represented by the following linear constraints:*

$$(i) \qquad \sum_{k \in M} w_{ijk} = q\, x_{ij}, \qquad \forall i, j \in N$$

$$(ii) \qquad w_{ijk} \leq z_{jk}, \qquad \forall i, j \in N, \forall k \in M$$

$$(iii) \qquad w_{ijk} \in \{0, 1\}, \qquad \forall i, j \in N, \forall k \in M$$

Proof.

Case 1. Suppose $w_{ijk} = z_{jk}x_{ij}$. All inequalities will be satisfied;

if $x_{ij} = 0$, $w_{ijk} = 0$ from (i) and (iii)

if $x_{ij} = 1$, $\sum_{k \in M} w_{ijk} = q$, and $\sum_{k \in M} w_{ijk} \leq \sum_{k \in M} z_{jk}$ from (ii). Since $x_{ij} = 1$ we know that y_j should also be 1. Therefore, (ii) is satisfied.

Case 2. Suppose all inequalities are true. If $z_{jk} = 0$, then we have $w_{ijk} = 0$ from (ii) and (iii). If $z_{jk} = 1$, it forces $w_{ijk} = x_{ij}$ from (i) and (iii). It can be concluded that $w_{ijk} = z_{jk}x_{ij}$.

Therefore, if and only if $w_{ijk} = z_{jk}x_{ij}$, where both x_{ij} and z_{jk}'s are binary, then (i)–(iii) are satisfied. □

Using Proposition 2, the term $z_{jk}x_{ij}$ in (3.5) of model **NM** will be written as w_{ijk} and the given constraints will be added to model. Then, the new linearized model which is called model $\mathbf{LM_2}$ will be as follows.

$(\mathbf{LM_2})$ Minimize $$\sum_{i\in N}\sum_{j\in N}\sum_{k\in M} d_{ijk}w_{ijk} \tag{3.31}$$

subject to:

$(3.6) - (3.9)$

$$\sum_{k\in M} w_{ijk} = q\, x_{ij}, \qquad \forall i,j \in N \tag{3.32}$$

$$w_{ijk} \leq z_{jk}, \qquad \forall i,j \in N, \forall k \in M \tag{3.33}$$

$$w_{ijk} \in \{0,1\}, \qquad \forall i,j \in N, \forall k \in M \tag{3.34}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i,j \in N \tag{3.35}$$

$$z_{jk} \in \{0,1\}, \qquad \forall j \in N, \forall k \in M \tag{3.36}$$

$$y_j \in \{0,1\}. \qquad \forall j \in N \tag{3.37}$$

The first four constraints, (3.6)–(3.9), are the same with the ones in model **NM**. (3.32) ensures that if data point i is assigned to cluster center j, then q number of w_{ijk} should be equal to 1. (3.33) is used for satisfying the condition that w_{ijk} may take positive value if the feature k is selected for the cluster center j. Otherwise, w_{ijk} will be 0.

When the model is analyzed, it is observed that since x_{ij}'s are either 0 or 1, even if y_j is relaxed to take any value between 0 and 1, it will always take 0 or 1 at the optimal solution due to constraint (3.6).

Also, when z_{jk}'s are binary decision variables, the right hand side of (3.33) will be 0 or 1. So, w_{ijk} is bounded by 0 or 1. Since w_{ijk} have nonnegative coefficients in the objective function, and we are solving a minimization problem, w_{ijk} will be equal to 0 or 1 if z_{jk} is equal to 1 instead of taking fractional values.

Based on those observations, the model was updated and the resulting model is provided below.

$$
\begin{aligned}
(\mathbf{LM_2}) \qquad \text{Minimize} \quad & \sum_{i\in N}\sum_{j\in N}\sum_{k\in M} d_{ijk}w_{ijk} && && (3.38)\\
\text{subject to:} \quad & x_{ij} \le y_j, && \forall i,j \in N && (3.39)\\
& \sum_{j\in N} x_{ij} = 1, && \forall i \in N && (3.40)\\
& \sum_{j\in N} y_j = p, && && (3.41)\\
& \sum_{k\in M} z_{jk} = q\, y_j, && \forall j \in N && (3.42)\\
& \sum_{k\in M} w_{ijk} = q\, x_{ij}, && \forall i,j \in N && (3.43)\\
& w_{ijk} \le z_{jk}, && \forall i,j \in N, \forall k \in M && (3.44)\\
& w_{ijk} \ge 0, && \forall i,j \in N, \forall k \in M && (3.45)\\
& x_{ij} \in \{0,1\}, && \forall i,j \in N && (3.46)\\
& z_{jk} \in \{0,1\}, && \forall j \in N, \forall k \in M && (3.47)\\
& y_j \in [0,1]. && \forall j \in N && (3.48)
\end{aligned}
$$

Following the study of Vinod (1969), constraint (3.39) can be rewritten as:

$$\sum_{i\in N} x_{ij} \le n\, y_j, \quad \forall j \in N$$

The logic behind this transformation is that if a data point is selected as a cluster center, then at most all data points may be assigned to that cluster.

The new model will be named as model $\mathbf{LM_3}$. In this formulation, the first constraint is different than the one in model $\mathbf{LM_2}$, but the rest is the same.

$$(\mathbf{LM_3}) \quad \text{Minimize} \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in M} d_{ijk}w_{ijk} \tag{3.49}$$

$$\text{subject to:} \quad \sum_{i\in N} x_{ij} \leq n\, y_j, \qquad \forall j \in N \tag{3.50}$$

$$\sum_{j\in N} x_{ij} = 1, \qquad \forall i \in N \tag{3.51}$$

$$\sum_{j\in N} y_j = p, \tag{3.52}$$

$$\sum_{k\in M} z_{jk} = q\, y_j, \qquad \forall j \in N \tag{3.53}$$

$$\sum_{k\in M} w_{ijk} = q\, x_{ij}, \qquad \forall i,j \in N \tag{3.54}$$

$$w_{ijk} \leq z_{jk}, \qquad \forall i,j \in N, \forall k \in M \tag{3.55}$$

$$w_{ijk} \geq 0, \qquad \forall i,j \in N, \forall k \in M \tag{3.56}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i,j \in N \tag{3.57}$$

$$z_{jk} \in \{0,1\}, \qquad \forall j \in N, \forall k \in M \tag{3.58}$$

$$y_j \in [0,1]. \qquad \forall j \in N \tag{3.59}$$

w_{ijk}'s and y_j's are defined as continuous decision variables. The reason behind this relaxation for w_{ijk}'s is explained during the discussion of model $\mathbf{LM_2}$. However, explanation about why y_j decision variable takes integer values at optimal solution is not obvious in this formulation.

Consider the constraint (3.55) over the features is given below

$$\sum_{k\in M} w_{ijk} \leq \sum_{k\in M} z_{jk}, \qquad \forall i,j \in N \tag{3.60}$$

The left-hand side of the constraint (3.60) is equal to $q\, x_{ij}$ due to (3.54). The right-hand side of the constraint (3.60) is equal to $q\, y_j$ due to (3.53).

$$q\, x_{ij} \leq q\, y_j, \qquad \forall i,j \in N \tag{3.61}$$

Then, inequality (3.61) reduces to:

$$x_{ij} \leq y_j, \qquad \forall i,j \in N \tag{3.62}$$

Since x_{ij}'s are binary decision variables, even if y_j' are continuous, they will always take 0 or 1 at the optimal solution since we are solving a minimization problem.

Table 3.3 shows number of constraints and variables in each linearization method. By writing the first constraint in a compact form in model **LM$_3$**, we are reducing total number of inequality constraints by $n^2 - n$ comparing to models **LM$_1$** and **LM$_2$**. Model **LM$_1$** has n^2 less equality constraints than models **LM$_2$** and **LM$_3$**. When it comes to number of decision variables, there are $n^2 - n$ less binary variables in model **LM$_1$** than models **LM$_2$** and **LM$_3$**, but model **LM$_1$** has $n^2 - n$ more continuous variables than the other models.

Table 3.3: Differences between Linearized Models

	Number of Constraints		**Number of Variables**	
Model	*Equality*	*Inequality*	*Binary*	*Continuous*
LM$_1$	$2\,n + 1$	$n^2\,(1 + m)$	$n\,(1 + m)$	$n^2\,(1 + m)$
LM$_2$	$n\,(2 + n) + 1$	$n^2\,(1 + m)$	$n\,(n + m)$	$n\,(1 + n\,m)$
LM$_3$	$n\,(2 + n) + 1$	$n\,(1 + n\,m)$	$n\,(n + m)$	$n\,(1 + n\,m)$

The performances of the proposed nonlinear model **NM** and linearized models **LM$_1$**, **LM$_2$**, and **LM$_3$** are tested on the simulated data sets. The details are given in the next chapter.

CHAPTER 4

COMPUTATIONAL RESULTS AND COMPARISON OF PROPOSED MODELS

In this chapter, results of the experimental studies conducted on proposed mathematical models will be provided. In Section 4.1, we describe the simulated data sets used in this study. In Section 4.2, performance measures used in evaluation of the proposed mathematical models are described. Comparison of those mathematical models is delivered in Section 4.3.

4.1 Simulated Data Sets

In this study, we use simulated data sets to compare the performance of the proposed models. We create data sets of different size which include relevant and irrelevant features. Table 4.1 shows the characteristics of the simulated data sets. Here, it can be said that given a data set with n data points, number of clusters (p), total number of features (m), and number of relevant features (q) are changing. Data sets include 2, 3, and 4 clusters, and the number of features in each data set will be {4, 5, 6, 8, 10, 12} where some of these features are relevant. Also, the relation between number of features and number of relevant features will be $q < m$, where q and m represent number of relevant features and total number of features in the data set, respectively. Therefore, for a given data set in size n, 60 problem instances in different settings are generated.

For every data set with a given number of data points, out of m features q of them are selected as relevant. We are generating the data sets using multivariate normal distribution for the relevant features. Here, each cluster has a different mean. For

Table 4.1: Details of the Simulated Data Sets

Number of Data Points (n)	40, 50, 80, 100, 200
Number of Clusters (p)	2, 3, 4
Number of Features (m)	4, 5, 6, 8, 10, 12
Number of Relevant Features (q)	2, 3, 4, 6

example, if two clusters will be generated when q is equal to two, cluster means will be located at [0, 0] and [5, 5]. Table 4.2 shows the mean of each cluster. In that notation, $\overrightarrow{\mathbf{1}_q}$ refers to a vector of ones with q entries. As given in the same table, variance-covariance matrix is always set to identity matrix which is symmetrical and positive definite for each cluster. By this way, spherical clusters will be generated. Note that, when creating the data sets, it is assumed that clusters have equal sizes.

Table 4.2: Parameters of Multivariate Normal Distribution for Clusters, q = Number of Relevant Features

Cluster Number	Parameters of the Distribution for the Cluster
1	$(\mu_1, \Sigma_1)=(0\ \overrightarrow{\mathbf{1}_q}, \mathbf{I}_q)$
2	$(\mu_2, \Sigma_2)=(5\ \overrightarrow{\mathbf{1}_q}, \mathbf{I}_q)$
3	$(\mu_3, \Sigma_3)=(-7\ \overrightarrow{\mathbf{1}_q}, \mathbf{I}_q)$
4	$(\mu_4, \Sigma_4)=(11\ \overrightarrow{\mathbf{1}_q}, \mathbf{I}_q)$

Remaining features of data set, $m - q$ features, will be named as irrelevant features. They are created from Uniform distribution. For all irrelevant features, the distribution parameters will be as Uniform[0, 20], Uniform[0, 10], and Uniform[0, 5].

If we geometrically interpret the generated data sets, multivariate normal distribution will generate dense data sets along relevant features but data points will be scattered over irrelevant features due to Uniform Distribution.

4.2 Performance Measures

Proposed mathematical models are compared by using different performance measures which are percent gaps from made objective, optimal solution, and best available solution, number of times the optimal solution found, number of times the best available solution found, computational time and number of times the model hits to the time limit. Details of those measures are described below and their abbreviations are provided between parentheses.

Percent gap from made objective ($\%Gap_M$): Data sets are generated using specific distributions. If we assume that generated clusters and their corresponding relevant features are taken as they are, we need to only know cluster centers in order to minimize total distance between cluster center and data points in that cluster. Made objective of the simulated data set is calculated by using the Equation 3.3 given in Chapter 3 after finding the cluster centers. Total L_1 distance of the generated data set will be denoted as Z_M. If we denote the objective function of any mathematical model with Z, then $\%Gap_M$ will be calculated as follows:

$$\%Gap_M = (\frac{Z - Z_M}{Z_M})\,100 \tag{4.1}$$

It may take negative values because we have a strong assumption that generated clusters are perfectly separable and they will be obtained by clustering solution. Negative gap means that proposed models found better clusters than the generated ones in terms of compactness.

Percent gap from optimal solution ($\%Gap_O$): It is the optimality gap which is directly obtained from the solver. It is calculated by considering the objective function of the current solution Z_C and the best possible objective function Z_P that can be obtained by searching through unexplored nodes.

$$\%Gap_O = (\frac{Z_C - Z_P}{Z_P})\,100 \tag{4.2}$$

Percent gap from best available solution ($\%Gap_B$): It is the performance measure that takes into account the best available objective function value Z_B obtained from any of the mathematical model. The measure can be calculated as follows where Z denotes the objective function value of a specific model:

$$\%Gap_B = (\frac{Z - Z_B}{Z_B})\,100 \tag{4.3}$$

Number of times the optimal solution found ($NOpt$): We will explicitly report how many times a mathematical model finds the optimal solution out of 60 problem instances for a given number of data points n.

Number of times the best available solution found ($NBest$): The best available solution may be obtained by more than one mathematical models. This performance measure is actually the summary of the performance measure $\%Gap_B$. We will explicitly report how many times a mathematical model find the best available solution out of 60 problem instances for a given number of data points n.

Computational time ($CPU\ s.$): It represents the solution time of the proposed model in seconds. It should be noted that the time limit for each mathematical model to solve the problem is specified as 7200 seconds. The data sets that cannot be solved within the time limit is stated as TL in the corresponding column.

Number of times the model hits time limit (NTL): This performance measure report the number of problem instances that cannot be solved optimally by a mathematical model within the time limit. Data sets which include same number of data points n will be considered together. That means, it is out of 60 as in $NBest$.

4.3 Computational Results of Proposed Methods for Simulated Data

Experimental studies were carried out on 64-bit Windows 10 PC with 3.0 GHz eight-core AMD Ryzen 7 1700 processor and 16 GB RAM. All solution methods were coded in C++. We compiled codes under Concert Library of CPLEX 12.7.1 on Microsoft Visual Studio 2017.

Proposed mathematical models are tested on simulated data sets. Results of the experiments conducted on a data set with 40 data points are provided in Tables 4.3–4.5. The results for the data sets which include 50, 80, 100, and 200 data points are given in Tables A.1–A.12 provided in Appendix A.

Parameter settings of simulated data sets are given in the first three columns of tables where the number of clusters, number of features and number of relevant features are represented as p, m and q. For example, results of experimental studies conducted on the data set with 40 data points, and 10 features where two of them will be selected for the case of two clusters ($p2m10q2$) can be found in the 13$^{\text{th}}$ row of Table 4.3. For each of the mathematical models, there are four columns showing the performance measures, $\%Gap_M$, $\%Gap_O$, $\%Gap_B$ and $CPU\ s.$. At the last row of each table, averages of performance measures will be provided. Average of $CPU\ s.$ is calculated by only considering the times less than the given time limit. In some of the tables, there is a symbol "-" for the averages of computational times. It means that none of the problem instances with n number of data points and p clusters can be solved within the time limit by a corresponding mathematical model, so the average is not calculated.

Averages of those performance measures are also summarized in Table 4.6 by grouping data sets with equal number of data points. In this table, there are two additional columns for the performance measures $NBest$ and NTL. Looking at only averages on $CPU\ s.$ may be misleading because of the number of terms considered in the calculation. Here, NTL will be helpful to come up with a conclusion about computation times of mathematical models.

In most of the data sets, nonlinear model **NM** cannot solve the problem within the given time limit. Usually, when the size of the data set increases, the performance measure showing the number of problem instances that cannot be solved within a time limit, NTL, also reports higher values. For example, almost half of the problem instances can be solved within 7200 seconds when the data set include only 50 data points. However, this value decreases sharply, and we can only solve eight problem instances out of 60 by using model **NM** for a data set with 100 data points.

Table 4.3: Results of Experimental Studies for Simulated Data Sets with 40 Data Points and 2 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
2	4	2	0	0	0	2731.27	0	0	0	41.72	0	0	0	3.90	0	0	0	3.61
2	4	3	0	0	0	13.72	0	0	0	11.22	0	0	0	1.19	0	0	0	1.00
2	5	2	0	0	0	3007.62	0	0	0	55.49	0	0	0	2.32	0	0	0	1.99
2	5	3	0	0	0	1274.29	0	0	0	51.72	0	0	0	3.70	0	0	0	2.70
2	5	4	0	0	0	13.85	0	0	0	18.15	0	0	0	2.28	0	0	0	1.48
2	6	2	0.42	40.98	1.38	TL	-0.95	0	0	124.21	-0.95	0	0	4.18	-0.95	0	0	3.16
2	6	3	0	0	0	4127.31	0	0	0	80.60	0	0	0	2.42	0	0	0	2.01
2	6	4	0	0	0	153.46	0	0	0	52.87	0	0	0	5.12	0	0	0	4.21
2	8	2	-0.02	73.09	0.35	TL	-0.37	0	0	327.08	-0.37	0	0	6.49	-0.37	0	0	5.69
2	8	3	3.48	77.02	3.48	TL	0	0	0	353.56	0	0	0	9.00	0	0	0	7.13
2	8	4	0	28.41	0	TL	0	0	0	185.73	0	0	0	8.17	0	0	0	6.01
2	8	6	-4.56	0	0	173.02	-4.56	0	0	118.84	-4.56	0	0	14.06	-4.56	0	0	10.69
2	10	2	0.52	79.68	0.52	TL	0	0	0	750.70	0	0	0	10.92	0	0	0	6.23
2	10	3	0.48	77.18	0.48	TL	0	0	0	1499.59	0	0	0	53.98	0	0	0	72.96
2	10	4	0	79.31	0	TL	0	0	0	1495.21	0	0	0	30.92	0	0	0	28.03
2	10	6	0	18.76	0	TL	0	0	0	173.54	0	0	0	14.01	0	0	0	11.20
2	12	2	5.33	81.63	5.33	TL	0	0	0	1291.89	0	0	0	18.67	0	0	0	14.88
2	12	3	-0.74	84.51	3.76	TL	-4.34	0	0	2783.18	-4.34	0	0	114.57	-4.34	0	0	110.58
2	12	4	2.43	89.57	2.43	TL	0	45.20	0	TL	0	0	0	123.33	0	0	0	110.35
2	12	6	-1.64	88.31	0.13	TL	-1.76	0	0	3757.71	-1.76	0	0	61.91	-1.76	0	0	51.70
Average			0.28	40.92	0.89	1436.82	-0.60	2.26	0	693.31	-0.60	0	0	24.55	-0.60	0	0	22.78

Table 4.4: Results of Experimental Studies for Simulated Data Sets with 40 Data Points and 3 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
3	4	2	0	0	0	337.80	0	0	0	380.32	0	0	0	2.90	0	0	0	2.02
3	4	3	0	0	0	64.28	0	0	0	63.08	0	0	0	1.37	0	0	0	1.08
3	5	2	-7.07	0	0	1376.53	-7.07	0	0	1596.79	-7.07	0	0	5.99	-7.07	0	0	4.55
3	5	3	0	0	0	390.91	0	0	0	288.70	0	0	0	2.40	0	0	0	2.04
3	5	4	0	0	0	89.98	0	0	0	125.45	0	0	0	2.70	0	0	0	1.73
3	6	2	-6.19	29.47	0	TL	-6.19	0	0	5939.53	-6.19	0	0	24.08	-6.19	0	0	21.72
3	6	3	-2.17	0	0	3067.34	-2.17	0	0	4455.76	-2.17	0	0	4.85	-2.17	0	0	5.27
3	6	4	0	0	0	248.88	0	0	0	504.40	0	0	0	2.98	0	0	0	2.46
3	8	2	-5.90	100.00	0	TL	-5.90	100.00	0	TL	-5.90	0	0	63.22	-5.90	0	0	59.19
3	8	3	0	80.83	0	TL	0	100.00	0	TL	0	0	0	18.31	0	0	0	15.43
3	8	4	0	78.61	0	TL	0	69.04	0	TL	0	0	0	10.95	0	0	0	10.44
3	8	6	0	0	0	187.02	0	0	0	252.96	0	0	0	6.47	0	0	0	4.48
3	10	2	-24.07	100.00	0	TL	-24.07	100.00	0	TL	-24.07	0	0	397.99	-24.07	0	0	1327.47
3	10	3	-0.35	100.00	0	TL	-0.35	100.00	0	TL	-0.35	0	0	683.38	-0.35	0	0	587.33
3	10	4	0	100.00	0	TL	0	100.00	0	TL	0	0	0	26.89	0	0	0	20.36
3	10	6	-4.16	65.87	0	TL	-4.16	65.37	0	TL	-4.16	0	0	15.23	-4.16	0	0	11.95
3	12	2	-24.42	100.00	0.85	TL	-24.85	100.00	0.27	TL	-25.05	10.63	0	TL	-23.91	12.22	1.52	TL
3	12	3	-3.57	100.00	0	TL	-3.57	100.00	0	TL	-3.57	0	0	1281.07	-3.57	0	0	3400.82
3	12	4	-3.66	100.00	0.07	TL	-3.73	100.00	0	TL	-3.73	0	0	198.79	-3.73	0	0	134.70
3	12	6	-2.50	100.00	0	TL	-2.50	100.00	0	TL	-2.50	0	0	30.50	-2.50	0	0	34.76
Average			-4.20	52.74	0.05	720.34	-4.23	51.72	0.01	1511.89	-4.24	0.53	0	146.32	-4.18	0.61	0.08	297.25

Table 4.5: Results of Experimental Studies for Simulated Data Sets with 40 Data Points and 4 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
4	4	2	0	70.33	0	TL	0	55.07	0	TL	0	0	0	3.79	0	0	0	3.22
4	4	3	-1.04	0	0	460.11	-1.04	0	0	336.39	-1.04	0	0	1.99	-1.04	0	0	1.47
4	5	2	-0.29	100.00	0	TL	-0.29	100.00	0	TL	-0.29	0	0	6.02	-0.29	0	0	4.76
4	5	3	-4.97	74.32	0	TL	-4.97	74.55	0	TL	-4.97	0	0	2.83	-4.97	0	0	2.26
4	5	4	-1.39	0	0	371.67	-1.39	0	0	336.77	-1.39	0	0	2.06	-1.39	0	0	1.84
4	6	2	0	100.00	0	TL	0	100.00	0	TL	0	0	0	8.96	0	0	0	7.60
4	6	3	0	100.00	0	TL	0	100.00	0	TL	0	0	0	5.23	0	0	0	4.39
4	6	4	0	62.55	0	TL	0	58.26	0	TL	0	0	0	3.41	0	0	0	2.34
4	8	2	-27.72	100.00	0	TL	-27.72	100.00	0	TL	-27.72	0	0	114.12	-27.72	0	0	123.37
4	8	3	-7.42	100.00	0	TL	-7.42	100.00	0	TL	-7.42	0	0	28.13	-7.42	0	0	24.12
4	8	4	-1.12	100.00	0	TL	-1.12	100.00	0	TL	-1.12	0	0	11.66	-1.12	0	0	11.26
4	8	6	-0.55	38.50	0	TL	-0.55	0	0	6655.43	-0.55	0	0	6.20	-0.55	0	0	5.20
4	10	2	-29.49	100.00	2.00	TL	-30.88	100.00	0	TL	-30.88	0	0	6211.23	-30.88	0	0	4939.73
4	10	3	0.39	100.00	2.07	TL	0.84	100.00	2.52	TL	-1.64	0	0	1946.74	-1.64	0	0	1846.36
4	10	4	-13.25	100.00	0.01	TL	-12.94	100.00	0.37	TL	-13.26	0	0	33.26	-13.26	0	0	28.06
4	10	6	0	100.00	0	TL	0.60	100.00	0.60	TL	0	0	0	11.90	0	0	0	9.30
4	12	2	-45.89	100.00	1.74	TL	-46.44	100.00	0.71	TL	-46.42	13.47	0.75	TL	-46.82	10.22	0	TL
4	12	3	-5.75	100.00	0	TL	-4.75	100.00	1.06	TL	-5.75	0	0	880.73	-5.75	0	0	1184.18
4	12	4	12.63	100.00	13.59	TL	1.94	100.00	2.81	TL	-0.84	0	0	1100.53	-0.84	0	0	709.58
4	12	6	-2.62	100.00	2.15	TL	-0.09	100.00	4.80	TL	-4.67	0	0	50.88	-4.67	0	0	41.49
Average			-6.43	82.28	1.08	415.89	-6.81	79.39	0.64	2442.87	-7.40	0.67	0.04	548.93	-7.42	0.51	0	471.08

Table 4.6: Summary of Performance Measures of Proposed Mathematical Models on Simulated Data Sets

Methods	$n = 40$							$n = 50$						
	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$NOpt$	$NBest$	$CPU\ s.$	NTL	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$NOpt$	$NBest$	$CPU\ s.$	NTL
NM	-3.45	58.65	0.67	18	43	1004.95	42	-1.75	52.05	0.75	26	48	1637.95	34
LM$_1$	-3.88	44.46	0.22	31	52	1100.28	29	-1.69	54.68	0.80	24	47	1377.45	36
LM$_2$	-4.08	0.40	0.01	58	59	236.22	2	-2.31	1.72	0.17	53	54	223.43	7
LM$_3$	-4.07	0.37	0.03	58	59	259.55	2	-2.38	1.67	0.10	53	56	286.62	7

Methods	$n = 80$							$n = 100$						
	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$NOpt$	$NBest$	$CPU\ s.$	NTL	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$NOpt$	$NBest$	$CPU\ s.$	NTL
NM	2.98	77.40	4.74	11	21	1596.54	49	12.55	84.34	14.15	8	12	2612.52	52
LM$_1$	3.98	78.24	5.76	11	20	1907.51	49	11.27	84.65	12.90	8	13	2810.82	52
LM$_2$	-1.58	3.51	0.13	49	57	367.35	11	-0.83	4.27	0.50	46	51	658.94	14
LM$_3$	-1.20	3.93	0.51	49	50	366.31	11	-1.19	3.98	0.15	47	56	672.47	13

Methods	$n = 200$						
	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$NOpt$	$NBest$	$CPU\ s.$	NTL
NM	24.83	97.39	12.65	1	14	5562.06	59
LM$_1$	26.70	98.41	0	14.34	10	-	60
LM$_2$	42.42	32.65	24.23	13	22	2473.95	47
LM$_3$	42.36	28.64	24.37	19	32	2153.03	41

Among linearized models, model **LM$_1$** performs poorly comparing with the others. By looking at $CPU\ s.$, model **LM$_1$** takes more time to solve the problem optimally, if it can within the time limit. As reported in Tables 4.3–4.5, the solver cannot state optimality for nine data sets out of 60. For example, even if model **LM$_1$**'s performance in terms of $\%Gap_M$ and $\%Gap_B$ are the same with models **LM$_2$** and **LM$_3$**, its gap from optimality $\%Gap_O$ is stated as approximately 45% in data set with two clusters four relevant features among 12 features ($p2m12q4$).

It can be said that the performance of model **LM$_1$** is similar to the performance of model **NM**. They both cannot solve many of the problem instances within the given time limit. Although their percent gap from optimal solution is high comparing to other two models, they actually ended up at the best available solution in most of the problem instances for smaller data sets. For example, model **LM$_1$** cannot solve 29 problem instances within time limit where data sets include 40 data points. However, when we look at the column showing $NBest$, model **LM$_1$** reports the same objective function with other two linearized models in 52 out of 60 instances, but with reported optimality gap.

If we analyze the performances of those two models deeply, one may see that model **NM** even takes less time on the average for the problems which it can solve within time limit. Also, its performance measures for the percent gaps are slightly worse than the model $\mathbf{LM_1}$ for the larger data sets. Also, there is a symbol "-" in the row of model $\mathbf{LM_1}$ for the averages of computational times where data set includes 200 data points since none of the problem instances can be solved within time limit.

When it comes to the performance of models $\mathbf{LM_2}$ and $\mathbf{LM_3}$, we can say that the problem size affects those two models as well. If the data set includes less number of data points, optimal solutions can be found in almost all problem instances by both of these models. When data sets get larger in terms of data points n, they cannot find the optimal solution in more of the problem instances, but they are still the ones that find the best available solution among the proposed four mathematical models.

The performances of those two models is comparable for a small data sets. For example, when data points in a data set is equal to 50, it looks like model $\mathbf{LM_2}$ performs better than model $\mathbf{LM_3}$ in less time, $CPU\ s.$, with a small sacrifice in the performance about percent gaps. However, model $\mathbf{LM_2}$ cannot state optimally in seven problem instances. Therefore, this will not be a realistic comparison. In this situation, it is obvious that model $\mathbf{LM_3}$ is better than model $\mathbf{LM_2}$. If we look at the data sets with 80 data points, it can be seen that model $\mathbf{LM_2}$ performs far better than $\mathbf{LM_3}$ with a one second increase in computational time on the average. The difference between the performances of those two models can be easily seen when data gets larger. Table 4.6 shows that model $\mathbf{LM_3}$ outperforms model $\mathbf{LM_2}$ in larger data sets.

Experiments show that the problem is getting harder to solve if the number of data points n increases. The same relation was also observed for the number of clusters p and number of features m. When we compare data sets with higher p and m values, $CPU\ s.$ to solve the problem increases as well. On the contrary, computational time is inversely related to the number of features to be selected q. If q increases, problem gets easier, and $CPU\ s.$ decreases.

To sum up, when we compare performances of mathematical models, it is observed that base model **NM** and our first linearized model $\mathbf{LM_1}$ cannot solve the problem within time limits for most of the data instances, whereas models $\mathbf{LM_2}$ and $\mathbf{LM_3}$

will be useful to find optimal solutions when data set includes small number of data points. In order to see whether the mean difference of the objective functions obtained by different approaches is significantly different from zero or not, we apply paired-t test at 5% significance level. From this test, it is observed that there is no significant difference between $\mathbf{LM_2}$ and $\mathbf{LM_3}$, whereas they are significantly different than the other two mathematical models. Among those two models, we will continue with model $\mathbf{LM_3}$ because it performs not significantly but slightly better than model $\mathbf{LM_2}$.

In this thesis, we are also proposing heuristic algorithms to solve the problem in a reasonable time with comparable results. In the next chapters, details of these heuristic algorithms will be delivered.

CHAPTER 5

BENDERS DECOMPOSITION OF THE MODEL AND A HEURISTIC SOLUTION APPROACH

Experimental studies show that proposed mathematical models perform poorly in terms of computational time especially when the size of the data set is larger. In this study, a well-known decomposition algorithm Benders Decomposition is applied to our problem. Also, a heuristic algorithm has been developed by following the decomposition approach. The details of the proposed algorithms will be delivered in this chapter.

5.1 Benders Decomposition of the Model

Benders Decomposition (BD) is a well-known solution method used for solving optimization problems. In (BD) approach, the problem is partitioned into smaller problems instead of solving the large-scale problem for all decision variables using all constraints. These smaller problems are called $master\ problem$ and $subproblem$, where those problems includes subsets of decision variables and subsets of constraints of the main problem. The method starts with solving the master problem, and the remaining decision variables are determined by the dual of the subproblem with the fixed values of the decision variables of master problem. With the solution of dual problem, a new constraint is generated which is called "Benders cut" and added to the master problem, which will be solved again. The stages of the decomposition method is also represented in Figure 5.1.

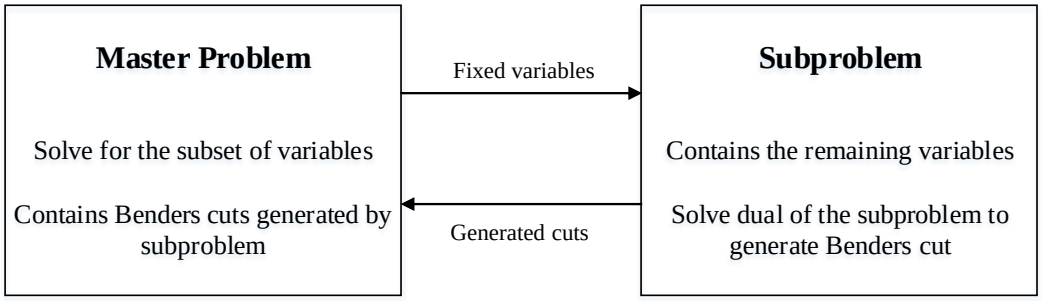

Figure 5.1: Stages of the Benders Decomposition Approach

Dual of the subproblem is solved to eliminate solutions which are worse than the current solution. The method is beneficial if the subproblem is a linear programming model. Considering our mixed integer programming models, it is seen that (BD) method can be suitable to solve our clustering problem. The method is applied on our first linearized model $\mathbf{LM_1}$ which is again provided below for the sake of completeness.

$$\textbf{(LM}_\textbf{1}\textbf{)} \quad \text{Minimize} \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in M} d_{ijk}w_{ijk} \tag{5.1}$$

$$\text{subject to:} \quad x_{ij} \leq y_j, \qquad \forall i,j \in N \tag{5.2}$$

$$\sum_{j\in N} x_{ij} = 1, \qquad \forall i \in N \tag{5.3}$$

$$\sum_{j\in N} y_j = p, \tag{5.4}$$

$$\sum_{k\in M} z_{jk} = q\, y_j, \qquad \forall j \in N \tag{5.5}$$

$$w_{ijk} \geq x_{ij} + z_{jk} - 1, \qquad \forall i,j \in N, \forall k \in M \tag{5.6}$$

$$w_{ijk} \geq 0, \qquad \forall i,j \in N, \forall k \in M \tag{5.7}$$

$$x_{ij} \geq 0, \qquad \forall i,j \in N \tag{5.8}$$

$$z_{jk} \in \{0,1\}, \qquad \forall j \in N, \forall k \in M \tag{5.9}$$

$$y_j \in \{0,1\}. \qquad \forall j \in N \tag{5.10}$$

The set of decision variables and constraints of model $\mathbf{LM_1}$ are partitioned into two. The master problem contains binary decision variables which indicate whether a data

point j is a cluster center or not, y_j, and if feature k is selected for a data point j, z_{jk}. The remaining two continuous decision variables will be solved in the subproblem.

The model $\mathbf{LM_1}$ can be rewritten in terms of variables y_j and z_{jk} as follows and the resulting mathematical model will be named as **MP**:

Master Problem

$$(\mathbf{MP}) \quad \text{Minimize} \quad 0 + \xi \tag{5.11}$$

$$\text{subject to:} \quad \sum_{j \in N} y_j = p, \tag{5.12}$$

$$\sum_{k \in M} z_{jk} = q\, y_j, \qquad \forall j \in N \tag{5.13}$$

$$z_{jk} \in \{0,1\}, \qquad \forall j \in N, \forall k \in M \tag{5.14}$$

$$y_j \in \{0,1\}. \qquad \forall j \in N \tag{5.15}$$

where ξ is the optimal objective function value of the subproblem **SP** given below. Model **MP** decides the location of cluster centers and relevant features of each cluster, y_j and z_{jk} respectively. In the formulation of **SP**, optimal decision variables of the model **MP** are fixed, and they are denoted as $\bar{y}_j$ and $\bar{z}_{jk}$. Assignments of data points are decided in **SP**.

Subproblem

$$(\mathbf{SP}) \quad \text{Minimize} \quad \sum_{i \in N} \sum_{j \in N} \sum_{k \in M} d_{ijk} w_{ijk} \tag{5.16}$$

$$\text{subject to:} \quad -x_{ij} \geq -\bar{y}_j, \qquad \forall i, j \in N \tag{5.17}$$

$$\sum_{j \in N} x_{ij} = 1, \qquad \forall i \in N \tag{5.18}$$

$$w_{ijk} - x_{ij} \geq \bar{z}_{jk} - 1, \qquad \forall i, j \in N, \forall k \in M \tag{5.19}$$

$$w_{ijk} \geq 0, \qquad \forall i, j \in N, \forall k \in M \tag{5.20}$$

$$x_{ij} \geq 0. \qquad \forall i, j \in N \tag{5.21}$$

By strong duality theory, the optimal objective function value of a primal solution can also be obtained by solving its dual. Let denote variables α_{ij}, β_i, and γ_{ijk} as the dual variables of the constraints (5.17), (5.18), and (5.19), respectively. The dual of the subproblem $\mathbf{D_{SP}}$ is as follows.

Dual of the Subproblem

$$(\mathbf{D_{SP}}) \quad \text{Maximize} \quad -\sum_{i\in N}\sum_{j\in N}\bar{y}_j\alpha_{ij} + \sum_{i\in N}\beta_i + \sum_{i\in N}\sum_{j\in N}\sum_{k\in M}(\bar{z}_{jk}-1)\gamma_{ijk} \tag{5.22}$$

$$\text{subject to:} \quad -\alpha_{ij} + \beta_i - \sum_{k\in M}\gamma_{ijk} \leq 0, \qquad \forall i,j \in N \tag{5.23}$$

$$\gamma_{ijk} \leq d_{ijk}, \qquad \forall i,j \in N, \forall k \in M \tag{5.24}$$

$$\alpha_{ij} \geq 0, \qquad \forall i,j \in N \tag{5.25}$$

$$\gamma_{ijk} \geq 0, \qquad \forall i,j \in N, \forall k \in M \tag{5.26}$$

$$\beta_i \; urs. \qquad \forall i \in N \tag{5.27}$$

Comparing to the feasible region of the subproblem **SP**, feasible region of the dual problem does not depend on the values of the decision variables y_j and z_{jk}, only the objective function will be affected. Also, notice that model $\mathbf{D_{SP}}$ always has a feasible solution since the origin is in the feasible region. Therefore, there is no feasibility issue. For a given center and features, it is always possible to find the assignments. In each iteration, with the solution of $\mathbf{D_{SP}}$, generated "Benders cut" will be as follows. Here, $\bar{\alpha}_{ij}$, $\bar{\beta}_i$, and $\bar{\gamma}_{ijk}$ show optimal values of α_{ij}, β_i, and γ_{ijk} variables, respectively.

$$\xi \geq -\sum_{i\in N}\sum_{j\in N} y_j\bar{\alpha}_{ij} + \sum_{i\in N}\bar{\beta}_i + \sum_{i\in N}\sum_{j\in N}\sum_{k\in M}(z_{jk}-1)\bar{\gamma}_{ijk}. \tag{5.28}$$

Steps of the Benders Decomposition method is given below in detail.

Step 1. Solve model **MP** to obtain lower bound to the model $\mathbf{LM_1}$.

If model **MP** is infeasible, conclude that the original model $\mathbf{LM_1}$ is also infeasible. STOP the process.

If model **MP** is feasible, set its objective function as the lower bound, LB, then go to **Step 2**.

Step 2. Solve model $\mathbf{D_{SP}}$. The optimal solution of the model $\mathbf{D_{SP}}$ will be an upper bound to the optimal solution of model $\mathbf{LM_1}$, set it as UB.

If $\mid UB - LB \mid \leq \epsilon$ where ϵ is a predefined small threshold value, then STOP.

Otherwise, generate a new cut for the model **MP**. Add Inequality (5.28) to the constraint set of model **MP** and go to **Step 1**.

Steps of the approach can also be seen in Figure 5.2.

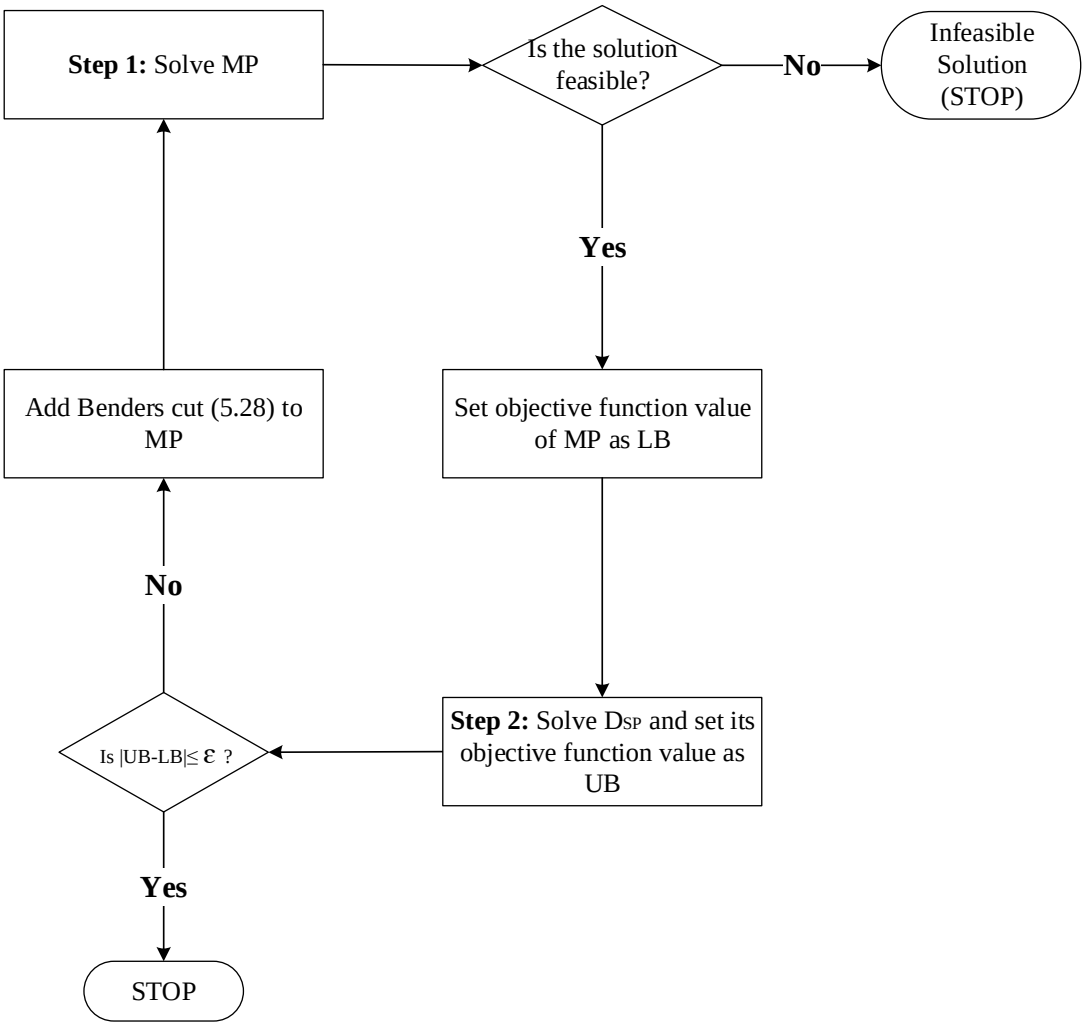

Figure 5.2: Flowchart of the Benders Decomposition Approach

The given solution approach is tested on our simulated data sets but the results cannot compete with the performance of our linearized models. Therefore, we add the following valid inequalities to the subproblem in order to have more information in the cut.

$$-\sum_{i\in N}\sum_{k\in M} w_{ijk} \geq -|N|\, q\, \bar{y}_j, \qquad \forall j \in N \tag{5.29}$$

$$-\sum_{i\in N} w_{ijk} \geq -|N|\, \bar{z}_{jk}, \qquad \forall j \in N, \forall k \in M \tag{5.30}$$

Inequality (5.29) states that if data point j is selected as a cluster center, then at most $|N|\,q$ number of the w_{ijk}'s may take the value of 1 since for each cluster center we will use q features and at most all data points may be assigned to that cluster. The same logic is true for (5.30). If feature k is used for data point j where it is selected as a cluster center, then at most $|N|$ number of w_{ijk}'s may take the value of 1 since at most all data points may use feature k in cluster where the center is data point j.

New subproblem will be as given below, **SP'**. The master problem is taken as it is.

New Subproblem

$$
\begin{aligned}
(\textbf{SP'}) \quad \text{Minimize} \quad & \sum_{i\in N}\sum_{j\in N}\sum_{k\in M} d_{ijk}w_{ijk} && && (5.31)\\
\text{subject to:} \quad & -x_{ij} \geq -\bar{y}_j, && \forall i,j \in N && (5.32)\\
& \sum_{j\in N} x_{ij} = 1, && \forall i \in N && (5.33)\\
& w_{ijk} - x_{ij} \geq \bar{z}_{jk} - 1, && \forall i,j \in N, \forall k \in M && (5.34)\\
& -\sum_{i\in N}\sum_{k\in M} w_{ijk} \geq -|N|\,q\,\bar{y}_j, && \forall j \in N && (5.35)\\
& -\sum_{i\in N} w_{ijk} \geq -|N|\,\bar{z}_{jk}, && \forall j \in N, \forall k \in M && (5.36)\\
& w_{ijk} \geq 0, && \forall i,j \in N, \forall k \in M && (5.37)\\
& x_{ij} \geq 0. && \forall i,j \in N && (5.38)
\end{aligned}
$$

The first three constraints are the same as the subproblem **SP**, and the following two constraints are the new valid inequalities.

Let denote variables α_{ij}, β_i, γ_{ijk}, θ_j, and δ_{jk} as the dual variables of the constraints (5.32)–(5.36), respectively. The dual of the new subproblem is as follows, $\mathbf{D_{SP'}}$.

Dual of the New Subproblem

$$(\mathbf{D_{SP}'}) \quad \text{Maximize} - \sum_{i\in N}\sum_{j\in N}\bar{y}_j\alpha_{ij} + \sum_{i\in N}\beta_i + \sum_{i\in N}\sum_{j\in N}\sum_{k\in M}(\bar{z}_{jk}-1)\gamma_{ijk}$$
$$-|N|\,q\sum_{j\in N}\bar{y}_j\theta_j - |N|\sum_{j\in N}\sum_{k\in M}\bar{z}_{jk}\delta_{jk} \quad (5.39)$$

$$\text{subject to:} \quad -\alpha_{ij} + \beta_i - \sum_{k\in M}\gamma_{ijk} \le 0, \qquad \forall i,j \in N \quad (5.40)$$

$$\gamma_{ijk} - \theta_j - \delta_{jk} \le d_{ijk}, \qquad \forall i,j \in N, \forall k \in M \quad (5.41)$$

$$\alpha_{ij} \ge 0, \qquad \forall i,j \in N \quad (5.42)$$

$$\gamma_{ijk} \ge 0, \qquad \forall i,j \in N, \forall k \in M \quad (5.43)$$

$$\theta_j \ge 0, \qquad \forall j \in N \quad (5.44)$$

$$\delta_{jk} \ge 0, \qquad \forall j \in N, \forall k \in M \quad (5.45)$$

$$\beta_i \; urs. \qquad \forall i \in N \quad (5.46)$$

The cut that will be added to the master problem will change also as given in (5.47).

$$\xi \ge -\sum_{i\in N}\sum_{j\in N}y_j\bar{\alpha}_{ij} + \sum_{i\in N}\bar{\beta}_i + \sum_{i\in N}\sum_{j\in N}\sum_{k\in M}(z_{jk}-1)\bar{\gamma}_{ijk}$$
$$-|N|\,q\sum_{j\in N}y_j\bar{\theta}_j - |N|\sum_{j\in N}\sum_{k\in M}z_{jk}\bar{\delta}_{jk} \quad (5.47)$$

Adding the given valid inequalities decreased the computational time comparing to the (BD) which uses **SP** and $\mathbf{D_{SP}}$. However, it is observed that the performance of the proposed Benders Decomposition is still lower than the performance of our linearized model $\mathbf{LM_1}$.

We inspired from the Benders Decomposition, and develop a heuristic algorithm by following the relations between master and subproblems used in (BD).

5.2 Benders like Heuristic Algorithm: $\mathbf{H_1}$

Benders Decomposition divides larger problems into smaller problems as *master* and *subproblem*. The solution of one of these smaller problems is fixed in the other problem. In this heuristic algorithm, we follow the same principal. If assignment of all data points are given in advance, selection of cluster centers and features can be decided simultaneously. Also, when the cluster centers and selected features are known, the assignments of data points can be obtained easily. Therefore, as in Benders decomposition, we have divided our problem into two smaller problems. In the first problem, cluster centers and selected features are decided by using a mathematical model, as in **MP**. The second problem uses the information obtained from the first problem, and decides the assignments of data points. Different than **SP**, we are using simple heuristic algorithm to assign data points to a cluster. Also, instead of adding the cut to provide information, we directly give the assignments to the first problem. Details of the proposed heuristic algorithm and its steps will be provided in this section.

5.2.1 Center and Feature Problem: $\mathbf{H_X}$

In this smaller problem, it is assumed that assignment of data points to clusters are given in advance. That means, we are eliminating the assignment variable from the optimization problem, and cluster centers and features to be selected will be decided. Schematic representation of this smaller problem for two clusters can be seen in Figure 5.3. Here, c_1 and c_2 are for cluster centers, and Q_1 and Q_2 are representing the features that define clusters. Solid line states that assignment of data points are known, whereas dashed lines show that cluster centers and features will be decided.

We propose a new mathematical model to decide cluster centers and relevant features for each cluster. This model contains the assignment information different than our previous mathematical models. The mathematical model is provided below and it will be called as model $\mathbf{H_X}$.

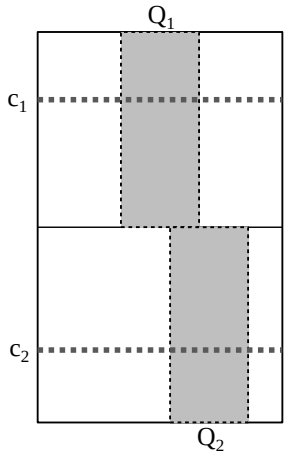

Figure 5.3: Schematic Representation of H_X

$$(\mathbf{H_X}) \quad \text{Minimize} \quad \sum_{p\in P}\sum_{i\in C_p}\sum_{j\in C_p}\sum_{k\in M} d_{ijk} z_{jpk} \tag{5.48}$$

$$\text{subject to:} \quad \sum_{j\in C_p} y_{jp} = 1, \qquad \forall p \in P \tag{5.49}$$

$$\sum_{k\in M} z_{jpk} = q\, y_{jp}, \qquad \forall j \in N, \forall p \in P \tag{5.50}$$

$$z_{jpk} \in \{0,1\}, \qquad \forall j \in N, \forall k \in M, \forall p \in P \tag{5.51}$$

$$y_{jp} \in \{0,1\}. \qquad \forall j \in N, \forall p \in P \tag{5.52}$$

Objective function of this formulation minimizes the total distance between the data points and cluster centers of those groups via selected features. Notice that the clusters are already formed since we know the data points which are assigned to the same group. In this formulation, z_{jpk} will be 1 if feature k is selected for cluster p where data point j is the center of that cluster. The decision variable y_{jp} will be 1 if data point j is selected as the cluster center of cluster p. Constraint (5.49) ensures that every cluster will have a cluster center which is one of the data points belonging to that cluster. Constraint (5.50) aims to select q features for each cluster.

The model looks like the master problem of the Benders Decomposition of our problem. However, model $\mathbf{H_X}$ includes the assignment information directly while selecting centers and features, and the objective function is different since the assignments are known in advance. We know the constructed clusters based on the given as-

signments, so the model $\mathbf{H_X}$ can be solved separately for each cluster to find cluster centers and relevant features.

Notation used in this problem can be found in Table 5.1

Table 5.1: Notation used in Mathematical Model H_X

Sets	
N	Set of data points
M	Set of features
P	Set of clusters
Parameters	
C_p	Data points in cluster p, $p \in P$
q	Number of features that should be chosen for each cluster
d_{ijk}	Distance between data points i and j on feature k, $i, j \in N$, $k \in M$
Decision Variables	
y_{jp}	Binary decision variable as 1 if data point j is selected as a cluster center of cluster p, $j \in N$, $p \in P$
z_{jpk}	Binary decision variable as 1 if feature k is selected for cluster center j where data point j is a center of cluster p, $j \in N$, $k \in M$, $p \in P$

5.2.2 Assignment Problem: $\mathbf{P_X}$

We have obtained the clusters centers and selected features from the mathematical model $\mathbf{H_X}$, and they will be represented as $\bar{y}_{jp}$ and $\bar{z}_{jpk}$, respectively. When they are known and fixed, only the assignment of data points remains. Schematic representation of assignment problem is given in Figure 5.4. Here, dashed line states that assignment of data points are not known, whereas solid lines show that cluster centers and features were decided.

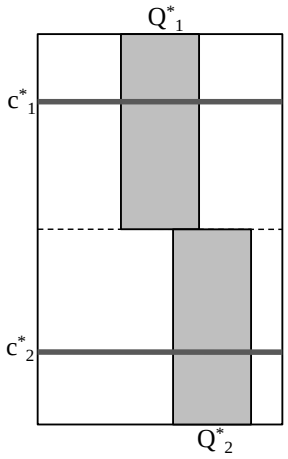

Figure 5.4: Schematic Representation of Assignment Problem

$\mathbf{P_X}$ is the mathematical formulation used to decide the assignments of data points.

$$(\mathbf{P_X}) \quad \text{Minimize} \quad \sum_{i \in N} \sum_{j \in C_p} \sum_{k \in M} d_{ijk} \bar{z}_{j x_i k} \tag{5.53}$$

$$\text{subject to:} \quad x_i \geq 1, \qquad \forall i \in N \tag{5.54}$$

$$x_i \leq |P|. \qquad \forall i \in N \tag{5.55}$$

x_i denotes the cluster number in which data point i is assigned, and $|P|$ gives the number of clusters. Together with the objective function, constraints ensure that every data point should be assigned to one cluster. Objective function will be minimized where all data points are assigned to their closest centers. We can use a simple search algorithm which calculates total distance of each data point to all cluster centers via selected features for those clusters, and assigns data points to do closest cluster center. Pseudocode of the assignment problem is provided in Algorithm 1, and the procedure will be called as $AssignmentPx()$.

The outer for loop ($lines\ 1-20$) is constructed for assigning all data points to a cluster. In the inner loop ($lines\ 3-18$), the distance between data points i and j is calculated if the data point j is a cluster center for cluster p, and feature k is selected for that cluster. Temporary center of data point i is assigned as p if d_{ij} is the minimum distance known so far and j is the center of that cluster ($lines\ 12-15$). When all possible clusters are considered for i, it is assigned to the temporary center t.

Algorithm 1: Assignment Algorithm I

```
Procedure AssignmentPx()
    input  : ȳ_jp, z̄_jpk
    output: x_i
1   for i=1,...,N
2       Let min be a very large number
3       for p=1,...,P
4           for j=1,...,N
5               Let d_ij = 0        % total distance between data points i and j
6               if ȳ_jp = 1 then
7                   for k=1,...,M
8                       if z̄_jpk = 1 then
9                           d_ij = d_ij + d_ijk
10                      end
11                  end for
12                  if d_ij < min then
13                      min = d_ij
14                      t = p
15                  end
16              end
17          end for
18      end for
19      x_i = t
20  end for
end
```

Our heuristic algorithm $\mathbf{H_1}$ iteratively use $\mathbf{H_X}$ and $\mathbf{P_X}$ in order to obtain clustering solution to our problem. At the start of the algorithm, each data point is randomly assigned to a cluster. Then, $\mathbf{H_X}$ is solved to obtain cluster centers and relevant features of those clusters according to the given assignments. After finding the data points which are selected as a cluster center and relevant features, each data point is assigned to its closest cluster center using the Algorithm 1. $\mathbf{H_1}$ terminates when the difference between the objective functions of the two consecutive iterations are smaller than a threshold value. The steps of $\mathbf{H_1}$ can also be seen below.

Step 0. Initialize $total\_dist_{old} = 0$, $total\_dist = 0$

Step 1. Randomly assign all data points to a cluster

Step 2. Solve $\mathbf{H_X}$ to obtain cluster centers and selected features

Step 3. Call $AssignmentPx()$ to obtain the assignments of data points using cluster centers and selected features

Step 4. Calculate the total distance between all data points and their cluster centers via selected features, set it as $total\_dist$.

If $\mid total\_dist - total\_dist_{old} \mid \leq \epsilon$ where ϵ is a predefined small threshold value; if iteration limit is not reached then go to **Step 1**, otherwise STOP.

If $\mid total\_dist - total\_dist_{old} \mid > \epsilon$, set $total\_dist_{old} = total\_dist$. Go to **Step 2**.

Note that, the algorithm is initialized 50 times to reduce the effect of random assignments.

The algorithm can also be tracked by using the flow chart given in Figure 5.5.

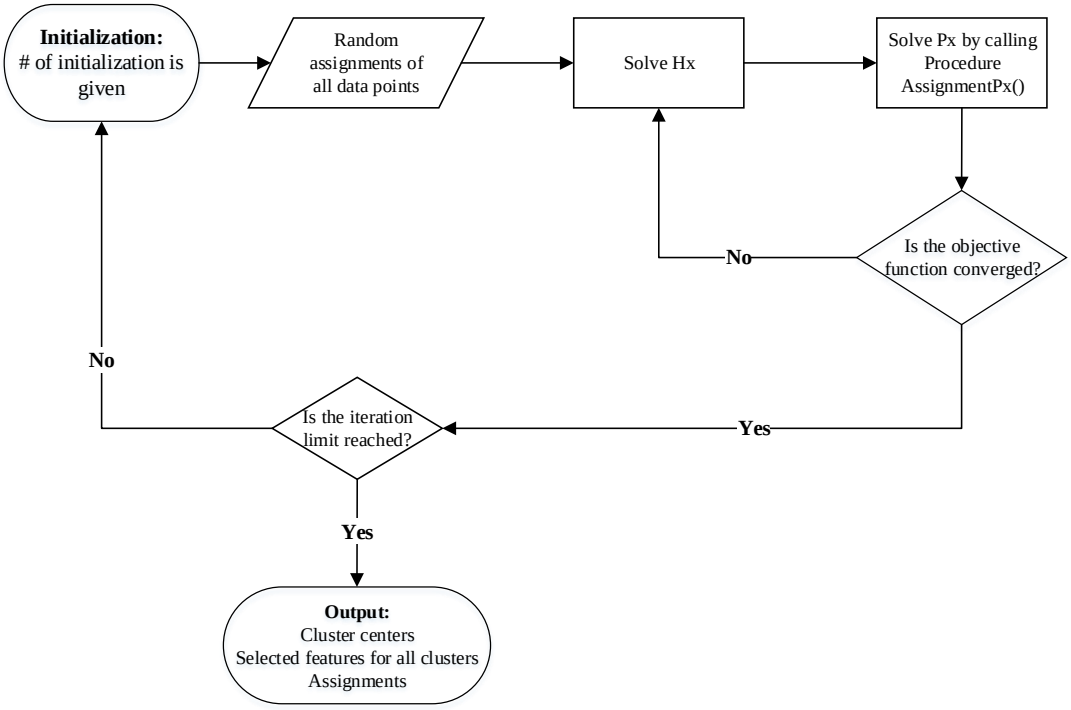

Figure 5.5: Flow Chart of the Benders like Heuristic Algorithm

The pseudocode of $\mathbf{H_1}$ is given in Algorithm 2.

Algorithm 2: Benders like Heuristic Algorithm

Procedure H_1 ()
input : $Data\ set\ N$
output: $Assignment\ of\ data\ points\ ,\ cluster\ centers\ ,\ selected\ features$

Read data set
Set $counter_{max}, stopping\_condition, total\_dist_{best}$
repeat
  Let $absdist$ be a very big number
  Step 0. Set $total\_dist = 0$ and $total\_dist_{old} = 0$
  Step 1. Randomly assign data points to clusters, x_i
  repeat
    $total\_dist_{old} \leftarrow total\_dist$
    Step 2. Center and Feature Selection
    Solve $H_X \rightarrow y_{jp}, z_{jpk}$
    Step 3. Assignment Update
    Call $AssignmentPx() \rightarrow x_i$
    Step 4. Calculate total distance $total\_dist$
    $absdist \leftarrow \mid total\_dist - total\_dist_{old} \mid$
  until $absdist \leq stopping\_condition$
  if $total\_dist < total\_dist_{best}$ **then**
    $total\_dist_{best} \leftarrow total\_dist$
    for *j=1,...,N*
      for *p=1,...,P*
        $y_{jp_{best}} \leftarrow y_{jp}$
        for *k=1,...,M*
          $z_{jpk_{best}} \leftarrow z_{jpk}$
        end for
      end for
    end for
    for *i=1,...,N*
      $x_{i_{best}} \leftarrow x_i$
    end for
  end
  $counter = counter + 1$
until $counter > counter_{max}$
end

Algorithm 2 starts with randomly assigning data points to a cluster. The inner loop updates cluster centers, selected features and assignments until convergence ($lines\ 9 - 15$). Best assignments, best cluster centers and their selected features are updated if the total distance of the current solution is better than the total distance of previously identified clustering solution ($lines\ 16 - 31$).

The next chapter will introduce a new heuristic algorithm which works in iterative manner, where all decision variables are decided by using a simple heuristic algorithm depending on the nature of the problems.

CHAPTER 6

ITERATIVE HEURISTIC ALGORITHM

Proposed mathematical models perform poorly with the size of the data set. In Chapter 5, Benders Decomposition solution method and Benders like heuristic algorithm have been introduced. In this chapter, a new heuristic algorithm will be delivered and it will be referred as $\mathbf{H_2}$.

Our problem depends on three decision variables. If X, C, and Q denote the decision variables for assignments of data points, cluster centers, and features to be selected, then the problem can be represented as:

$$\begin{aligned} \text{minimize} \quad & f(X, C, Q) \\ \text{s.to} \quad & \Omega \end{aligned}$$

where Ω shows the feasible region of the problem.

Our heuristic algorithm uses the idea of fixing two decision variables and solving the problem for the remaining decision variable. Therefore, we can mention about three main problems.

- Assignment Problem where C and Q are fixed. It is called $\mathbf{P_X}$.
- Center Selection Problem where X and Q are fixed. It is called $\mathbf{P_C}$.
- Feature Selection Problem where X and C are fixed. It is called $\mathbf{P_Q}$.

Details of the these main problems will be provided before introducing the proposed heuristic $\mathbf{H_2}$ which benefits from those problems in an iterative fashion.

6.1 Assignment Problem: P_X

This main problem takes cluster centers and selected features as given, and it only decides the assignments of data points to a cluster. The mentioned problem can be represented as given in Figure 6.1 where there are two clusters. Dashed line shows clusters so the assignment of data points which are not decided yet, and solid lines show the fixed decision variables, cluster centers and selected features, c_1^*, c_2^* and Q_1^*, Q_2^*, respectively.

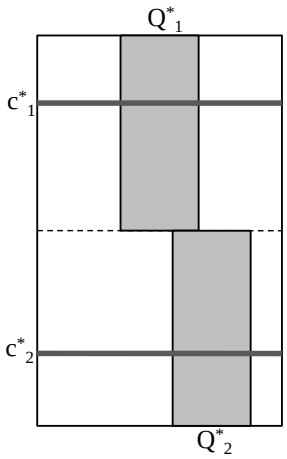

Figure 6.1: Schematic Representation of Assignment Problem

Conceptual model of this main problem will as follows. Here, we fix the cluster centers and selected features, and try to find best assignments, where Ω_X represents the feasible region of the problem.

$$\begin{aligned} \text{minimize} \quad & f(X, C = C^*, Q = Q^*) \\ \text{s.to} \quad & \Omega_X \end{aligned}$$

The mathematical model of the given problem $\mathbf{P_X}$ can be seen below.

$$(\mathbf{P_X}) \quad \text{Minimize} \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in M} d_{ijk}x_{ij}\bar{z}_{jk} \tag{6.1}$$

$$\text{subject to:} \quad x_{ij} \leq \bar{y}_j, \qquad \forall i,j \in N \tag{6.2}$$

$$\sum_{j\in N} x_{ij} = 1, \qquad \forall i \in N \tag{6.3}$$

$$x_{ij} \in \{0,1\}. \qquad \forall i,j \in N \tag{6.4}$$

where x_{ij} is a binary variable which takes the value of 1 if a data point i is assigned to data point j. In this formulation, $\bar{y}_j$ will denote the fixed cluster centers and $\bar{z}_{jk}$ will represent the given selected feature k for cluster j. The aim is to minimize the total distance between data point i and data point j where feature k is selected for that cluster. Constraint (6.2) ensures that a data point i may be assigned to j if data point j is a center, and (6.3) states that a data point must be assigned to a cluster.

The Assignment Problem in $\mathbf{H_1}$ mentioned in the Chapter 5 is different than the problem $\mathbf{P_X}$. In the former, data points are assigned to a cluster, that means we are forming the groups of data points without considering the centers. Here in $\mathbf{P_X}$, data points are assigned to one of the given cluster centers. The data points assigned to the same center form the clusters. As in $\mathbf{H_1}$, model $\mathbf{P_X}$ can be solved separately for each data point because objective function will be minimized where all data points are assigned to their closest cluster centers via the features defining those clusters. Therefore, a simple search algorithm can be used to find the closest center of each data point. Pseudocode of the assignment problem is provided in Algorithm 3.

Algorithm 3: Assignment Algorithm II

```
   Procedure Assignment()
   |  input : ȳ_j, z̄_jk
   |  output: x_ij
1  |  for i=1,...,N
2  |  |  Let min be a very large number
3  |  |  for j=1,...,N
4  |  |  |  Let d_ij = 0          % total distance between data points i and j
5  |  |  |  if ȳ_j = 1 then
6  |  |  |  |  for k=1,...,M
7  |  |  |  |  |  if z̄_jk = 1 then
8  |  |  |  |  |  |  d_ij = d_ij + d_ijk
9  |  |  |  |  |  end
10 |  |  |  |  end for
11 |  |  |  |  if d_ij < min then
12 |  |  |  |  |  min = d_ij
13 |  |  |  |  |  t = j
14 |  |  |  |  end
15 |  |  |  end
16 |  |  end for
17 |  |  x_it = 1
18 |  end for
   end
```

Different than Algorithm 1, cluster center of data point i is selected as data point j if d_{ij} is the minimum distance known so far ($lines\ 11-14$). When all possible cluster centers are considered for data point i, it is assigned to the cluster center t in $line\ 17$.

6.2 Center Selection Problem: $\mathbf{P_C}$

This problem is used to decide cluster centers given the assignments of data points and features that should be used in clustering. Since we know the groups of data points, center of a group should be selected among data points which are assigned to that group. Figure 6.2 schematically represents center selection problem. Dashed lines say that centers of the clusters have not been decided. With the solid lines, assignments of data points and selected features are shown. Unknown cluster centers and selected features are represented with c_1, c_2 and Q_1^*, Q_2^*, respectively.

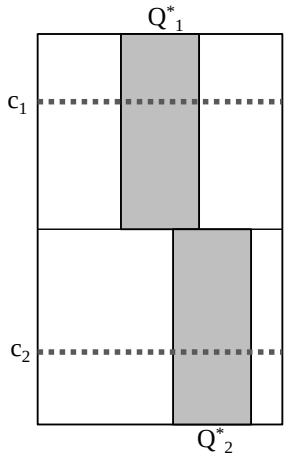

Figure 6.2: Schematic Representation of Center Selection Problem

Conceptual model of this main problem will as follows. Here, assignments of data points and selected features are fixed, and we try to find best cluster centers. In the formulation, Ω_C represents the feasible region.

$$\begin{aligned} \text{minimize} \quad & f(C, X = X^*, Q = Q^*) \\ \text{s.to} \quad & \Omega_C \end{aligned}$$

The mathematical model of the given problem can be seen below named as model $\mathbf{P_C}$.

$$(\mathbf{P_C}) \quad \text{Minimize} \quad \sum_{p\in P}\sum_{i\in C_p}\sum_{j\in C_p}\sum_{k\in Q_p} d_{ijk}\bar{z}_{jk}y_j \tag{6.5}$$

$$\text{subject to:} \quad \sum_{j\in C_p} y_j = 1, \qquad \forall p \in P \tag{6.6}$$

$$y_j \in \{0,1\}. \qquad \forall j \in N \tag{6.7}$$

where y_j is a binary variable which takes the value of 1 if a data point j is selected as a cluster center among the data points which are in cluster p, where those data points are in set C_p which includes the data points in cluster p. In this formulation, Q_p is the set of features used in cluster p. (6.6) ensures that center of each cluster will be selected among the data points in that cluster. Notice that objective function of the problem will be minimized when the data point which is located in the middle

of the cluster is selected as a center. Since we know the assignments of data points and selected features, we can decompose the problem into smaller problems where center of each cluster is decided separately. For each data point, total distance to other data points in a cluster is calculated, and the one which is closest to all other cluster members is selected as the cluster center. The following Algorithm 4 is provided to find cluster centers.

Algorithm 4: Center Selection Algorithm

```
Procedure Center()
    input : x̄_ij, z̄_jk
    output: y_j
1   for i=1,...,N
2       Let d_i = 0                          % total distance to data point i
3       for n=1,...,N
4           for j=1,...,N
5               if x̄_ij = 1 and x̄_nj = 1 then
6                   for k=1,...,M
7                       if z̄_jk = 1 then
8                           d_i = d_i + d_ink
9                       end
10                  end for
11              end
12          end for
13      end for
14  end for
15  for i=1,...,N
16      Let min = d_i
17      best = i
18      for n=1,...,N
19          for j=1,...,N
20              if x̄_ij = 1 and x̄_nj = 1 and d_n < min then
21                  min = d_n
22                  best = n
23                  j = N
24              end
25          end for
26      end for
27      y_best = 1
28  end for
end
```

The algorithm starts with a for loop to calculate the total distance to data point i by considering the data points which are assigned to the same cluster with data point

i ($lines\ 1-14$). The for loop is repeated for all data points. After calculating all distances, data point which is closest to all data points in its cluster is selected as the cluster center ($lines\ 15-28$).

6.3 Feature Selection Problem: P_Q

Feature selection problem takes the assignments of data points and cluster center as given. Then, the problem only decides the relevant features for each cluster that will minimize the total distance between data points and cluster centers via selected features. Feature selection problem is given in Figure 6.3. Dashed lines say that relevant features of each cluster have not been decided. With the solid lines, it is stated that assignments of data points and cluster centers are known. In figure, cluster centers and unknown relevant features are represented with c_1^*, c_2^* and Q_1, Q_2, respectively.

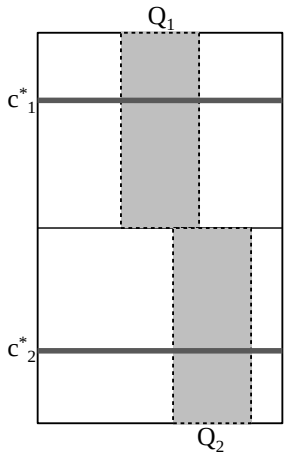

Figure 6.3: Schematic Representation of Feature Selection Problem

Conceptual model of this main problem will as follows. Here, assignment of data points and cluster centers are known and fixed. The model states that relevant features of each cluster should be selected, where Ω_Q represents the feasible region.

$$\begin{aligned} \text{minimize} \quad & f(Q, C = C^*, X = X^*) \\ \text{s.to} \quad & \Omega_Q \end{aligned}$$

The mathematical model of feature selection problem $\mathbf{P_Q}$ is presented below.

$$(\mathbf{P_Q}) \quad \text{Minimize} \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in M} d_{ijk}\bar{x}_{ij}z_{jk} \tag{6.8}$$

$$\text{subject to:} \quad \sum_{k\in M} z_{jk} = q\,\bar{y}_j, \qquad \forall p \in P \tag{6.9}$$

$$z_{jk} \in \{0,1\}. \qquad \forall j \in N, \forall k \in M \tag{6.10}$$

where z_{jk} takes 1 if feature k is selected for cluster where j is the center. Here, given assignments and cluster centers will be represented as $\bar{x}_{ij}$ and $\bar{y}_j$, respectively. Constraint (6.9) is used to select q features for each cluster. Notice that objective function of the problem includes nonnegative coefficients for all z_{jk} variables. Therefore, the objective function will be minimized when the features which are the most compact ones are selected. Since we are allowing that one feature may be used in more than one cluster, relevant features of each cluster can be selected by sorting the total distance on each feature in ascending order. That means, we can decompose feature selection problem as well. The following Algorithm 5 finds those q features for each cluster separately.

In this algorithm, for each cluster centers, total distance through each feature is calculated, d_{jk}, ($lines\ 1-12$). In order to find the features which are relevant for the cluster centers, d_{jk} values are sorted in ascending order, Q features with the smallest d_{jk} values are selected for each cluster ($lines\ 13-32$).

Algorithm 5: Feature Selection Algorithm

```
Procedure Feature()
    input  : ȳ_j, x̄_ij
    output: z_jk
1   for k=1,...,M
2       for j=1,...,N
3           Let d_jk = 0      % total distance to data point j through feature k
4           if ȳ_j = 1 then
5               for i=1,...,N
6                   if x̄_ij = 1 then
7                       d_jk = d_jk + d_ijk
8                   end
9               end for
10          end
11      end for
12  end for
13  for j=1,...,N
14      if ȳ_j = 1 then
15          for k=1,...,M
16              sort_dist_k = d_jk
17          end for
18          Sort sort_dist smallest to largest → sorted_dist
19          total_selected = 0
20          for k=1,...,M
21              if total_selected < Q then
22                  for q=1,...,Q
23                      if d_jk = sorted_dist_q then
24                          z_jk = 1
25                          q = Q
26                          total_selected = total_selected + 1
27                      end
28                  end for
29              end
30          end for
31      end
32  end for
end
```

6.4 Iterative Heuristic Algorithm: H_2

Three main problems, $\mathbf{P_X}$, $\mathbf{P_C}$, and $\mathbf{P_Q}$, are discussed before introducing heuristic $\mathbf{H_2}$. Iterative heuristic algorithm $\mathbf{H_2}$ benefits from those problems in an iterative manner when two of the decision variables are fixed and the problem is solved for the remaining one. $\mathbf{H_2}$ can be divided into three subroutines in which one of the decision variables is fixed, and the other two decision variables are updated iteratively by using the algorithms proposed for the solution of $\mathbf{P_X}$, $\mathbf{P_C}$, and $\mathbf{P_Q}$ problems. For example, if we fix the cluster centers, we need to update assignments and selected features iteratively. To do that, in each iteration, we fix one of them. When selected features are fixed, the problem is reduced to the problem $\mathbf{P_X}$ which is solved by Algorithm 3. After deciding the assignments, we will fix them besides the cluster centers, and relevant features are found by using the Algorithm 5 for the solution of problem $\mathbf{P_Q}$. This subroutine is named as feature-assignment update subroutine, and it is called as $\mathbf{P_{QX}}$. $\mathbf{H_2}$ will also have assignment-center update $\mathbf{P_{XC}}$ and center-assignment update $\mathbf{P_{CX}}$ subroutines which work in the same manner. In the following subsections, we will cover those subroutines in detail, and Figure 6.4 schematically represents the relation between subroutines.

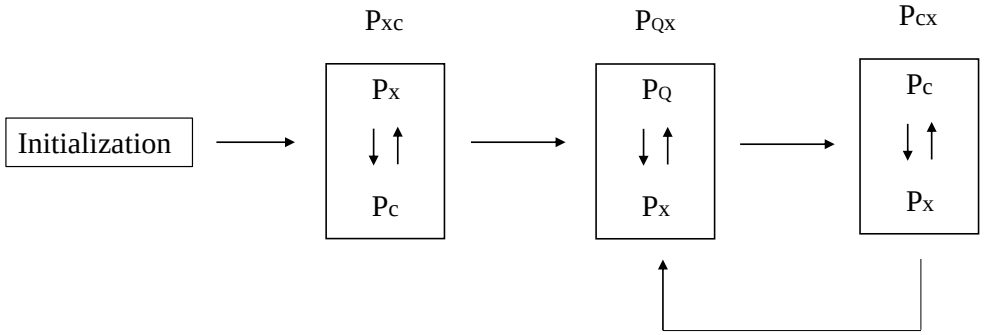

Figure 6.4: Schematic Representation of the Relations between Subroutines

6.4.1 Assignment-Center Update: P_{XC}

In this subroutine, features of each cluster are fixed, assignments of data points and cluster centers will be found. Assignments and cluster centers are updated iteratively where in each step, one of them is fixed. When the assignments are fixed, since the

features are also given, the problem turns into the problem $\mathbf{P_C}$. Similarly, the problem turns into $\mathbf{P_X}$ when the cluster centers are fixed. Therefore, this subroutine is solving $\mathbf{P_X}$ and $\mathbf{P_C}$ problems iteratively and finds the best cluster centers and assignments of data points while the features are given. For solving $\mathbf{P_X}$ and $\mathbf{P_C}$, we utilize $Assignment()$ and $Center()$ procedures in Algorithm 3 and Algorithm 4, respectively. Steps of this subroutine are provided below.

Step 0. Initialize $total\_dist_{old} = 0$, $total\_dist = 0$

Step 1. Solve for X by calling $Assignment()$ for given cluster centers ($\bar{y}_j$) and selected features ($\bar{z}_{jk}$)

Step 2. Solve for C by calling $Center()$ for given assignments ($\bar{x}_{ij}$) and selected features ($\bar{z}_{jk}$)

Step 3. Calculate the total distance between all data points and their cluster centers through selected features, set it as $total\_dist$

If $\mid total\_dist - total\_dist_{old} \mid \leq \epsilon$ where ϵ is a predefined small threshold value, STOP.

If $\mid total\_dist - total\_dist_{old} \mid > \epsilon$, set $total\_dist_{old} = total\_dist$. Go to **Step 1**.

The pseudocode of the subroutine can also be found in Algorithm 6.

Algorithm 6: Assignment-Center Update Algorithm

Procedure $\mathtt{P_{XC}}$ ()
  input : $absdist$, $stopping\_condition$, $\bar{y}_j$, $\bar{z}_{jk}$
  output: x_{ij}, y_j
1 **Step 0.** $total\_dist_{old} = 0$, $total\_dist = 0$
2 **repeat**
3 $total\_dist_{old} \leftarrow total\_dist$
4 **Step 1.** Assignment Update
5 Call $Assignment() \rightarrow x_{ij}$
6 **Step 2.** Center Update
7 Call $Center() \rightarrow y_j$
8 **Step 3.** Calculate total distance $total\_dist$
9 $absdist \leftarrow \mid total\_dist - total\_dist_{old} \mid$
10 **until** $absdist \leq stopping\_condition$
end

6.4.2 Feature-Assignment Update: $\mathbf{P_{QX}}$

In this subroutine, assignments of data points and relevant features of each cluster will be found while cluster centers are given. In each step, relevant features and assignments are updated iteratively by fixing one of them. When the assignments are fixed, since the cluster centers are also given, the problem turns into the problem $\mathbf{P_Q}$. Similarly, if the relevant features are fixed, then the problem turns into $\mathbf{P_X}$. That means, this subroutine is solving $\mathbf{P_Q}$ and $\mathbf{P_X}$ problems iteratively. For solving $\mathbf{P_X}$ and $\mathbf{P_Q}$, we utilize $Assignment()$ and $Feature()$ procedures in Algorithm 3 and Algorithm 5, respectively. The steps of the feature-assignment update subroutine can be seen below.

Step 0. Initialize $total\_dist_{old} = 0$, $total\_dist = 0$

Step 1. Solve for Q by calling $Feature()$ for given cluster centers ($\bar{y}_j$) and assignments ($\bar{x}_{ij}$)

Step 2. Solve for X by calling $Assignment()$ for given cluster centers ($\bar{y}_j$) and selected features ($\bar{z}_{jk}$)

Step 3. Calculate the total distance between all data points and their cluster centers through selected features, set it as $total\_dist$

If $\mid total\_dist - total\_dist_{old} \mid \leq \epsilon$ where ϵ is a predefined small threshold value, STOP.

If $\mid total\_dist - total\_dist_{old} \mid > \epsilon$, set $total\_dist_{old} = total\_dist$. Go to **Step 1**.

The pseudocode of the subroutine is given in Algorithm 7.

Algorithm 7: Feature-Assignment Update Algorithm

Procedure $\mathtt{P_{QX}}$ ()
  input : $absdist$, $stopping\_condition$, $\bar{y}_j$, $\bar{x}_{ij}$
  output: x_{ij}, z_{jk}
1 **Step 0.** $total\_dist_{old} = 0$, $total\_dist = 0$
2 **repeat**
3 $total\_dist_{old} \leftarrow total\_dist$
4 **Step 1.** Feature Selection
5 Call $Feature() \rightarrow z_{jk}$
6 **Step 2.** Assignment Update
7 Call $Assignment() \rightarrow x_{ij}$
8 **Step 3.** Calculate total distance C_3
9 $absdist \leftarrow \mid total\_dist - total\_dist_{old} \mid$
10 **until** $absdist \leq stopping\_condition$
end

6.4.3 Center-Assignment Update: $\mathbf{P_{CX}}$

In this subroutine, features of each cluster are fixed, assignments of data points and cluster centers will be found as in subroutine $\mathbf{P_{XC}}$. We will start by fixing assignment of data points besides selected features and iteratively update cluster centers and assignments, but subroutine $\mathbf{P_{XC}}$ starts with fixed selected features and randomly decided cluster centers. For the solution of $\mathbf{P_X}$ and $\mathbf{P_C}$ problem, $Assignment()$ and $Center()$ procedures in Algorithm 3 and Algorithm 4 will be utilized, respectively. Steps of this subroutine are provided below.

Step 0. Initialize $total\_dist_{old} = 0$, $total\_dist = 0$

Step 1. Solve for C by calling $Center()$ for given assignments ($\bar{x}_{ij}$) and selected features ($\bar{z}_{jk}$)

Step 2. Solve for X by calling $Assignment()$ for given cluster centers ($\bar{y}_j$) and selected features ($\bar{z}_{jk}$)

Step 3. Calculate the total distance between all data points and their cluster centers through selected features, set it as $total\_dist$

If $\mid total\_dist - total\_dist_{old} \mid \leq \epsilon$ where ϵ is a predefined small threshold value, STOP.

If $\mid total\_dist - total\_dist_{old} \mid > \epsilon$, set $total\_dist_{old} = total\_dist$. Go to **Step 1**.

The pseudocode of the subroutine is provided in Algorithm 8.

Algorithm 8: Center-Assignment Update Algorithm

Procedure $\texttt{P}_{\texttt{CX}}$ ()
  input : $absdist$, $stopping\_condition$, $\bar{x}_{ij}$, $\bar{z}_{jk}$
  output: x_{ij}, y_j
1 **Step 0.** $total\_dist_{old} = 0$, $total\_dist = 0$
2 **repeat**
3 $total\_dist_{old} \leftarrow total\_dist$
4 **Step 1.** Center Update
5 Call $Center() \rightarrow y_j$
6 **Step 2.** Assignment Update
7 Call $Assignment() \rightarrow x_{ij}$
8 **Step 3.** Calculate total distance $total\_dist$
9 $absdist \leftarrow \mid total\_dist - total\_dist_{old} \mid$
10 **until** $absdist \leq stopping\_condition$
end

$\mathbf{H_2}$ starts with random selection of cluster centers, and updates assignment and clusters using $\mathbf{P_{XC}}$ until convergence. Then, cluster centers are fixed, and selected features and assignment of data points are updated by $\mathbf{P_{QX}}$. In the next step of the algorithm, given the selected features, assignments and cluster centers are changed iteratively using $\mathbf{P_{CX}}$. Input and output relations between subroutines can be seen in Figure 6.5.

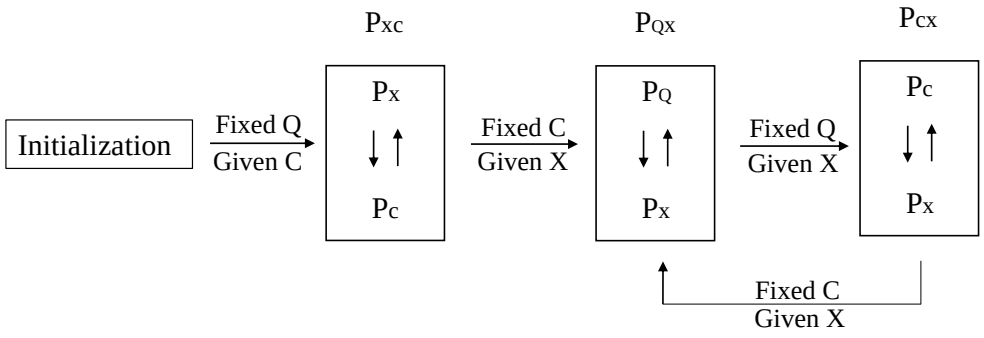

Figure 6.5: Schematic Representation of the Relations between Subroutines with Inputs/Outputs

Steps of our heuristic algorithm $\mathbf{H_2}$ is given below. Note that, the algorithm is initialized 50 times to reduce the effect of random start.

Step 0. Initialize $total\_dist_{old} = 0$, $total\_dist = 0$

Step 1. Randomly select cluster centers

Step 2. Call $\mathbf{P_{XC}}$ to obtain assignments and update cluster centers using all features

Step 3. Call $\mathbf{P_{QX}}$ to update selected features and assignments given the cluster centers

Step 4. Call $\mathbf{P_{CX}}$ to update cluster centers and assignments given the selected features

Step 5. Calculate the total distance between all data points and their cluster centers via selected features, set it as $total\_dist$

If $\mid total\_dist - total\_dist_{old} \mid \leq \epsilon$ where ϵ is a predefined small threshold value; if iteration limit is not reached then go to **Step 1**, otherwise STOP.

If $\mid total\_dist - total\_dist_{old} \mid > \epsilon$, set $total\_dist_{old} = total\_dist$. Go to **Step 3**.

Figure 6.6 provides the flow chart of the heuristic algorithm.

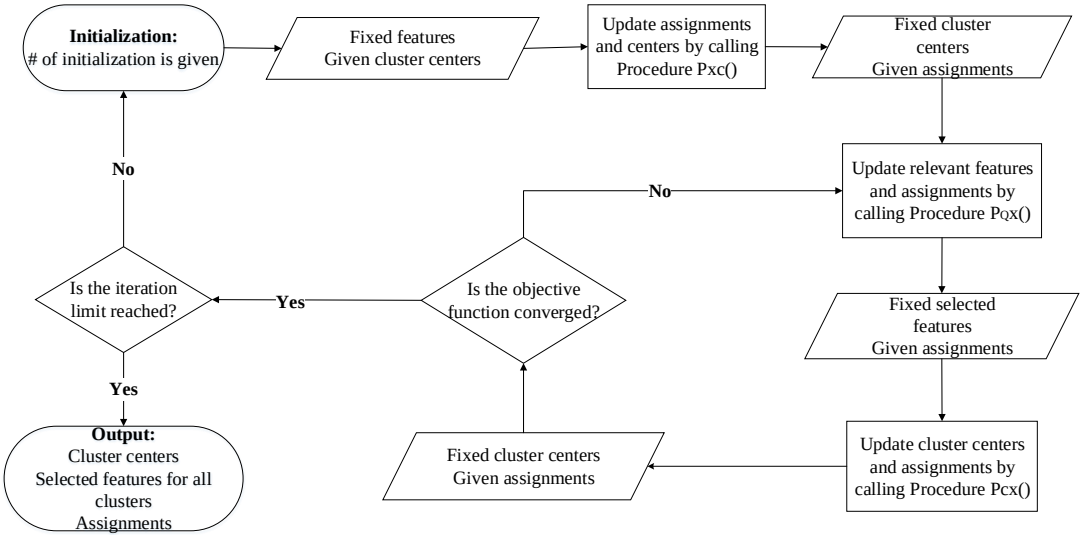

Figure 6.6: Flow Chart of the Iterative Heuristic Algorithm

The pseudocode of $\mathbf{H_2}$ is given in Algorithm 9. The algorithm starts with random selection of cluster centers in $line$ 6. Until convergence, assignments of data points and cluster centers are updated in $line$ 9. Updates of selected features and assignments, and then cluster centers and assignments are performed iteratively by using the subroutines $\mathbf{P_{QX}}$ and $\mathbf{P_{CX}}$ in $lines$ $10-18$, respectively. $Lines$ $19-30$ record the best solution which has the minimum distance obtained so far.

Algorithm 9: Iterative Heuristic Algorithm

Procedure H_2 ()

input : $Data\ set\ N$

output: $Assignment\ of\ data\ points,\ cluster\ centers\ ,\ selected\ features$

Read data set

Set $counter_{max}, stopping\_condition$

repeat

Let $absdist$ be a very big number

Step 0. Set $total\_dist = 0$ and $total\_dist_{old} = 0$

Step 1. Randomly select cluster centers, y_j

Assume $z_{jk} = 1$ for all $k \in M$

Step 2. Assignment-Center Update

Call $P_{XC}() \rightarrow x_{ij}, y_j$

repeat

$total\_dist_{old} \leftarrow total\_dist$

Step 3. Feature-Assignment Update

Call $P_{QX}() \rightarrow x_{ij}, z_{jk}$

Step 4. Center-Assignment Update

Call $P_{CX}() \rightarrow x_{ij}, y_j$

Step 5. Calculate total distance $total\_dist$

$absdist \leftarrow \mid total\_dist - total\_dist_{old} \mid$

until $absdist \leq stopping\_condition$

if $total\_dist < total\_dist_{best}$ **then**

$total\_dist_{best} \leftarrow total\_dist$

for *j=1,...,N*

$y_{j_{best}} \leftarrow y_j$

for *k=1,...,M*

$z_{jk_{best}} \leftarrow z_{jk}$

end for

for *i=1,...,N*

$x_{ij_{best}} \leftarrow x_{ij}$

end for

end for

end

$counter = counter + 1$

until $counter > counter_{max}$

end

In the next chapter, experimental studies conducted on the proposed heuristic algorithms will be discussed.

CHAPTER 7

COMPUTATIONAL RESULTS AND COMPARISON OF HEURISTICS ALGORITHMS

Empirical results that compare heuristic algorithms will be provided in this chapter. In the first part of this chapter, in Section 7.1, we state the changes in the performance measures delivered in Chapter 4, and one additional performance measure is introduced. Section 7.2 will deliver the results of the experimental studies conducted on the introduced simulated data sets to compare the performances of proposed mathematical models and heuristic algorithms.

7.1 Performance Measure

We will benefit from the performance measures mentioned in Chapter 4. Percent gap from optimal solution ($\%Gap_O$) is removed from consideration. Also, percent gap from best available solution ($\%Gap_B$) will be calculated by considering all available solutions instead of taking only the ones obtained from mathematical models. In order to report the number of hits to known optimal solutions, there will be additional information provided with the performance measure $NOpt$. The number of known optimal solutions will be provided within parenthesis with $NOpt$. We will also use one other measure to represent performances of heuristic algorithms, namely $Hits$.

Number of hits ($Hits$): In our study, all heuristic algorithms start with random initialization of one of the decision variables. In order to eliminate the bias of the solution to initial parameters, algorithms start with 50 different random initializations. Apart from the best and worst solutions in terms of objective function, we also keep the number of hits to those solutions.

7.2 Computational Results of Proposed Methods for Simulated Data

Performances of mathematical models are compared in Chapter 4, and it is concluded that nonlinear model **NM** and model $\mathbf{LM_1}$ cannot solve the problem within reasonable time. Among the other mathematical models, model $\mathbf{LM_3}$ performs slightly better than model $\mathbf{LM_2}$. Hence, our heuristic algorithms will be compared also with the mathematical model $\mathbf{LM_3}$.

Proposed heuristic algorithms are tested on simulated data sets, and results for data sets including 40 data points can be seen in Tables 7.1–7.3. Tables B.1–B.12 provided in Appendix B will report the results of experimental studies conducted on data sets with 50, 80, 100, and 200 data points. For further comparing the performances of heuristic algorithms H_1 and H_2, there are additional data sets which include 500 and 1000 data points. The results of the experimental studies conducted on those data sets are reported in Tables B.13–B.18 in Appendix B.

Tables are constructed as in Chapter 4. The first there columns contain the parameters of data sets, which are number of clusters, number of features and number of relevant features. For our mathematical model $\mathbf{LM_3}$, we report three performance measures, $\%Gap_M$, $\%Gap_B$ and $CPU\ s.$. Clustering solutions with minimum and maximum objective functions obtained by heuristic algorithms $\mathbf{H_1}$ and $\mathbf{H_2}$ are evaluated based on $\%Gap_M$, $\%Gap_B$, and $Hits$. Completion time of the heuristic algorithms are also reported on those tables, $CPU\ s.$. Averages of all performance measures can be seen at the last row of each table.

For the mathematical model and heuristic algorithms, averages of performance measures $\%Gap_M$, $\%Gap_B$, and $CPU\ s.$ on the data sets with equal number of data points are summarized in Table 7.4. That means, the performance of each solution method is evaluated by considering 60 problem instances where data sets include n number of data points. It should be noted that the best and worst clustering solutions are taken into account when reporting the performances of heuristic algorithms. In this table, there is also a new column $NBest$ which reports how many times the corresponding solution method has found the best available solution.

Table 7.1: Comparison of LM_3, H_1, and H_2 for Data Sets with 40 Data Points and 2 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
2	4	2	0	0	3.61	0.25	0.25	20	47.16	47.16	1	9.73	0	0	7	3.74	3.74	32	0.02
2	4	3	0	0	1.00	0	0	50	0	0	50	5.14	0	0	50	0	0	50	0.02
2	5	2	0	0	1.99	0	0	40	105.09	105.09	8	7.40	0	0	48	57.74	57.74	2	0.02
2	5	3	0	0	2.70	0	0	50	0	0	50	7.32	0	0	50	0	0	50	0.02
2	5	4	0	0	1.48	0	0	50	0	0	50	6.09	0	0	50	0	0	50	0.02
2	6	2	-0.95	0	3.16	-0.95	0	18	73.04	74.70	2	9.21	-0.95	0	49	59.02	60.54	1	0.02
2	6	3	0	0	2.01	0	0	39	169.01	169.01	1	6.62	0	0	50	0	0	50	0.02
2	6	4	0	0	4.21	0	0	50	0	0	50	7.31	0	0	50	0	0	50	0.02
2	8	2	-0.37	0	5.69	-0.37	0	8	94.07	94.79	3	8.61	-0.37	0	32	70.22	70.86	1	0.02
2	8	3	0	0	7.13	0	0	31	96.72	96.72	1	10.18	3.48	3.48	49	84.13	84.13	1	0.02
2	8	4	0	0	6.01	0	0	45	128.20	128.20	5	8.48	0	0	47	121.12	121.12	3	0.02
2	8	6	-4.56	0	10.69	-4.56	0	50	-4.56	0	50	8.38	-4.56	0	50	-4.56	0	50	0.02
2	10	2	0	0	6.23	0	0	5	69.18	69.18	1	14.43	3.68	3.68	38	64.73	64.73	1	0.02
2	10	3	0	0	72.96	0	0	11	63.51	63.51	1	10.29	0	0	7	9.40	9.40	42	0.03
2	10	4	0	0	28.03	0	0	46	93.58	93.58	1	13.47	0	0	48	62.76	62.76	2	0.03
2	10	6	0	0	11.20	0	0	46	139.30	139.30	1	10.54	0	0	50	0	0	50	0.02
2	12	2	0	0	14.88	0	0	9	97.52	97.52	1	11.25	0	0	42	80.53	80.53	2	0.03
2	12	3	-4.34	0	110.58	-4.34	0	9	60.28	67.55	1	11.53	-4.34	0	25	55.71	62.78	4	0.03
2	12	4	0	0	110.35	0	0	27	51.32	51.32	1	14.15	0	0	43	46.37	46.37	1	0.03
2	12	6	-1.76	0	51.70	-1.76	0	47	83.57	86.86	3	12.82	-1.76	0	49	-0.89	0.88	1	0.03
Average			-0.60	0	22.78	-0.59	0.01	32.55	68.35	69.22	14.05	9.65	-0.24	0.36	41.70	35.50	36.28	22.15	0.02

Table 7.2: Comparison of LM_3, H_1, and H_2 for Data Sets with 40 Data Points and 3 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
3	4	2	0	0	2.02	0	0	1	71.31	71.31	2	10.52	0	0	36	103.10	103.10	5	0.01
3	4	3	0	0	1.08	0	0	50	0	0	50	9.00	0	0	38	157.98	157.98	2	0.01
3	5	2	-7.07	0	4.55	-7.07	0	2	39.44	50.05	1	10.79	-7.07	0	4	40.24	50.91	2	0.01
3	5	3	0	0	2.04	0	0	48	85.54	85.54	2	11.87	0	0	17	87.95	87.95	1	0.02
3	5	4	0	0	1.73	0	0	50	0	0	50	29.56	0	0	38	131.60	131.60	2	0.01
3	6	2	-6.19	0	21.72	13.61	21.11	8	77.15	88.85	1	10.31	-6.19	0	1	45.59	55.20	2	0.02
3	6	3	-2.17	0	5.27	-2.17	0	15	109.46	114.12	1	14.49	-2.17	0	32	111.48	116.18	4	0.01
3	6	4	0	0	2.46	0	0	50	0	0	50	11.54	0	0	38	103.94	103.94	1	0.02
3	8	2	-5.90	0	59.19	-4.96	1.00	15	23.75	31.50	1	15.74	-3.97	2.05	2	44.53	53.58	1	0.02
3	8	3	0	0	15.43	16.66	16.66	1	93.68	93.68	1	12.35	0	0	37	64.71	64.71	1	0.02
3	8	4	0	0	10.44	0	0	37	144.18	144.18	1	18.46	0	0	1	133.07	133.07	1	0.02
3	8	6	0	0	4.48	0	0	50	0	0	50	15.45	0	0	38	143.10	143.10	1	0.02
3	10	2	-24.07	0	1327.47	-23.18	1.18	2	3.26	36.01	1	15.27	-18.40	7.47	1	18.59	56.19	1	0.02
3	10	3	-0.35	0	587.33	12.58	12.98	4	53.73	54.27	1	13.89	-0.35	0	14	42.12	42.62	6	0.02
3	10	4	0	0	20.36	0	0	1	99.71	99.71	1	14.22	0	0	26	72.01	72.01	1	0.02
3	10	6	-4.16	0	11.95	-4.16	0	44	193.09	205.81	1	15.91	2.39	6.84	39	94.64	103.09	2	0.02
3	12	2	-23.91	1.519	TL	-25.05	0	2	-2.73	29.78	1	17.35	-17.76	9.73	1	5.92	41.31	1	0.02
3	12	3	-3.57	0	3400.82	18.21	22.58	1	54.10	59.80	1	18.19	-3.57	0	2	47.27	52.72	1	0.02
3	12	4	-3.73	0	134.70	-1.28	2.54	4	75.06	81.84	1	19.77	-3.73	0	27	50.59	56.42	2	0.03
3	12	6	-2.50	0	34.76	-2.50	0	44	137.16	143.23	1	19.76	-0.01	2.55	32	117.62	123.19	1	0.03
Average			-4.18	0.08	297.25	-0.47	3.90	21.45	62.89	69.48	10.90	15.22	-3.04	1.43	21.20	80.80	87.44	1.90	0.02

Table 7.3: Comparison of LM_3, H_1, and H_2 for Data Sets with 40 Data Points and 4 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
4	4	2	0	0	3.22	0	0	1	147.79	147.79	1	17.87	0	0	28	129.82	129.82	1	0.01
4	4	3	-1.04	0	1.47	-1.04	0	34	192.66	195.74	1	12.29	-1.04	0	27	248.48	252.14	1	0.01
4	5	2	-0.29	0	4.76	-0.29	0	2	73.87	74.37	1	16.35	-0.29	0	2	83.80	84.33	1	0.01
4	5	3	-4.97	0	2.26	-4.97	0	39	139.14	151.66	1	16.67	-4.90	0.08	31	93.50	103.63	3	0.01
4	5	4	-1.39	0	1.84	-1.39	0	36	111.46	114.45	2	40.59	-1.39	0	29	199.87	204.11	1	0.01
4	6	2	0	0	7.60	0	0	1	83.33	83.33	1	15.24	0	0	20	74.41	74.41	1	0.02
4	6	3	0	0	4.39	0	0	17	138.23	138.23	1	16.94	6.54	6.54	26	111.01	111.01	1	0.02
4	6	4	0	0	2.34	0	0	42	191.26	191.26	1	16.38	0	0	30	128.75	128.75	5	0.02
4	8	2	-27.72	0	123.37	-27.72	0	1	-17.54	14.09	1	19.35	-26.90	1.15	1	26.11	74.49	2	0.02
4	8	3	-7.42	0	24.12	7.86	16.51	1	104.30	120.67	1	21.09	-7.42	0	20	93.83	109.37	1	0.02
4	8	4	-1.12	0	11.26	-1.12	0	28	187.73	191.00	1	23.39	-1.12	0.01	21	145.65	148.44	2	0.02
4	8	6	-0.55	0	5.20	-0.55	0	47	125.27	126.52	1	20.58	-0.55	0	34	243.24	245.15	1	0.02
4	10	2	-30.88	0	4939.73	-27.49	4.89	1	-1.75	42.13	1	20.61	-26.47	6.37	1	11.39	61.15	1	0.02
4	10	3	-1.64	0	1846.36	22.45	24.49	1	67.95	70.75	1	21.50	0.65	2.33	11	81.35	84.37	1	0.02
4	10	4	-13.26	0	28.06	-13.26	0	4	90.17	119.23	1	25.39	-9.68	4.12	2	119.63	153.19	2	0.02
4	10	6	0	0	9.30	0	0	45	137.58	137.58	1	24.07	0	0	34	210.40	210.40	1	0.03
4	12	2	-46.82	0	TL	-39.65	13.46	1	-21.72	47.18	1	25.19	-39.78	13.22	1	15.22	116.65	1	0.02
4	12	3	-5.75	0	1184.18	-4.38	1.45	1	102.56	114.91	1	31.81	-4.72	1.09	15	68.50	78.77	1	0.02
4	12	4	-0.84	0	709.58	-0.84	0	2	81.99	83.54	1	34.79	0	0.85	18	97.69	99.37	2	0.02
4	12	6	-4.67	0	41.49	-4.67	0	37	121.88	132.75	1	32.78	-3.57	1.16	1	147.41	159.54	2	0.03
Average			-7.42	0	471.08	-4.85	3.04	17.05	102.81	114.86	1.05	22.64	-6.03	1.85	17.60	116.50	131.45	1.55	0.02

Table 7.4: Summary of Performance Measures of LM_3, H_1, and H_2 for Simulated Data Sets

Methods	$n = 40$					$n = 50$				
	$\%Gap_M$	$\%Gap_B$	$NOpt(58)$	$NBest$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$NOpt(54)$	$NBest$	$CPU\ s.$
LM_3	-4.07	0.03	58	59	259.55	-2.38	0.10	53	56	414.65
H_{1b}	-1.97	2.32	46	47	15.84	0.89	3.42	41	42	21.07
H_{1w}	78.02	84.52	9	9		75.01	78.66	10	10	
H_{2b}	-3.11	1.21	42	42	0.02	-1.74	0.77	38	40	0.02
H_{2w}	77.60	85.06	7	7		83.89	88.10	8	8	

Methods	$n = 80$					$n = 100$				
	$\%Gap_M$	$\%Gap_B$	$NOpt(49)$	$NBest$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$NOpt(47)$	$NBest$	$CPU\ s.$
LM_3	-1.20	0.97	49	49	366.31	-1.19	0.58	47	51	672.47
H_{1b}	2.32	4.49	42	46	35.77	3.22	5.02	40	43	50.55
H_{1w}	75.75	78.61	9	9		75.91	78.15	10	10	
H_{2b}	-1.24	0.95	41	47	0.05	-1.31	0.47	41	48	0.05
H_{2w}	81.92	85.10	9	9		77.11	79.84	10	10	

Methods	$n = 200$				
	$\%Gap_M$	$NOpt(19)$	$\%Gap_B$	$NBest$	$CPU\ s.$
LM_3	42.43	43.93	19	20	2153.03
H_{1b}	4.71	5.98	17	46	213.64
H_{1w}	62.92	64.43	7	13	
H_{2b}	-1.05	0.27	19	52	0.30
H_{2w}	94.57	96.48	6	7	

Table 7.5: Summary of Performance Measures of H_1 and H_2 for Simulated Data Sets

Methods	$n = 500$					$n = 1000$				
	$\%Gap_M$	$\%Gap_B$	$NBest$	NTL	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$NBest$	NTL	$CPU\ s.$
H_{1b}	7.04	8.10	42	0	1655.15	7.06	7.90	43	11	2757.11
H_{1w}	68.96	70.18	11			59.48	60.41	16		
H_{2b}	-0.64	0.37	57	0	2.25	-0.64	0.20	58	0	9.46
H_{2w}	88.69	90.17	6			96.80	97.97	7		

It is observed that model **LM$_3$** finds the best available solution in most of the data instances for the data sets which includes small number of data points. This performance decreases sharply when we look at the data set with the highest number of data points. The same trend is also shown in the other performance measures. Computational time and percent gaps from both made objective and best available solution increases with the increase in the data points.

When $NBest$ is considered, **H$_1$** found the best available solution in many of the data instances when its best clustering solutions are considered, **H$_{1b}$**. $NBest$ for the **H$_{1w}$** corresponds to the number of data instances where even in the worst case **H$_1$** finds the best available solution. It can be said that **H$_1$** behaves as the mathematical model **LM$_3$** does in terms of increase in the computational time and percent gaps when data sets get larger. With the increase in the data points in a data set, averages of $\%Gap_M$ and $\%Gap_B$ increase as well, whereas there is no such change in $NBest$. That means, when **H$_1$** could not find the best available solution, its deviations are higher in the data sets with more data points. Comparing average $CPU\ s.$ of **H$_1$** with model **LM$_3$**'s, the former can find clustering solutions of the data instances within at most four minutes, whereas the latter takes approximately 10 minutes on the average if it could solve the problem within the given time limit. Even the latter could not find clustering solutions within the time limit in most of the data instances when data sets get larger.

If we add **H$_2$** to the comparison, it is seen that its biggest advantage is the computational time. In our biggest simulated data set, it only takes less than one second on the average to obtain clustering solution. That means, the increase in the size of the data set does not affect $CPU\ s.$ of **H$_2$** too much. $NBest$ increases and its percent gap from best available solution decreases with the increase in size. Therefore, we can say that its solution is used as the best available solution among the clustering solutions obtained from the other solution methods. In order to further compare the performances of heuristic algorithms **H$_1$** and **H$_2$**, additional experimentation has been conducted, and result are given in Table 7.5. Here, it can be said that **H$_2$** finds almost all best available solutions which are obtained by heuristic algorithms. **H$_1$** hits to the time limit in 11 data instances among 60 when data set includes 1000 data points. However, **H$_2$** still find the solutions in less than 10 seconds on the average. Paired t-test at 5% significance level is applied to see if the mean difference between objective

functions of different approaches is significantly different. The results show that there is a significant difference between both heuristic algorithms, and heuristic algorithms and $\mathbf{LM_3}$.

CHAPTER 8

CONCLUSION

Clustering problem has been extensively discussed in variety of disciplines. It can be briefly described as the grouping of similar data points in the same clusters while separating them from the dissimilar data points. Selection of similarity measure and objective function, and also size of the data set may affect the performance of a clustering algorithm.

In this thesis, we address the clustering problem with cluster based feature selection. For the specified problem, center-based clustering is applied where each data point is assigned to only one of the clusters, and cluster centers are selected among the data points in that cluster. For the defined clustering problem, we work with data sets which include only continuous features in different number of features, clusters and data points. We show the analogy between classical *p-median* and clustering problems, and a nonlinear mixed integer programming model has been proposed. The study also includes three linearized model with different properties. All of those models ensure to find (i) the locations of cluster centers, (ii) features to be selected for each cluster, and (iii) assignment of data points to a cluster simultaneously. Number of clusters that will be constructed is given a priori as well as number of features to be selected for each cluster. As other partitional clustering methods, we aim to minimize total distance between the data points and the cluster centers. Different from traditional clustering algorithms, we are also performing feature selection that will provide relevant features for each cluster. Hence, distances are calculated using only selected features.

Number of features, data points and clusters in a data set affect the dimensionality of the data. When it gets larger, the solution time of the proposed mathematical models

gets worse. Therefore, in the second part of the study, Benders Decomposition approach is applied to our problem as an exact approach, and two heuristic algorithms have been proposed. We divide our problem into two subproblems in Benders Decomposition. Location of cluster centers and selection of features are decided in the master problem, whereas the assignments are decided in the subproblem. The experimental studies show that it is not beneficial to use Benders Decomposition, but the idea of decomposing the problem into subproblems is used in our heuristic algorithms. We propose a Benders like Heuristic Algorithm ($\mathbf{H_1}$) which uses a new mathematical model to only decide cluster centers and relevant features of the clusters. When the cluster centers and relevant features are fixed, the objective function will be minimized by assigning each data point to its closest cluster center. For finding the closest center a simple procedure is used, Assignment Problem ($\mathbf{P_X}$).

It has been observed that when fixing two of the decision variables the solution of the remaining problem will be trivial. Therefore, our problem can be divided into three main problems. When the cluster centers and relevant features are fixed, each data point should be assigned to its closest cluster center, Assignment Problem ($\mathbf{P_X}$). Center Selection Problem ($\mathbf{P_C}$) decides cluster centers when constructed groups and relevant features are fixed. In this case, cluster centers will be the data point which is located in the middle of the cluster. In order to find relevant features of each cluster, Feature Selection Problem ($\mathbf{P_Q}$) sorts features in the ascending order of compactness and select Q most compact features. Iterative Heuristic Algorithm ($\mathbf{H_2}$), decides each decision variable by solving the defined smaller problems iteratively.

The experiments are conducted on simulated data sets which differ in terms of number of data points, features, and clusters. All data sets are generated as they reflect discussed clustering structure where clusters are located in different dimensions and global feature selection may not ensure to construct all clusters. Data has been generated by using Multivariate Normal Distribution for relevant features and Uniform Distribution for irrelevant features. By this way, dense clusters along relevant features has been obtained, whereas they are scattered through irrelevant features. The empirical results show that optimal solutions cannot be obtained within the given time limit by using mathematical models in most of the data sets. Since the Benders like Heuristic Algorithm ($\mathbf{H_1}$) uses mathematical model to obtain cluster center

and select features, it takes more time to converge than Iterative Heuristic Algorithm ($\mathbf{H_2}$). We applied paired-t test at 5% significance level to test if the mean difference of the objective function obtained by different approaches is significantly different from zero. It is observed that there is no significant difference between models $\mathbf{LM_2}$ and $\mathbf{LM_3}$, whereas they are significantly different than the other two mathematical models. Also, heuristics algorithms perform significantly different than the mathematical models especially when the size of the data set increases, and their performances are different than each other.

To the best of our knowledge, there is no study modeling clustering and cluster based feature selection as a mixed integer programming. It should also be noted that the proposed solution approaches may select same features for different clusters. Therefore, all approaches can also be used in classical feature selection problem.

The data sets used in this study are only include continuous features and clusters are all in the spherical shapes since we are using identity matrix in the formation of data sets. Proposed algorithms can be analyzed in different data sets having arbitrary shaped clusters with density differences or having categorical features. In this study, $L_1 - norm$ is used as a similarity measure. Another future research issue may be analyzing the performance of proposed mathematical models and heuristic algorithms in different distance measures. Also, the number of features that should be selected may be considered as a decision variable instead of parameters in different problem formulations.

The focus of this study is to maximize the compactness by decreasing the distance between data points assigned to same cluster. We may also analyze our approaches using different objectives such as maximization of separation between clusters or maximization of the ratio between separation and compactness.

Different valid inequalities can be added to Benders Decomposition solution method to improve its performance. Also, with the some preliminary studies, we have observed that dual variables of the subproblem may be directly found using strong duality and Karush–Kuhn–Tucker optimality conditions. If it is valid in all conditions, there will be no need to solve mathematical model to obtain dual variables and generate cut. We will also focus on these observations as a future work.

REFERENCES

Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., and Park, J. S. (1999). Fast algorithms for projected clustering. *ACM SIGMoD Record*, 28:61–72.

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). *Automatic subspace clustering of high dimensional data for data mining applications*. ACM.

Alelyani, S., Tang, J., and Liu, H. (2013). Feature selection for clustering: A review. In *Data Clustering*, pages 29–60. Chapman and Hall/CRC.

Andrews, J. L. and McNicholas, P. D. (2014). Variable selection for clustering and classification. *Journal of Classification*, 31(2):136–153.

Ben-Israel, A. and Iyigun, C. (2008). Probabilistic d-clustering. *Journal of Classification*, 25(1):5–26.

Benati, S. and García, S. (2014). A mixed integer linear model for clustering with variable selection. *Computers & Operations Research*, 43:280–285.

Benati, S., García, S., and Puerto, J. (2018). Mixed integer linear programming and heuristic methods for feature selection in clustering. *Journal of the Operational Research Society*, 69(9):1379–1395.

Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2–3):191–203.

Bradley, P. S., Mangasarian, O. L., and Street, W. N. (1996). Clustering via concave minimization. In *Advances in neural information processing systems*, pages 368–374.

Brusco, M. J. (2003). An enhanced branch-and-bound algorithm for a partitioning problem. *British Journal of Mathematical and Statistical Psychology*, 56(1):83–92.

Brusco, M. J. (2004). Clustering binary data in the presence of masking variables. *Psychological Methods*, 9(4):510.

Brusco, M. J. and Cradit, J. D. (2001). A variable-selection heuristic for k-means clustering. *Psychometrika*, 66(2):249–270.

Brusco, M. J. and Köhn, H.-F. (2008). Optimal partitioning of a data set based on the p-median model. *Psychometrika*, 73(1):89.

Brusco, M. J. and Stahl, S. (2006). *Branch-and-bound applications in combinatorial data analysis*. Springer Science & Business Media.

Cai, J., Luo, J., Wang, S., and Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79.

Casasent, D. P. and Chen, X.-W. (2003). Waveband selection for hyperspectral data: optimal feature selection. *Optical Pattern Recognition XIV*, 5106:259–270.

Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

Chen, X.-w. (2003). An improved branch and bound algorithm for feature selection. *Pattern Recognition Letters*, 24(12):1925–1933.

Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. *Ismb*, 8:93–103.

Diao, R. and Shen, Q. (2015). Nature inspired feature selection meta-heuristics. *Artificial Intelligence Review*, 44(3):311–340.

Dy, J. G. and Brodley, C. E. (2004). Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5(Aug):845–889.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., Foufou, S., and Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Transactions on Emerging Topics in Computing*, 2(3):267–279.

Friedman, J. H. and Meulman, J. J. (2004). Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(4):815–849.

Frigui, H. and Nasraoui, O. (2004). Unsupervised learning of prototypes and attribute weights. *Pattern Recognition*, 37(3):567–581.

García, S., Labbé, M., and Marín, A. (2011). Solving large p-median problems with a radius formulation. *INFORMS Journal on Computing*, 23(4):546–556.

Guha, S., Rastogi, R., and Shim, K. (1998). Cure: an efficient clustering algorithm for large databases. *ACM Sigmod Record*, 27:73–84.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323.

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

Kim, J., Yang, J., and Ólafsson, S. (2009). An optimization approach to partitional data clustering. *Journal of the Operational Research Society*, 60(8):1069–1084.

Klein, G. and Aronson, J. E. (1991). Optimal clustering: A model and method. *Naval Research Logistics (NRL)*, 38(3):447–461.

Koontz, W. L. G., Narendra, P. M., and Fukunaga, K. (1975). A branch and bound clustering algorithm. *IEEE Transactions on Computers*, 100(9):908–915.

Kriegel, H.-P., Kröger, P., and Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1–58.

Law, M. H., Figueiredo, M. A., and Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166.

Li, H.-L. (1994). A global approach for general 0–1 fractional programming. *European Journal of Operational Research*, 73(3):590–596.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1–14:281–297.

Mulvey, J. M. and Crowder, H. P. (1979). Cluster analysis: An application of lagrangian relaxation. *Management Science*, 25(4):329–340.

Nakariyakul, S. and Casasent, D. P. (2007). Adaptive branch and bound algorithm for selecting optimal features. *Pattern Recognition Letters*, 28(12):1415–1427.

Nanda, S. J. and Panda, G. (2014). A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary computation*, 16:1–18.

Narendra, P. M. and Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, C-26(9):917–922.

Olafsson, S., Li, X., and Wu, S. (2008). Operations research and data mining. *European Journal of Operational Research*, 187(3):1429–1448.

Ólafsson, S. and Yang, J. (2005). Intelligent partitioning for feature selection. *INFORMS Journal on Computing*, 17(3):339–355.

Padilha, V. A. and Campello, R. J. (2017). A systematic comparative evaluation of biclustering techniques. *BMC Bioinformatics*, 18(1):55.

Park, H.-S. and Jun, C.-H. (2009). A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341.

Pontes, B., Giráldez, R., and Aguilar-Ruiz, J. S. (2015). Biclustering on expression data: A review. *Journal of Biomedical Informatics*, 57:163–180.

Rao, M. (1971). Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, 66(335):622–626.

Reynolds, A. P., Richards, G., de la Iglesia, B., and Rayward-Smith, V. J. (2006). Clustering rules: a comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms*, 5(4):475–504.

Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

Sağlam, B., Salman, F. S., Sayın, S., and Türkay, M. (2006). A mixed-integer programming approach to the clustering problem with an application in customer segmentation. *European Journal of Operational Research*, 173(3):866–879.

Shi, L. and Ólafsson, S. (2000). Nested partitions method for global optimization. *Operations Research*, 48(3):390–407.

Somol, P., Pudil, P., and Kittler, J. (2004). Fast branch & bound algorithms for optimal feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):900–912.

Steinley, D. and Brusco, M. J. (2008). A new variable weighting and selection procedure for k-means cluster analysis. *Multivariate Behavioral Research*, 43(1):77–108.

Teitz, M. B. and Bart, P. (1968). Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16(5):955–961.

Vinod, H. D. (1969). Integer programming and the theory of grouping. *Journal of the American Statistical Association*, 64(326):506–519.

Xu, R. and Wunsch, D. C. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678.

Xue, B., Zhang, M., Browne, W. N., and Yao, X. (2015). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626.

Yang, J. and Olafsson, S. (2006). Optimization-based feature selection with adaptive instance sampling. *Computers & Operations Research*, 33(11):3088–3106.

Yu, B. and Yuan, B. (1993). A more efficient branch and bound algorithm for feature selection. *Pattern Recognition*, 26(6):883–889.

Zadegan, S. M. R., Mirzaie, M., and Sadoughi, F. (2013). Ranked k-medoids: A fast and accurate rank-based partitioning algorithm for clustering large datasets. *Knowledge-Based Systems*, 39:133–143.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. *ACM Sigmod Record*, 25:103–114.

APPENDIX A

EXPERIMENTAL RESULTS OF PROPOSED MATHEMATICAL MODELS

In this appendix, we give experimental results of the proposed mathematical models for each problem instances.

Performance measures used in all tables are explained below.

$\%Gap_M$: Percent gap from made objective
$\%Gap_O$: Percent gap from optimal solution
$\%Gap_B$: Percent gap from best available solution
$CPUs.$: Computational time

Table A.1: Results of Experimental Studies for Simulated Data Sets with 50 Data Points and 2 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
2	4	2	-2.68	0	0	147.17	-2.68	0	0	114.30	-2.68	0	0	11.45	-2.68	0	0	8.10
2	4	3	0	0	0	27.01	0	0	0	30.35	0	0	0	4.00	0	0	0	3.08
2	5	2	0	0	0	201.41	0	0	0	227.40	0	0	0	2.42	0	0	0	2.20
2	5	3	0	0	0	38.38	0	0	0	71.85	0	0	0	3.41	0	0	0	2.23
2	5	4	0	0	0	38.41	0	0	0	27.06	0	0	0	6.71	0	0	0	4.69
2	6	2	-1.95	0	0	309.20	-1.95	0	0	498.06	-1.95	0	0	8.15	-1.95	0	0	8.03
2	6	3	0	0	0	330.97	0	0	0	300.94	0	0	0	5.93	0	0	0	3.96
2	6	4	0	0	0	104.38	0	0	0	94.83	0	0	0	9.18	0	0	0	8.86
2	8	2	-3.21	0	0	1178.15	-3.21	0	0	2112.54	-3.21	0	0	40.92	-3.21	0	0	30.56
2	8	3	0	0	0	1667.76	0	0	0	1325.81	0	0	0	21.64	0	0	0	12.96
2	8	4	0	0	0	1479.98	0	0	0	1116.68	0	0	0	23.57	0	0	0	24.68
2	8	6	0	0	0	163.06	0	0	0	333.52	0	0	0	23.61	0	0	0	23.32
2	10	2	0	0	0	2967.92	0	0	0	2413.05	0	0	0	13.03	0	0	0	8.81
2	10	3	0	32.66	0	TL	0	0	0	3551.30	0	0	0	127.65	0	0	0	124.17
2	10	4	0	0	0	4618.34	5.59	38.31	5.59	TL	0	0	0	86.67	0	0	0	63.35
2	10	6	0	0	0	1027.27	0	0	0	731.61	0	0	0	77.71	0	0	0	74.50
2	12	2	-0.19	0	0	5370.76	-0.19	54.51	0	TL	-0.19	0	0	12.94	-0.19	0	0	10.41
2	12	3	0	0	0	5678.43	0	100.00	0	TL	0	0	0	30.79	0	0	0	28.88
2	12	4	0	60.01	0	TL	0	71.15	0	TL	0	0	0	104.26	0	0	0	123.84
2	12	6	0	34.56	0	TL	0	23.15	0	TL	0	0	0	37.57	0	0	0	54.56
Average			-0.40	6.36	0	1491.09	-0.12	14.36	0.28	863.29	-0.40	0	0	32.58	-0.40	0	0	31.06

Table A.2: Results of Experimental Studies for Simulated Data Sets with 50 Data Points and 3 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
3	4	2	-3.52	0	0	2426.17	-3.52	0	0	3915.40	-3.52	0	0	14.57	-3.52	0	0	12.49
3	4	3	0	0	0	117.21	0	0	0	133.96	0	0	0	3.32	0	0	0	2.36
3	5	2	-0.47	0	0	7090.55	-0.47	0	0	6040.11	-0.47	0	0	14.13	-0.47	0	0	9.55
3	5	3	0	0	0	1693.92	0	0	0	2068.47	0	0	0	5.06	0	0	0	3.06
3	5	4	0	0	0	197.55	0	0	0	268.03	0	0	0	6.07	0	0	0	4.28
3	6	2	-0.64	100.00	0	TL	-0.64	100.00	0	TL	-0.64	0	0	31.18	-0.64	0	0	22.26
3	6	3	-2.47	79.43	0	TL	-2.47	77.31	0	TL	-2.47	0	0	17.12	-2.47	0	0	10.89
3	6	4	0	0	0	1205.91	0	0	0	1396.73	0	0	0	9.14	0	0	0	6.05
3	8	2	-4.52	100.00	0	TL	-4.52	100.00	0	TL	-4.52	0	0	376.71	-4.52	0	0	500.96
3	8	3	0	100.00	0	TL	0	100.00	0	TL	0	0	0	85.96	0	0	0	69.89
3	8	4	0	100.00	0	TL	0	100.00	0	TL	0	0	0	56.94	0	0	0	47.03
3	8	6	-0.44	0	0	744.83	-0.44	0	0	1197.01	-0.44	0	0	17.07	-0.44	0	0	13.66
3	10	2	-6.82	100.00	2.11	TL	-8.74	100.00	0	TL	-8.24	11.49	0.55	TL	-8.00	12.12	0.81	TL
3	10	3	-1.01	100.00	0.08	TL	-1.08	100.00	0	TL	1.17	12.41	2.28	TL	-1.08	11.10	0	TL
3	10	4	0.78	100.00	0.78	TL	0.01	100.00	0.01	TL	0	0	0	83.32	0	0	0	74.67
3	10	6	-1.12	78.89	0	TL	-1.12	77.86	0	TL	-1.12	0	0	33.45	-1.12	0	0	34.22
3	12	2	-8.68	100.00	0	TL	-8.68	100.00	0	TL	-7.42	20.30	1.39	TL	-7.64	19.39	1.14	TL
3	12	3	1.89	100.00	1.89	TL	0	100.00	0	TL	0	0	0	448.84	0	0	0	551.89
3	12	4	-2.61	100.00	0	TL	0.34	100.00	3.04	TL	-2.61	0	0	6828.38	-2.61	6.46	0	TL
3	12	6	-1.78	100.00	0	TL	2.11	100.00	3.96	TL	-1.78	0	0	201.24	-1.78	0	0	263.35
Average			-1.57	62.92	0.24	1925.16	-1.46	62.76	0.35	2145.67	-1.60	2.21	0.21	484.26	-1.71	2.45	0.10	519.23

Table A.3: Results of Experimental Studies for Simulated Data Sets with 50 Data Points and 4 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
4	4	2	0	100.00	0	TL	0	100.00	0	TL	0	0	0	10.33	0	0	0	6.28
4	4	3	0	0	0	2242.62	0	0	0	2260.89	0	0	0	4.76	0	0	0	3.09
4	5	2	0	100.00	0	TL	0	100.00	0	TL	0	0	0	11.96	0	0	0	8.17
4	5	3	0	85.88	0	TL	0	87.55	0	TL	0	0	0	8.65	0	0	0	5.72
4	5	4	0	0	0	1519.24	0	0	0	2828.90	0	0	0	5.23	0	0	0	4.18
4	6	2	-5.00	100.00	0.24	TL	-5.00	100.00	0.24	TL	-5.23	0	0	801.85	-5.23	0	0	2362.39
4	6	3	-4.71	100.00	0	TL	-4.71	100.00	0	TL	-4.71	0	0	29.62	-4.71	0	0	34.03
4	6	4	-0.85	83.79	0	TL	-0.85	79.74	0	TL	-0.85	0	0	9.50	-0.85	0	0	7.12
4	8	2	-26.84	100.00	0	TL	-26.84	100.00	0	TL	-26.84	0	0	346.12	-26.84	0	0	249.95
4	8	3	-0.55	100.00	0.97	TL	-0.55	100.00	0.97	TL	-1.51	0	0	261.24	-1.51	0	0	273.52
4	8	4	2.51	100.00	4.52	TL	-1.60	100.00	0.33	TL	-1.92	0	0	26.66	-1.92	0	0	23.48
4	8	6	-1.69	67.60	0	TL	-1.69	71.22	0	TL	-1.69	0	0	16.24	-1.69	0	0	10.69
4	10	2	-20.35	100.00	0	TL	-19.78	100.00	0.72	TL	-20.35	7.20	0	TL	-20.35	0	0	6572.27
4	10	3	-2.04	100.00	0	TL	3.41	100.00	5.57	TL	0.57	20.44	2.67	TL	-0.40	20.15	1.68	TL
4	10	4	3.14	100.00	3.58	TL	9.58	100.00	10.05	TL	-0.43	0	0	80.68	-0.43	0	0	76.85
4	10	6	4.62	100.00	6.67	TL	-1.92	100.00	0	TL	-1.92	0	0	41.90	-1.92	0	0	35.98
4	12	2	-26.47	100.00	0	TL	-26.47	100.00	0	TL	-25.28	22.89	1.62	TL	-24.67	22.96	2.44	TL
4	12	3	2.90	100.00	3.02	TL	2.44	100.00	2.56	TL	1.63	8.60	1.75	TL	-0.12	7.86	0	TL
4	12	4	1.87	100.00	9.62	TL	1.49	100.00	9.20	TL	-7.07	0	0	1200.05	-7.07	0	0	3131.26
4	12	6	8.13	100.00	11.30	TL	2.68	100.00	5.70	TL	-2.85	0	0	102.78	-2.85	0	0	138.32
Average			-3.27	86.86	2.00	1880.93	-3.49	86.93	1.77	2544.90	-4.92	2.96	0.30	184.85	-5.03	2.55	0.21	761.37

Table A.4: Results of Experimental Studies for Simulated Data Sets with 80 Data Points and 2 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
2	4	2	0	0	0	1158.19	0	0	0	1211.64	0	0	0	9.19	0	0	0	7.44
2	4	3	0	0	0	171.02	0	0	0	260.30	0	0	0	7.62	0	0	0	8.41
2	5	2	0	0	0	2388.23	0	0	0	5426.07	0	0	0	15.59	0	0	0	15.27
2	5	3	0	0	0	540.94	0	0	0	775.26	0	0	0	70.40	0	0	0	64.90
2	5	4	0	0	0	204.37	0	0	0	197.19	0	0	0	14.34	0	0	0	11.99
2	6	2	-3.11	0	0	5697.87	-3.11	0	0	6007.65	-3.11	0	0	62.18	-3.11	0	0	35.16
2	6	3	0	0	0	4423.94	0	0	0	3189.63	0	0	0	31.04	0	0	0	20.53
2	6	4	0	0	0	481.77	0	0	0	599.77	0	0	0	36.95	0	0	0	31.95
2	8	2	-1.24	100.00	0.83	TL	-1.03	100.00	1.03	TL	-2.05	0	0	366.46	-2.05	0	0	314.79
2	8	3	0	100.00	0	TL	0	100.00	0	TL	0	0	0	115.99	0	0	0	99.50
2	8	4	-0.06	100.00	0	TL	3.68	100.00	3.74	TL	-0.06	0	0	153.16	-0.06	0	0	140.65
2	8	6	0	0	0	766.76	0	0	0	926.53	0	0	0	194.60	0	0	0	169.49
2	10	2	0	100.00	0	TL	0	100.00	0	TL	0	0	0	42.79	0	0	0	31.91
2	10	3	4.69	100.00	4.69	TL	4.69	100.00	4.69	TL	0	0	0	320.27	0	0	0	366.91
2	10	4	6.46	100.00	6.46	TL	0	100.00	0	TL	0	0	0	580.63	0	0	0	560.25
2	10	6	0	28.61	0	TL	0	48.64	0	TL	0	0	0	468.80	0	0	0	372.94
2	12	2	-1.09	100.00	5.16	TL	-5.63	100.00	0.33	TL	-5.94	0	0	717.82	-5.94	0	0	775.30
2	12	3	3.51	100.00	3.51	TL	2.81	100.00	2.81	TL	0	0	0	498.05	0	0	0	333.92
2	12	4	1.96	100.00	1.96	TL	8.57	100.00	8.57	TL	0	0	0	873.72	0	0	0	1166.27
2	12	6	7.81	100.00	7.81	TL	9.51	100.00	9.51	TL	0	0	0	574.03	0	0	0	604.21
Average			0.95	51.43	1.52	1759.23	0.97	52.43	1.53	2066.01	-0.56	0	0	257.68	-0.56	0	0	256.59

Table A.5: Results of Experimental Studies for Simulated Data Sets with 80 Data Points and 3 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
3	4	2	-2.25	100.00	0	TL	-0.89	100.00	1.39	TL	-2.25	0	0	131.30	-2.25	0	0	143.80
3	4	3	0	0	0	825.09	0	0	0	1153.13	0	0	0	18.29	0	0	0	16.68
3	5	2	-0.92	100.00	1.73	TL	0.17	100.00	2.85	TL	-2.60	0	0	105.07	-2.60	0	0	61.17
3	5	3	1.38	100.00	1.38	TL	0	100.00	0	TL	0	0	0	21.20	0	0	0	12.47
3	5	4	0	0	0	903.79	0	0	0	1235.39	0	0	0	34.66	0	0	0	20.33
3	6	2	-1.37	100.00	0.05	TL	-0.63	100.00	0.81	TL	-1.42	0	0	242.67	-1.42	0	0	162.80
3	6	3	2.89	100.00	5.13	TL	-2.13	100.00	0	TL	-2.13	0	0	119.69	-2.13	0	0	74.50
3	6	4	0	68.16	0	TL	4.07	100.00	4.07	TL	0	0	0	70.46	0	0	0	37.59
3	8	2	-11.39	100.00	1.28	TL	-9.05	100.00	3.96	TL	-12.51	0	0	2587.13	-12.51	0	0	2128.34
3	8	3	7.67	100.00	7.67	TL	11.10	100.00	11.10	TL	0	0	0	1760.42	0	0	0	2504.22
3	8	4	4.06	100.00	4.06	TL	16.59	100.00	16.59	TL	0	0	0	240.47	0	0	0	243.02
3	8	6	0	23.84	0	TL	0.44	25.02	0.44	TL	0	0	0	101.02	0	0	0	91.94
3	10	2	-8.33	100.00	0	TL	-2.18	100.00	6.71	TL	-7.94	26.22	0.42	TL	-7.92	24.33	0.44	TL
3	10	3	14.20	100.00	15.36	TL	1.47	100.00	2.51	TL	-1.01	4.46	0	TL	1.84	13.07	2.87	TL
3	10	4	4.02	100.00	4.02	TL	11.18	100.00	11.18	TL	0	0	0	502.19	0	0	0	537.13
3	10	6	10.78	100.00	10.78	TL	15.27	100.00	15.27	TL	0	0	0	226.12	0	0	0	221.55
3	12	2	-20.22	100.00	1.72	TL	-19.61	100.00	2.49	TL	-21.57	27.79	0	TL	-17.79	31.29	4.82	TL
3	12	3	10.32	100.00	12.69	TL	15.57	100.00	18.05	TL	-2.11	10.55	0	TL	-2.07	11.05	0.04	TL
3	12	4	17.78	100.00	17.48	TL	12.55	100.00	12.25	TL	0.26	12.87	0	TL	0.53	13.36	0.27	TL
3	12	6	13.82	100.00	13.82	TL	33.96	100.00	33.96	TL	0	0	0	545.53	0	0	0	533.00
Average			2.12	84.60	4.86	864.44	4.40	86.25	7.18	1194.26	-2.66	4.09	0.02	447.08	-2.32	4.65	0.42	452.57

Table A.6: Results of Experimental Studies for Simulated Data Sets with 80 Data Points and 4 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
4	4	2	1.08	100.00	1.08	TL	3.04	100.00	3.04	TL	0	0	0	39.67	0	0	0	41.47
4	4	3	0	75.77	0	TL	0	77.05	0	TL	0	0	0	21.00	0	0	0	20.95
4	5	2	-3.75	100.00	0.51	TL	-1.73	100.00	2.63	TL	-4.24	0	0	194.90	-4.24	0	0	121.41
4	5	3	4.11	100.00	4.11	TL	3.75	100.00	3.75	TL	0	0	0	28.71	0	0	0	17.69
4	5	4	0	47.45	0	TL	0	43.79	0	TL	0	0	0	23.38	0	0	0	12.90
4	6	2	1.73	100.00	3.75	TL	1.09	100.00	3.09	TL	-1.95	0	0	417.30	-1.95	0	0	380.52
4	6	3	9.34	100.00	9.34	TL	7.02	100.00	7.02	TL	0	0	0	98.35	0	0	0	69.53
4	6	4	4.46	100.00	4.46	TL	4.93	100.00	4.93	TL	0	0	0	59.51	0	0	0	32.53
4	8	2	-7.21	100.00	2.16	TL	-9.17	100.00	0	TL	-6.26	22.13	3.20	TL	-4.29	25.28	5.37	TL
4	8	3	17.80	100.00	17.80	TL	14.74	100.00	14.74	TL	0	0	0	3592.82	0	0	0	3538.41
4	8	4	22.65	100.00	22.85	TL	14.40	100.00	14.59	TL	-0.16	0	0	223.63	-0.16	0	0	231.76
4	8	6	2.38	100.00	2.38	TL	7.21	100.00	7.21	TL	0	0	0	85.15	0	0	0	73.99
4	10	2	-7.09	100.00	3.50	TL	-7.01	100.00	3.59	TL	-10.23	22.92	0	TL	-10.13	23.84	0.11	TL
4	10	3	13.17	100.00	12.90	TL	8.14	100.00	7.88	TL	4.25	20.31	4.01	TL	0.23	18.13	0	TL
4	10	4	11.44	100.00	11.44	TL	24.63	100.00	24.63	TL	0	0	0	544.24	0	0	0	767.25
4	10	6	11.03	100.00	11.03	TL	8.05	100.00	8.05	TL	0	0	0	193.90	0	0	0	162.85
4	12	2	-16.82	100.00	11.02	TL	-14.85	100.00	13.65	TL	-25.07	25.38	0	TL	-22.90	25.16	2.90	TL
4	12	3	20.71	100.00	7.01	TL	13.52	100.00	0.64	TL	12.80	28.34	0	TL	25.74	37.53	11.47	TL
4	12	4	20.28	100.00	15.45	TL	36.74	100.00	31.25	TL	4.18	9.63	0	TL	6.86	13.05	2.57	TL
4	12	6	11.92	100.00	15.98	TL	16.63	100.00	20.86	TL	-3.50	0	0	617.52	-3.50	0	0	557.76
Average			5.86	96.16	7.84	-	6.56	96.04	8.58	-	-1.51	6.44	0.36	438.58	-0.72	7.15	1.12	430.64

Table A.7: Results of Experimental Studies for Simulated Data Sets with 100 Data Points and 2 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
2	4	2	-0.40	0	0	5316.52	-0.40	0	0	6059.82	-0.40	0	0	31.74	-0.40	0	0	23.84
2	4	3	0	0	0	372.89	0	0	0	371.57	0	0	0	31.30	0	0	0	23.20
2	5	2	0.29	100.00	0.29	TL	0	100.00	0	TL	0	0	0	55.49	0	0	0	30.20
2	5	3	0	0	0	3112.37	0	0	0	3100.03	0	0	0	131.94	0	0	0	67.78
2	5	4	0	0	0	1017.13	0	0	0	644.25	0	0	0	26.50	0	0	0	32.16
2	6	2	0.73	100.00	0.73	TL	0	100.00	0	TL	0	0	0	77.89	0	0	0	82.44
2	6	3	2.23	100.00	2.23	TL	0	100.00	0	TL	0	0	0	184.04	0	0	0	114.26
2	6	4	0	0	0	1364.95	0	0	0	1564.85	0	0	0	284.81	0	0	0	141.91
2	8	2	-0.59	100.00	0.44	TL	0.63	100.00	1.68	TL	-1.03	0	0	549.36	-1.03	0	0	481.76
2	8	3	1.06	100.00	1.06	TL	1.57	100.00	1.57	TL	0	0	0	382.74	0	0	0	282.91
2	8	4	0	100.00	0	TL	12.17	100.00	12.17	TL	0	0	0	290.50	0	0	0	130.80
2	8	6	0	0	0	1882.95	0	0	0	1913.50	0	0	0	399.94	0	0	0	399.45
2	10	2	-2.54	100.00	1.35	TL	2.42	100.00	6.52	TL	-3.84	0	0	835.86	-3.84	0	0	479.91
2	10	3	3.81	100.00	3.81	TL	7.28	100.00	7.28	TL	0	0	0	1684.92	0	0	0	1037.66
2	10	4	16.08	100.00	16.08	TL	15.13	100.00	15.13	TL	0	0	0	1028.03	0	0	0	1250.38
2	10	6	0.60	44.87	0.60	TL	0	45.18	0	TL	0	0	0	857.04	0	0	0	972.61
2	12	2	1.52	100.00	1.52	TL	7.45	100.00	7.45	TL	0	0	0	952.38	0	0	0	517.76
2	12	3	25.54	100.00	25.54	TL	27.56	100.00	27.56	TL	3.28	5.89	3.28	TL	0	0	0	4428.47
2	12	4	10.57	100.00	10.57	TL	25.67	100.00	25.67	TL	0	0	0	2694.86	0	0	0	2698.05
2	12	6	19.84	100.00	19.84	TL	17.93	100.00	17.93	TL	0	0	0	1488.54	0	0	0	1714.60
Average			3.94	67.24	4.20	2177.80	5.87	67.26	6.15	2275.67	-0.10	0.29	0.16	630.94	-0.26	0	0	745.51

Table A.8: Results of Experimental Studies for Simulated Data Sets with 100 Data Points and 3 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
3	4	2	4.32	100.00	4.32	TL	6.08	100.00	6.08	TL	0	0	0	230.12	0	0	0	116.99
3	4	3	0	0	0	5066.99	0	0	0	5401.04	0	0	0	62.47	0	0	0	52.74
3	5	2	0.37	100.00	0.90	TL	1.27	100.00	1.80	TL	-0.52	0	0	386.99	-0.52	0	0	238.42
3	5	3	5.40	100.00	5.40	TL	5.35	100.00	5.35	TL	0	0	0	71.12	0	0	0	33.93
3	5	4	0	0	0	2766.35	0	0	0	3431.49	0	0	0	132.39	0	0	0	85.04
3	6	2	5.41	100.00	9.66	TL	9.94	100.00	14.37	TL	-3.87	0	0	1437.46	-3.87	0	0	1113.52
3	6	3	7.04	100.00	7.04	TL	2.03	100.00	2.03	TL	0	0	0	472.84	0	0	0	194.05
3	6	4	0	100.00	0	TL	4.69	100.00	4.69	TL	0	0	0	108.14	0	0	0	75.25
3	8	2	8.19	100.00	7.18	TL	6.19	100.00	5.20	TL	0.97	12.94	0.02	TL	0.94	13.30	0	TL
3	8	3	8.95	100.00	7.04	TL	25.25	100.00	23.06	TL	6.99	12.78	5.11	TL	1.78	8.02	0	TL
3	8	4	31.98	100.00	31.98	TL	7.48	100.00	7.48	TL	0	0	0	830.32	0	0	0	935.35
3	8	6	-0.87	57.61	0	TL	1.39	61.54	2.29	TL	-0.87	0	0	386.39	-0.87	0	0	317.36
3	10	2	0.73	100.00	8.46	TL	12.46	100.00	21.09	TL	-7.13	23.30	0	TL	-6.57	23.57	0.60	TL
3	10	3	32.47	100.00	33.04	TL	23.67	100.00	24.20	TL	-0.43	7.57	0	TL	-0.43	7.08	0	TL
3	10	4	26.71	100.00	26.71	TL	46.18	100.00	46.18	TL	0	0	0	1603.40	0	0	0	1576.53
3	10	6	11.47	100.00	11.47	TL	12.93	100.00	12.93	TL	0	0	0	791.57	0	0	0	825.50
3	12	2	-5.71	100.00	11.35	TL	-10.98	100.00	5.13	TL	-14.68	29.39	0.77	TL	-15.33	28.49	0	TL
3	12	3	45.96	100.00	45.52	TL	33.92	100.00	33.51	TL	0.30	10.79	0	TL	3.78	14.00	3.47	TL
3	12	4	16.42	100.00	15.06	TL	20.39	100.00	18.98	TL	1.19	15.41	0	TL	5.76	18.86	4.52	TL
3	12	6	66.93	100.00	66.93	TL	13.61	100.00	13.61	TL	0	0	0	2048.16	0	0	0	2241.76
Average			13.29	87.88	14.60	3916.67	11.09	88.08	12.40	4416.27	-0.90	5.61	0.30	658.57	-0.77	5.67	0.43	600.49

Table A.9: Results of Experimental Studies for Simulated Data Sets with 100 Data Points and 4 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
4	4	2	7.47	100.00	7.47	TL	7.92	100.00	7.92	TL	0	0	0	126.03	0	0	0	81.16
4	4	3	1.00	100.00	1.00	TL	0.58	100.00	0.58	TL	0	0	0	76.60	0	0	0	62.65
4	5	2	21.83	100.00	22.15	TL	8.79	100.00	9.08	TL	-0.26	0	0	183.44	-0.26	0	0	130.15
4	5	3	28.26	100.00	28.26	TL	32.85	100.00	32.85	TL	0	0	0	197.08	0	0	0	116.12
4	5	4	0	57.91	0	TL	0	71.98	0	TL	0	0	0	66.17	0	0	0	41.94
4	6	2	15.82	100.00	18.87	TL	5.16	100.00	7.93	TL	-2.56	0	0	843.08	-2.56	0	0	541.84
4	6	3	8.74	100.00	10.09	TL	11.13	100.00	12.52	TL	-1.23	0	0	264.07	-1.23	0	0	176.16
4	6	4	8.44	100.00	8.44	TL	11.24	100.00	11.24	TL	0	0	0	194.54	0	0	0	124.09
4	8	2	-4.32	100.00	20.31	TL	-10.15	100.00	12.98	TL	-19.50	15.98	1.23	TL	-20.47	14.83	0	TL
4	8	3	20.75	100.00	20.75	TL	27.14	100.00	27.14	TL	0	0	0	2697.55	0	0	0	2803.20
4	8	4	83.72	100.00	83.72	TL	20.13	100.00	20.13	TL	0	0	0	797.14	0	0	0	513.96
4	8	6	28.44	100.00	28.44	TL	23.35	100.00	23.35	TL	0	0	0	392.56	0	0	0	377.77
4	10	2	-6.29	100.00	19.32	TL	2.63	100.00	30.68	TL	-19.50	20.40	2.50	TL	-21.46	19.98	0	TL
4	10	3	13.59	100.00	11.15	TL	18.38	100.00	15.84	TL	12.32	27.09	9.92	TL	2.19	19.76	0	TL
4	10	4	30.53	100.00	32.54	TL	45.60	100.00	47.84	TL	-1.52	0	0	1538.77	-1.52	0	0	1515.99
4	10	6	17.13	100.00	17.13	TL	17.83	100.00	17.83	TL	0	0	0	629.15	0	0	0	670.75
4	12	2	-2.37	100.00	25.80	TL	1.46	100.00	30.73	TL	-22.39	27.72	0	TL	-22.11	27.88	0.36	TL
4	12	3	34.58	100.00	14.30	TL	31.23	100.00	11.46	TL	25.42	40.75	6.52	TL	17.74	37.44	0	TL
4	12	4	51.58	100.00	51.03	TL	66.35	100.00	65.74	TL	1.22	6.37	0.85	TL	0.37	5.58	0	TL
4	12	6	49.74	100.00	52.31	TL	15.37	100.00	17.35	TL	-1.69	0	0	1755.78	-1.69	0	0	1733.80
Average			20.43	97.90	23.65	-	16.85	98.60	20.16	-	-1.48	6.92	1.05	697.28	-2.55	6.27	0.02	634.97

Table A.10: Results of Experimental Studies for Simulated Data Sets with 200 Data Points and 2 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
2	4	2	4.11	100.00	4.57	TL	8.80	100.00	9.29	TL	-0.44	0	0	303.57	-0.44	0	0	183.41
2	4	3	0	0	0	5562.06	0	100.00	0	TL	0	0	0	280.96	0	0	0	214.33
2	5	2	-1.97	100.00	0.79	TL	2.88	100.00	5.77	TL	-1.54	3.85	1.23	TL	-2.74	0	0	4025.89
2	5	3	3.56	100.00	3.56	TL	8.21	56.60	8.21	TL	0	0	0	4931.80	0	0	0	2189.46
2	5	4	0	100.00	0	TL	13.46	100.00	13.46	TL	0	0	0	309.58	0	0	0	284.15
2	6	2	9.47	100.00	9.47	TL	13.55	100.00	13.55	TL	0	0	0	1693.36	0	0	0	419.38
2	6	3	8.67	100.00	8.67	TL	19.36	100.00	19.36	TL	21.62	19.24	21.62	TL	0	0	0	2766.46
2	6	4	3.29	100.00	3.29	TL	3.61	100.00	3.61	TL	3.19	4.04	3.19	TL	0	0	0	4281.79
2	8	2	8.36	100.00	9.47	TL	4.58	100.00	5.65	TL	0.38	9.39	1.41	TL	-1.02	8.13	0	TL
2	8	3	1.51	100.00	0	TL	7.50	100.00	5.89	TL	38.84	37.73	36.77	TL	9.55	18.86	7.91	TL
2	8	4	8.87	100.00	7.65	TL	13.71	100.00	12.43	TL	32.56	31.14	31.07	TL	1.14	3.85	0	TL
2	8	6	8.81	100.00	8.81	TL	3.61	100.00	3.61	TL	7.85	9.97	7.85	TL	0	0	0	6148.12
2	10	2	2.70	100.00	3.72	TL	42.15	100.00	43.55	TL	-0.98	13.31	0	TL	-0.98	11.15	0	TL
2	10	3	18.21	100.00	12.99	TL	4.63	100.00	0	TL	52.40	42.25	45.66	TL	56.00	40.19	49.10	TL
2	10	4	14.99	100.00	8.94	TL	27.53	100.00	20.82	TL	98.72	59.33	88.25	TL	5.56	18.90	0	TL
2	10	6	1.93	100.00	0	TL	29.57	100.00	27.12	TL	4.17	11.86	2.20	TL	15.44	18.82	13.25	TL
2	12	2	-0.25	100.00	0	TL	28.76	100.00	29.09	TL	14.55	21.65	14.84	TL	1.95	9.95	2.21	TL
2	12	3	16.47	100.00	0.02	TL	16.45	100.00	0	TL	18.84	27.61	2.05	TL	100.75	56.09	72.39	TL
2	12	4	34.33	100.00	18.41	TL	28.53	100.00	13.30	TL	13.45	25.94	0	TL	25.69	29.50	10.79	TL
2	12	6	23.99	100.00	17.28	TL	18.77	100.00	12.33	TL	5.73	15.14	0	TL	75.53	47.71	66.02	TL
Average			8.35	95.00	5.88	5562.06	14.78	97.83	12.35	-	15.47	16.62	12.81	1503.85	14.32	13.16	11.08	2279.22

Table A.11: Results of Experimental Studies for Simulated Data Sets with 200 Data Points and 3 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
3	4	2	14.54	100.00	14.54	TL	5.95	100.00	5.95	TL	15.29	14.75	15.29	TL	0	0	0	3455.53
3	4	3	0	65.79	0	TL	10.65	100.00	10.65	TL	0	0	0	1978.92	0	0	0	894.60
3	5	2	39.05	100.00	40.73	TL	15.96	100.00	17.36	TL	-1.19	0	0	5820.60	-1.19	0	0	2907.22
3	5	3	16.55	100.00	16.55	TL	46.30	100.00	46.30	TL	0	0	0	768.22	0	0	0	762.74
3	5	4	15.20	100.00	15.20	TL	10.42	48.65	10.42	TL	0	0	0	4972.42	0	0	0	2072.19
3	6	2	15.50	100.00	4.33	TL	66.37	100.00	50.28	TL	22.79	25.32	10.91	TL	10.71	14.05	0	TL
3	6	3	64.71	100.00	62.83	TL	22.71	100.00	21.31	TL	1.15	7.69	0	TL	39.52	29.94	37.93	TL
3	6	4	63.12	100.00	63.12	TL	20.16	100.00	20.16	TL	0	0	0	4404.55	0	0	0	2019.48
3	8	2	27.85	100.00	14.23	TL	52.08	100.00	35.88	TL	28.35	49.43	14.67	TL	11.93	30.24	0	TL
3	8	3	33.56	100.00	0	TL	34.07	100.00	0.38	TL	76.36	63.89	32.05	TL	46.12	49.67	9.41	TL
3	8	4	42.34	100.00	24.00	TL	38.70	100.00	20.83	TL	14.79	35.61	0	TL	19.25	31.19	3.89	TL
3	8	6	11.90	100.00	7.94	TL	3.67	100.00	0	TL	40.67	40.44	35.69	TL	16.41	23.99	12.29	TL
3	10	2	26.69	100.00	19.12	TL	17.88	100.00	10.83	TL	11.17	46.76	4.52	TL	6.36	40.48	0	TL
3	10	3	51.77	100.00	0	TL	52.15	100.00	0.25	TL	130.20	71.38	51.68	TL	69.72	57.67	11.83	TL
3	10	4	15.33	100.00	0	TL	39.28	100.00	20.77	TL	88.68	66.02	63.59	TL	73.12	59.65	50.10	TL
3	10	6	30.36	100.00	13.49	TL	14.87	100.00	0	TL	56.75	50.51	36.46	TL	265.66	77.54	218.33	TL
3	12	2	7.46	100.00	6.40	TL	7.09	100.00	6.03	TL	3.24	49.14	2.22	TL	1.00	45.67	0	TL
3	12	3	48.54	100.00	0	TL	61.63	100.00	8.81	TL	115.73	69.57	45.24	TL	136.08	71.67	58.93	TL
3	12	4	44.25	100.00	4.14	TL	38.51	100.00	0	TL	173.12	78.54	97.19	TL	54.92	58.07	11.85	TL
3	12	6	90.07	100.00	38.74	TL	37.00	100.00	0	TL	182.87	77.78	106.48	TL	63.16	58.39	19.10	TL
Average			32.94	98.29	17.27	-	29.77	97.43	14.31	-	48.00	37.34	25.80	3588.94	40.64	32.41	21.68	2018.63

Table A.12: Results of Experimental Studies for Simulated Data Sets with 200 Data Points and 4 Clusters

p	m	q	NM				LM_1				LM_2				LM_3			
			$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_O$	$\%Gap_B$	$CPU\ s.$
4	4	2	29.44	100.00	29.44	TL	22.95	100.00	22.95	TL	0	3.39	0	TL	0	0.61	0	TL
4	4	3	16.59	100.00	16.59	TL	71.69	100.00	71.69	TL	0	0	0	1701.07	0	0	0	1007.09
4	5	2	6.27	100.00	4.26	TL	10.97	100.00	8.88	TL	9.40	19.94	7.33	TL	1.92	9.46	0	TL
4	5	3	32.64	100.00	32.64	TL	42.73	100.00	42.73	TL	0	0	0	3503.48	0	0	0	1970.24
4	5	4	9.93	78.46	9.93	TL	21.15	100.00	21.15	TL	0	0	0	1420.40	0	0	0	815.56
4	6	2	53.55	100.00	27.47	TL	22.15	100.00	1.40	TL	36.97	35.50	13.70	TL	20.47	23.03	0	TL
4	6	3	40.60	100.00	0	TL	53.62	100.00	9.26	TL	76.13	52.37	25.27	TL	360.76	79.60	227.71	TL
4	6	4	27.85	100.00	27.85	TL	18.65	100.00	18.65	TL	3.29	7.64	3.29	TL	0	0	0	4467.98
4	8	2	29.20	100.00	41.65	TL	2.77	100.00	12.67	TL	-8.78	39.53	0	TL	-0.52	35.42	9.06	TL
4	8	3	74.14	100.00	31.08	TL	32.85	100.00	0	TL	258.27	84.58	169.68	TL	67.77	56.41	26.29	TL
4	8	4	40.78	100.00	7.71	TL	30.70	100.00	0	TL	67.53	58.89	28.18	TL	60.04	51.33	22.45	TL
4	8	6	40.57	100.00	20.11	TL	17.04	100.00	0	TL	19.47	30.66	2.08	TL	30.16	29.81	11.21	TL
4	10	2	-6.05	100.00	10.43	TL	3.61	100.00	21.79	TL	-14.92	45.69	0	TL	2.11	52.41	20.02	TL
4	10	3	31.84	100.00	0.58	TL	61.65	100.00	23.32	TL	69.20	65.67	29.09	TL	31.08	52.62	0	TL
4	10	4	49.78	100.00	0	TL	87.39	100.00	25.12	TL	119.16	72.74	46.32	TL	111.18	64.98	41.00	TL
4	10	6	41.49	100.00	6.23	TL	66.40	100.00	24.93	TL	112.27	67.19	59.37	TL	33.19	41.36	0	TL
4	12	2	0.83	100.00	13.53	TL	-10.76	100.00	0.48	TL	-11.19	55.81	0	TL	19.68	65.03	34.75	TL
4	12	3	44.71	100.00	0	TL	55.27	100.00	7.29	TL	187.86	81.18	98.92	TL	249.14	83.94	141.26	TL
4	12	4	48.32	100.00	0	TL	72.08	100.00	16.02	TL	162.20	82.14	76.78	TL	208.95	82.95	108.30	TL
4	12	6	55.48	100.00	17.36	TL	32.49	100.00	0	TL	199.82	82.61	126.30	TL	261.31	83.98	172.72	TL
Average			33.40	98.92	14.84	-	35.77	100.00	16.42	-	64.33	44.28	34.32	2208.32	72.86	40.65	40.74	2065.22

APPENDIX B

COMPARISON OF LM_3, H_1, AND H_2 FOR SIMULATED DATA SETS

In this appendix, we give experimental results of the mathematical model LM_3 and heuristic algorithms H_1 and H_2 for each problem instances.

Performance measures used in all tables are explained below.

$\%Gap_M$: Percent gap from made objective
$\%Gap_B$: Percent gap from best available solution
$Hits$: Number of hits to the solution
$CPUs.$: Computational time

Table B.1: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 50 Data Points and 2 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
2	4	2	-2.68	0	8.10	4.26	7.13	19	44.36	48.33	3	8.93	-2.679	0	41	49.89	54.02	1	0.01
2	4	3	0	0	3.08	0	0	50	0	0	50	6.64	0	0	50	0	0	50	0.02
2	5	2	0	0	2.20	0	0	46	118.64	118.64	3	9.53	0	0	50	0	0	50	0.02
2	5	3	0	0	2.23	0	0	50	0	0	50	7.16	0	0	50	0	0	50	0.02
2	5	4	0	0	4.69	0	0	50	0	0	50	6.90	0	0	50	0	0	50	0.02
2	6	2	-1.95	0	8.03	-1.56	0.39	3	66.83	70.14	6	10.00	-1.95	0	23	67.68	71.01	1	0.02
2	6	3	0	0	3.96	0	0	50	0	0	50	9.65	0	0	48	131.05	131.05	2	0.02
2	6	4	0	0	8.86	0	0	49	119.38	119.38	1	8.36	0	0	42	1.38	1.38	8	0.02
2	8	2	-3.21	0	30.56	0.20	3.52	18	40.28	44.93	1	9.04	2.17	5.56	50	2.17	5.56	50	0.03
2	8	3	0	0	12.96	0	0	46	109.92	109.92	1	12.41	0	0	50	0	0	50	0.03
2	8	4	0	0	24.68	0	0	50	0	0	50	10.68	0	0	50	0	0	50	0.03
2	8	6	0	0	23.32	0	0	48	144.10	144.10	2	9.65	0	0	50	0	0	50	0.02
2	10	2	0	0	8.81	0	0	47	119.94	119.94	1	20.30	0	0	39	152.40	152.40	2	0.03
2	10	3	0	0	124.17	0	0	33	87.81	87.81	5	13.45	0	0	29	71.04	71.04	8	0.03
2	10	4	0	0	63.35	0	0	18	87.00	87.00	1	12.63	0	0	48	86.48	86.48	2	0.03
2	10	6	0	0	74.50	0	0	50	0	0	50	15.20	0.73	0.73	50	0.73	0.73	50	0.03
2	12	2	-0.19	0	10.41	-0.19	0	26	110.98	111.38	1	13.35	8.48	8.68	11	84.45	84.79	7	0.03
2	12	3	0	0	28.88	0	0	32	132.99	132.99	1	15.12	0	0	48	128.02	128.02	1	0.04
2	12	4	0	0	123.84	0	0	25	98.87	98.87	1	12.45	0	0	49	101.22	101.22	1	0.04
2	12	6	0	0	54.56	0	0	50	0	0	50	13.79	0	0	50	0	0	50	0.03
Average			-0.40	0	31.06	0.14	0.55	38.00	64.06	64.67	18.85	11.26	0.34	0.75	43.90	43.83	44.38	26.65	0.02

Table B.2: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 50 Data Points and 3 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
3	4	2	-3.52	0	12.49	4.26	8.06	29	17.53	21.82	2	11.95	-3.52	0	19	59.54	65.36	7	0.02
3	4	3	0	0	2.36	0	0	50	0	0	50	11.44	0	0	35	172.67	172.67	1	0.02
3	5	2	-0.47	0	9.55	3.43	3.92	5	58.91	59.66	1	12.72	-0.47	0	26	68.11	68.91	3	0.02
3	5	3	0	0	3.06	0	0	47	174.15	174.15	1	12.84	0	0	36	176.91	176.91	2	0.02
3	5	4	0	0	4.28	0	0	50	0	0	50	13.30	0	0	42	133.53	133.53	1	0.02
3	6	2	-0.64	0	22.26	25.54	26.34	6	54.08	55.07	1	14.24	-0.38	0.26	25	74.01	75.12	1	0.02
3	6	3	-2.47	0	10.89	-2.47	0	3	90.40	95.23	2	13.16	-2.47	0	35	107.08	112.32	2	0.02
3	6	4	0	0	6.05	0	0	41	212.88	212.88	1	14.01	0	0	32	214.23	214.23	1	0.02
3	8	2	-4.52	0	500.96	-1.61	3.04	18	18.44	24.04	1	46.65	-4.52	0	1	46.41	53.33	2	0.02
3	8	3	0	0	69.89	42.75	42.75	2	71.46	71.46	1	15.01	0	0	20	66.86	66.86	1	0.03
3	8	4	0	0	47.03	0	0	25	146.67	146.67	1	47.26	0.06	0.06	9	70.75	70.75	1	0.03
3	8	6	-0.44	0	13.66	-0.44	0	50	-0.44	0	50	17.25	0.33	0.77	26	205.35	206.70	9	0.03
3	10	2	-8.00	0.81	TL	-8.24	0.55	3	23.98	35.85	0.55	20.21	-8.24	1	1	25.32	37.32	1	0.03
3	10	3	-1.08	0	TL	15.49	16.75	1	44.99	46.58	1	15.89	1.17	2.28	27	49.05	50.68	2	0.03
3	10	4	0	0	74.67	3.76	3.76	2	89.79	89.79	1	22.39	0	0	20	78.70	78.70	1	0.03
3	10	6	-1.12	0	34.22	-1.12	0	49	163.75	166.74	1	20.22	0.35	1.49	31	168.57	171.61	6	0.03
3	12	2	-7.64	1.14	TL	-8.68	0	7	26.81	38.88	1	25.50	-5.44	3.55	1	59.54	74.71	2	0.03
3	12	3	0	0	551.89	0	0	1	108.77	108.77	1	26.30	1.77	1.77	3	79.53	79.53	1	0.03
3	12	4	-2.61	0	7200.28	26.16	29.54	1	60.51	64.82	1	21.62	-2.21	0.42	30	47.21	51.16	2	0.03
3	12	6	-1.78	0	263.35	-1.78	0	38	120.72	124.71	1	31.01	-1.78	0	33	117.20	121.13	1	0.04
Average			-1.71	0.10	519.23	4.85	6.74	21.40	74.17	76.86	8.45	20.65	-1.27	0.56	22.60	101.03	104.08	2.35	0.02

Table B.3: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 50 Data Points and 4 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
4	4	2	0	0	6.28	0	0	3	168.51	168.51	1	18.51	0	0	27	218.10	218.10	1	0.02
4	4	3	0	0	3.09	0	0	44	111.79	111.79	1	16.01	0	0	31	110.92	110.92	1	0.02
4	5	2	0	0	8.17	0	0	1	91.94	91.94	1	18.12	0	0	35	72.75	72.75	4	0.02
4	5	3	0	0	5.72	0	0	40	165.10	165.10	1	19.66	2.37	2.37	27	160.16	160.16	1	0.02
4	5	4	0	0	4.18	0	0	43	82.87	82.87	1	19.89	0	0	36	105.48	105.48	1	0.02
4	6	2	-5.23	0	2362.39	-4.76	0.49	1	29.50	36.64	1	19.01	-5.23	0	1	23.70	30.52	1	0.02
4	6	3	-4.71	0	34.03	-4.71	0	1	68.98	77.34	1	21.78	-4.71	0	1	118.91	129.74	1	0.02
4	6	4	-0.85	0	7.12	-0.85	0	46	95.44	97.12	1	22.91	-0.05	0.81	8	95.07	96.74	12	0.02
4	8	2	-26.84	0	249.95	-24.65	2.98	15	-3.75	31.55	1	58.12	-26.52	0.43	1	17.04	59.98	1	0.02
4	8	3	-1.51	0	273.52	11.37	13.08	1	96.81	99.82	1	27.11	-1.51	0	9	106.71	109.86	1	0.03
4	8	4	-1.92	0	23.48	-1.92	0	16	135.09	139.70	1	72.58	-1.92	0	21	170.32	175.62	1	0.02
4	8	6	-1.69	0	10.69	-1.69	0	46	101.96	105.44	1	28.05	-1.69	0	29	239.76	245.62	1	0.03
4	10	2	-20.35	0	6572.27	-20.35	0	21	1.27	27.15	1	28.40	-18.80	1.95	1	15.69	45.26	1	0.03
4	10	3	-0.40	1.87	TL	5.74	8.16	1	60.76	64.43	1	26.01	-2.23	0	14	45.62	48.94	1	0.03
4	10	4	-0.43	0	76.85	-0.43	0	9	155.57	156.67	1	32.06	-0.43	0	11	130.08	131.06	1	0.03
4	10	6	-1.92	0	35.98	-1.92	0	38	118.80	123.07	1	32.47	-0.45	1.50	21	119.34	123.62	3	0.03
4	12	2	-24.67	2.44	TL	-25.29	1.60	1	0.03	36.03	1	35.85	-23.42	4.15	1	21.90	65.78	1	0.03
4	12	3	-0.12	0	TL	33.02	33.18	1	103.45	103.70	1	34.43	-0.12	0	5	126.56	126.83	1	0.04
4	12	4	-7.07	0	3131.26	-7.07	0	2	63.13	75.54	1	46.00	-1.96	5.50	10	86.87	101.08	1	0.03
4	12	6	-2.85	0	138.32	-2.85	0	42	89.07	94.62	1	49.20	0.65	3.61	1	151.08	158.45	1	0.03
Average			-5.03	0.22	761.37	-2.32	2.97	18.60	86.82	94.45	1.00	31.31	-4.30	1.02	14.50	106.80	115.83	1.80	0.02

Table B.4: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 80 Data Points and 2 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
2	4	2	0	0	7.44	0	0	50	0	0	50	11.55	0	0	49	5.66	5.66	1	0.03
2	4	3	0	0	8.41	0	0	50	0	0	50	9.35	0	0	50	0	0	50	0.03
2	5	2	0	0	15.27	0	0	50	0	0	50	13.32	0	0	50	0	0	50	0.03
2	5	3	0	0	64.90	0	0	47	95.01	95.01	3	12.91	0	0	50	0	0	50	0.04
2	5	4	0	0	11.99	0	0	50	0	0	50	10.01	0	0	50	0	0	50	0.04
2	6	2	-3.11	0	35.16	-3.11	0	48	57.31	62.37	1	13.51	-3.11	0	43	85.88	91.85	2	0.04
2	6	3	0	0	20.53	0	0	50	0	0	50	22.78	0	0	50	0	0	50	0.04
2	6	4	0	0	31.95	0	0	50	0	0	50	12.12	0	0	50	0	0	50	0.04
2	8	2	-2.05	0	314.79	-2.05	0	4	61.33	64.70	1	13.14	-2.05	0	28	45.69	48.73	3	0.05
2	8	3	0	0	99.50	0	0	50	0	0	50	23.61	0	0	48	78.31	78.31	1	0.05
2	8	4	-0.06	0	140.65	-0.06	0	49	119.47	119.60	1	19.33	-0.06	0	50	-0.06	0	50	0.06
2	8	6	0	0	169.49	0	0	48	135.78	135.78	1	17.51	0	0	50	0	0	50	0.05
2	10	2	0	0	31.91	0	0	6	78.92	78.92	1	19.06	0	0	34	81.93	81.93	2	0.05
2	10	3	0	0	366.91	0	0	36	98.22	98.22	1	24.91	0	0	13	92.03	92.03	1	0.05
2	10	4	0	0	560.25	0	0	36	96.40	96.40	3	22.27	0	0	50	0	0	50	0.06
2	10	6	0	0	372.94	0	0	47	92.67	92.67	3	21.52	1.24	1.24	50	1.24	1.24	50	0.06
2	12	2	-5.94	0	775.30	-5.94	0	45	47.34	56.65	1	21.91	-5.94	0	1	63.25	73.57	1	0.06
2	12	3	0	0	333.92	0	0	39	71.87	71.87	11	30.91	0	0	45	89.59	89.59	1	0.06
2	12	4	0	0	1166.27	0	0	49	61.36	61.36	1	30.81	0	0	46	28.58	28.58	1	0.08
2	12	6	0	0	604.21	0	0	23	89.13	89.13	1	24.84	0	0	48	3.15	3.15	2	0.08
Average			-0.56	0	256.59	-0.56	0	41.35	55.24	56.13	18.95	18.77	-0.50	0.06	42.75	28.76	29.73	25.75	0.05

Table B.5: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 80 Data Points and 3 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
3	4	2	-2.25	0	143.80	15.54	18.20	18	28.47	31.43	1	19.08	-2.25	0	17	68.67	72.54	10	0.03
3	4	3	0	0	16.68	0	0	49	186.22	186.22	1	16.55	0	0	37	184.80	184.80	1	0.03
3	5	2	-2.60	0	61.17	6.77	9.63	3	57.64	61.85	1	23.50	6.77	9.63	13	55.65	59.81	7	0.04
3	5	3	0	0	12.47	0	0	50	0	0	50	22.05	0	0	36	234.09	234.09	1	0.04
3	5	4	0	0	20.33	0	0	49	180.83	180.83	1	20.06	1.43	1.43	33	178.56	178.56	7	0.03
3	6	2	-1.42	0	162.80	29.48	31.35	2	56.94	59.20	7	22.22	-1.42	0	14	71.61	74.09	1	0.04
3	6	3	-2.13	0	74.50	-2.13	0	15	88.37	92.48	1	25.54	-2.13	0	31	100.73	105.11	1	0.05
3	6	4	0	0	37.59	0	0	47	123.52	123.52	1	24.34	0	0	31	185.61	185.61	5	0.04
3	8	2	-12.51	0	2128.34	-12.51	0	25	-3.17	10.67	2	30.18	-12.51	0	4	7.93	23.36	2	0.05
3	8	3	0	0	2504.22	17.49	17.49	2	30.15	30.15	1	25.68	0	0	8	38.02	38.02	1	0.05
3	8	4	0	0	243.02	0	0	30	124.65	124.65	1	38.42	6.20	6.20	42	70.62	70.62	2	0.06
3	8	6	0	0	91.94	0	0	50	0	0	50	32.73	0	0	31	193.50	193.50	2	0.05
3	10	2	-7.92	0.44	TL	-6.76	1.71	1	15.99	26.53	1	44.68	-5.33	3.27	1	44.52	57.65	1	0.05
3	10	3	1.84	2.87	TL	24.67	25.94	2	51.24	52.78	1	36.89	-0.92	0.08	2	54.51	56.08	1	0.06
3	10	4	0	0	537.13	0	0	8	102.24	102.24	1	43.95	0	0	1	91.05	91.05	1	0.05
3	10	6	0	0	221.55	0	0	46	176.41	176.41	1	38.18	6.25	6.25	38	109.54	109.54	1	0.06
3	12	2	-17.79	9.65	TL	-25.02	0	1	-4.25	27.71	1	51.06	-18.67	8.47	1	3.13	37.56	1	0.06
3	12	3	-2.07	0.48	TL	0.63	3.25	1	73.97	78.50	1	53.22	-2.54	0	17	64.74	69.04	2	0.06
3	12	4	0.53	0.53	TL	38.08	38.08	1	71.60	71.60	1	36.90	0	0	9	52.16	52.16	1	0.07
3	12	6	0	0	533.00	0	0	44	146.56	146.56	1	64.02	0	0	10	139.91	139.91	1	0.09
Average			-2.32	0.70	452.57	4.31	7.28	22.20	75.37	79.17	6.25	33.46	-1.26	1.77	18.80	97.47	101.66	2.45	0.05

Table B.6: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 80 Data Points and 4 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
4	4	2	0	0	41.47	17.15	17.15	9	36.87	36.87	1	31.43	0	0	24	115.26	115.26	2	0.03
4	4	3	0	0	20.95	0	0	47	120.61	120.61	1	30.33	0	0	32	107.41	107.41	1	0.03
4	5	2	-4.24	0	121.41	-4.24	0	1	67.49	74.91	1	35.31	-4.24	0	3	36.95	43.02	1	0.03
4	5	3	0	0	17.69	0	0	44	198.93	198.93	1	31.52	0	0	3	224.41	224.41	1	0.04
4	5	4	0	0	12.90	0	0	8	140.62	140.62	2	30.83	0	0	35	260.20	260.20	1	0.04
4	6	2	-1.95	0	380.52	35.99	38.69	1	67.10	70.41	1	35.23	1.50	3.52	6	63.93	67.18	1	0.04
4	6	3	0	0	69.53	0	0	20	99.07	99.07	1	44.07	0	0	28	102.41	102.41	3	0.04
4	6	4	0	0	32.53	0	0	29	181.37	181.37	1	37.67	0	0	8	183.13	183.13	1	0.04
4	8	2	-4.29	6.21	TL	-9.89	0	2	14.43	26.99	1	46.33	-9.89	0	1	26.24	40.09	1	0.06
4	8	3	0	0	3538.41	22.56	22.56	1	113.58	113.58	1	50.77	0	0	13	88.64	88.64	1	0.05
4	8	4	-0.16	0	231.76	-0.16	0	35	144.26	144.66	1	63.53	-0.16	0	28	184.76	185.23	1	0.06
4	8	6	0	0	73.99	0	0	17	169.61	169.61	1	46.23	0	0	35	251.62	251.62	1	0.06
4	10	2	-10.13	1.58	TL	-11.53	0	1	3.87	17.41	1	50.92	-2.89	9.77	2	34.20	51.69	1	0.06
4	10	3	0.23	0.92	TL	24.30	25.16	1	70.59	71.76	1	54.79	-0.68	0	2	59.34	60.43	1	0.06
4	10	4	0	0	767.25	0	0	5	95.64	95.64	1	73.03	0	0	6	127.20	127.20	1	0.06
4	10	6	0	0	162.85	0	0	24	145.88	145.88	1	73.92	0.67	0.67	28	227.07	227.07	1	0.07
4	12	2	-22.90	2.90	TL	-20.74	5.78	2	-3.12	29.31	1	82.25	-20.40	6.24	1	11.03	48.19	1	0.07
4	12	3	25.74	25.74	TL	14.33	14.33	1	94.13	94.13	1	83.18	0	0	9	62.97	62.97	1	0.07
4	12	4	6.86	6.86	TL	0	0	1	82.30	82.30	1	97.21	0	0	3	90.52	90.52	1	0.08
4	12	6	-3.50	0	557.76	-3.50	0	45	89.44	96.31	1	103.27	-3.19	0.32	9	133.15	141.60	1	0.09
Average			-0.72	2.21	430.64	3.21	6.18	14.70	96.64	100.52	1.05	55.09	-1.96	1.03	13.80	119.52	123.91	1.15	0.05

Table B.7: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 100 Data Points and 2 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
2	4	2	-0.40	0	23.84	-0.40	0	50	-0.40	0	50	14.82	-0.40	0	50	-0.40	0	50	0.05
2	4	3	0	0	23.20	0	0	50	0	0	50	12.01	0	0	50	0	0	50	0.05
2	5	2	0	0	30.20	0	0	35	134.95	134.95	1	14.26	0	0	2	4.06	4.06	48	0.05
2	5	3	0	0	67.78	0	0	50	0	0	50	14.82	0	0	50	0	0	50	0.05
2	5	4	0	0	32.16	0	0	50	0	0	50	12.78	0	0	50	0	0	50	0.06
2	6	2	0	0	82.44	0	0	9	105.87	105.87	1	19.46	0.01	0.01	39	74.61	74.61	2	0.08
2	6	3	0	0	114.26	0	0	50	0	0	50	19.62	0	0	50	0	0	50	0.06
2	6	4	0	0	141.91	0	0	50	0	0	50	16.18	0	0	50	0	0	50	0.07
2	8	2	-1.03	0	481.76	-0.94	0.09	32	64.46	66.17	18	21.96	0	0	4	15.67	16.87	10	0.07
2	8	3	0	0	282.91	0	0	45	133.07	133.07	1	24.12	0	0	44	81.01	81.01	1	0.07
2	8	4	0	0	130.80	0	0	50	0	0	50	19.44	0	0	50	0	0	50	0.07
2	8	6	0	0	399.45	0	0	47	175.06	175.06	3	21.64	0	0	50	0	0	50	0.07
2	10	2	-3.84	0	479.91	-3.84	0	41	60.76	67.19	1	35.31	-3.84	0	30	60.64	67.07	3	0.07
2	10	3	0	0	1037.66	0	0	21	72.71	72.71	2	31.69	0	0	50	0	0	50	0.09
2	10	4	0	0	1250.38	0	0	24	88.81	88.81	1	43.13	0	0	44	103.03	103.03	2	0.09
2	10	6	0	0	972.61	0	0	47	158.37	158.37	1	31.19	0	0	50	0	0	50	0.09
2	12	2	0	0	517.76	0	0	1	102.43	102.43	3	39.20	0	0	7	20.54	20.54	12	0.09
2	12	3	0	0	4428.47	0	0	32	70.76	70.76	1	41.62	0	0	41	62.38	62.38	4	0.11
2	12	4	0	0	2698.05	0	0	33	80.26	80.26	1	45.57	0	0	2	84.73	84.73	1	0.14
2	12	6	0	0	1714.60	0	0	35	94.65	94.65	15	40.94	0	0	42	11.20	11.20	1	0.12
Average			-0.26	0	745.51	-0.26	0	37.60	67.09	67.51	19.95	25.99	-0.21	0	37.75	25.87	26.27	29.20	0.08

Table B.8: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 100 Data Points and 3 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
3	4	2	0	0	116.99	28.68	28.68	6	39.58	39.58	1	27.84	0	0	13	80.90	80.90	3	0.06
3	4	3	0	0	52.74	0	0	50	0	0	50	22.46	0	0	38	141.89	141.89	1	0.06
3	5	2	-0.52	0	238.42	25.91	26.57	1	62.36	63.21	5	28.98	-0.52	0	38	57.54	58.36	2	0.08
3	5	3	0	0	33.93	0	0	50	0	0	50	24.33	0	0	37	100.51	100.51	1	0.07
3	5	4	0	0	85.04	0	0	48	125.90	125.90	1	27.69	0	0	38	123.47	123.47	10	0.07
3	6	2	-3.87	0	1113.52	23.02	27.98	1	44.41	50.23	2	31.40	0.85	4.92	9	93.32	101.12	1	0.07
3	6	3	0	0	194.05	0	0	6	121.87	121.87	1	32.54	0	0	38	119.34	119.34	3	0.07
3	6	4	0	0	75.25	0	0	50	0	0	50	32.66	0	0	37	130.67	130.67	4	0.08
3	8	2	0.94	0	TL	0.94	0	8	16.86	15.77	1	42.48	0.94	0	3	47.12	45.75	1	0.09
3	8	3	1.78	1.78	TL	24.44	24.44	3	34.61	34.61	1	35.75	0	0	30	31.92	31.92	1	0.09
3	8	4	0	0	935.35	0	0	30	121.80	121.80	1	48.80	0	0	36	105.60	105.60	3	0.09
3	8	6	-0.87	0	317.36	-0.87	0	44	110.36	112.21	5	41.94	-0.87	0	37	111.86	113.73	3	0.10
3	10	2	-6.57	0.60	TL	-6.01	1.21	1	19.12	28.27	1	57.64	2.43	10.29	1	31.33	41.42	1	0.11
3	10	3	-0.43	0.23	TL	26.35	27.19	5	66.03	67.13	1	61.21	-0.66	0	11	52.95	53.96	1	0.11
3	10	4	0	0	1576.53	0	0	1	107.10	107.10	1	56.14	0	0	9	90.21	90.21	1	0.11
3	10	6	0	0	825.50	0	0	49	189.23	189.23	1	61.99	2.05	2.05	31	188.91	188.91	3	0.11
3	12	2	-15.33	0	TL	-13.67	1.96	1	1.28	19.61	1	65.27	-13.67	1.96	1	9.33	29.12	1	0.12
3	12	3	3.78	3.78	TL	26.08	26.08	1	88.44	88.44	1	75.96	0	0	9	56.91	56.91	1	0.13
3	12	4	5.76	4.73	TL	15.58	14.46	1	76.32	74.61	1	68.83	0.98	0	3	38.18	36.84	1	0.12
3	12	6	0	0	2241.76	0	0	33	138.18	138.18	1	75.95	0	0	1	113.69	113.69	1	0.16
Average			-0.77	0.56	600.49	7.52	8.93	19.45	68.17	69.89	8.80	45.99	-0.42	0.96	21.00	86.28	88.22	2.15	0.09

Table B.9: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 100 Data Points and 4 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
4	4	2	0	0	81.16	24.06	24.06	1	34.54	34.54	1	34.65	0	0	21	99.99	99.99	6	0.04
4	4	3	0	0	62.65	0	0	42	124.09	124.09	1	38.22	0.58	0.58	28	123.55	123.55	1	0.05
4	5	2	-0.26	0	130.15	-0.26	0	9	95.29	95.80	1	52.54	-0.26	0	4	67.17	67.60	1	0.05
4	5	3	0	0	116.12	0	0	40	183.02	183.02	1	46.03	0	0	21	100.99	100.99	1	0.06
4	5	4	0	0	41.94	0	0	21	146.38	146.38	1	41.87	0	0	35	260.19	260.19	1	0.06
4	6	2	-2.56	0	541.84	15.09	18.11	1	56.79	60.91	1	53.13	-2.56	0	13	73.82	78.39	1	0.06
4	6	3	-1.23	0	176.16	-1.23	0	31	139.06	142.04	1	63.04	1.86	3.13	27	152.62	155.77	1	0.06
4	6	4	0	0	124.09	0	0	38	166.21	166.21	2	52.47	0	0	30	228.16	228.16	1	0.06
4	8	2	-20.47	1.79	TL	-21.87	0	1	-12.30	12.26	1	69.59	-19.99	2.41	2	26.56	61.99	2	0.08
4	8	3	0	0	2803.20	22.37	22.37	1	101.44	101.44	1	74.75	0	0	15	105.21	105.21	1	0.08
4	8	4	0	0	513.96	0	0	37	142.84	142.84	1	84.57	0	0	27	206.37	206.37	1	0.07
4	8	6	0	0	377.77	0	0	38	93.48	93.48	1	90.41	0	0	31	242.60	242.60	1	0.09
4	10	2	-21.46	0	TL	-20.37	1.39	1	-9.25	15.55	1	86.94	-20.04	1.82	1	12.25	42.93	1	0.09
4	10	3	2.19	1.64	TL	19.33	18.69	1	73.99	73.05	1	83.29	0.54	0	3	48.66	47.86	2	0.10
4	10	4	-1.52	0	1515.99	-1.52	0	8	103.00	106.12	1	102.64	-1.52	0	7	115.44	118.76	1	0.09
4	10	6	0	0	670.75	0	0	37	145.59	145.59	1	111.79	0	0	29	238.57	238.57	1	0.09
4	12	2	-22.11	2.59	TL	-24.08	0	1	-7.87	21.35	1	121.90	-23.42	0.87	1	16.97	54.07	1	0.10
4	12	3	17.74	17.74	TL	15.85	15.85	1	91.44	91.44	1	121.51	0	0	2	58.98	58.98	1	0.11
4	12	4	0.37	0	TL	22.50	22.06	1	74.27	73.64	1	113.56	0.60	0.23	20	78.28	77.63	1	0.11
4	12	6	-1.69	0	1733.80	-1.69	0	42	107.68	111.25	1	150.62	-1.46	0.23	1	126.80	130.70	1	0.11
Average			-2.55	1.19	634.97	2.41	6.13	17.60	92.48	97.05	1.05	79.68	-3.28	0.46	15.90	119.16	125.02	1.35	0.08

Table B.10: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 200 Data Points and 2 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			*CPU s.*	*best*			*worst*			*CPU s.*
			$\%Gap_M$	$\%Gap_B$	*CPU s.*	$\%Gap_M$	$\%Gap_B$	*Hits*	$\%Gap_M$	$\%Gap_B$	*Hits*		$\%Gap_M$	$\%Gap_B$	*Hits*	$\%Gap_M$	$\%Gap_B$	*Hits*	
2	4	2	-0.44	0	183.41	-0.44	0	50	-0.44	0	50	27.25	-0.44	0	50	-0.44	0	50	0.20
2	4	3	0	0	214.33	0	0	50	0	0	50	20.81	0	0	50	0	0	50	0.17
2	5	2	-2.74	0	4025.89	-2.74	0	40	48.37	52.54	10	44.47	-2.74	0	8	48.37	52.54	1	0.19
2	5	3	0	0	2189.46	0	0	48	105.13	105.13	2	40.31	0	0	50	0	0	50	0.18
2	5	4	0	0	284.15	0	0	50	0	0	50	34.67	0	0	50	0	0	50	0.20
2	6	2	0	0	419.38	0	0	50	0	0	50	78.15	0	0	44	127.61	127.61	4	0.19
2	6	3	0	0	2766.46	0	0	48	91.01	91.01	2	52.10	0	0	49	38.06	38.06	1	0.23
2	6	4	0	0	4281.79	0	0	50	0	0	50	53.69	0	0	50	0	0	50	0.22
2	8	2	-1.02	0.42	TL	-1.43	0	25	64.48	66.87	3	68.56	-1.43	0	5	61.57	63.91	1	0.25
2	8	3	9.55	10.35	TL	-0.73	0	12	72.16	73.42	3	61.76	-0.73	0	40	72.16	73.42	1	0.26
2	8	4	1.14	1.14	TL	0	0	50	0	0	50	86.28	0	0	49	138.01	138.01	1	0.32
2	8	6	0	0	6148.12	0	0	50	0	0	50	58.52	0	0	50	0	0	50	0.28
2	10	2	-0.98	1.12	TL	-2.08	0	26	52.55	55.79	1	89.29	-2.08	0	3	52.34	55.57	1	0.28
2	10	3	56.00	56.62	TL	-0.40	0	47	80.09	80.80	2	116.50	-0.40	0	47	83.01	83.74	1	0.33
2	10	4	5.56	5.56	TL	0	0	50	0	0	50	114.20	0	0	47	95.78	95.78	2	0.32
2	10	6	15.44	15.44	TL	0	0	50	0	0	50	98.13	0	0	50	0	0	50	0.33
2	12	2	1.95	2.99	TL	-1.00	0	16	60.95	62.59	33	113.34	-1.00	0	2	67.23	68.93	2	0.36
2	12	3	100.75	101.13	TL	-0.19	0	9	83.04	83.39	17	107.36	-0.19	0	42	90.12	90.48	1	0.34
2	12	4	25.69	25.69	TL	0	0	32	107.63	107.63	6	121.93	0	0	45	116.30	116.30	1	0.40
2	12	6	75.53	75.53	TL	0	0	50	0	0	50	156.16	0	0	49	2.58	2.58	1	0.37
Average			14.32	14.80	2279.22	-0.45	0	40.15	38.25	38.96	28.95	77.17	-0.45	0	39.00	49.63	50.35	18.40	0.27

Table B.11: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 200 Data Points and 3 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
3	4	2	0	0	3455.53	26.47	26.47	37	36.38	36.38	2	83.83	0	0	34	95.89	95.89	2	0.16
3	4	3	0	0	894.60	0	0	50	0	0	50	49.30	0	0	38	140.27	140.27	1	0.17
3	5	2	-1.19	0	2907.22	7.13	8.43	1	61.53	63.48	1	84.89	-1.19	0	12	62.81	64.78	1	0.17
3	5	3	0	0	762.74	0	0	48	116.93	116.93	1	74.70	0	0	36	197.24	197.24	2	0.19
3	5	4	0	0	2072.19	0	0	49	110.72	110.72	1	72.62	0	0	36	173.75	173.75	3	0.18
3	6	2	10.71	12.08	TL	46.77	48.59	16	58.44	60.42	1	113.65	-1.23	0	19	89.83	92.20	2	0.20
3	6	3	39.52	39.52	TL	0	0	4	92.98	92.98	6	97.42	0	0	40	98.28	98.28	2	0.19
3	6	4	0	0	2019.48	0	0	47	202.32	202.32	1	90.63	0	0	39	119.64	119.64	11	0.22
3	8	2	11.93	7.88	TL	3.75	0	12	22.73	18.29	1	155.65	3.84	0.09	4	38.16	33.17	1	0.28
3	8	3	46.12	46.12	TL	41.14	41.14	8	48.05	48.05	2	132.02	0	0	32	48.90	48.90	3	0.26
3	8	4	19.25	19.25	TL	0	0	38	135.48	135.48	1	200.93	0	0	2	148.63	148.63	1	0.37
3	8	6	16.41	16.41	TL	0	0	50	0	0	50	159.90	1.99	1.99	32	196.09	196.08	1	0.34
3	10	2	6.36	10.52	TL	-3.77	0	1	15.79	20.33	1	291.55	-2.04	1.80	1	38.57	44.00	1	0.41
3	10	3	69.72	69.75	TL	26.87	26.89	3	63.18	63.21	2	240.40	-0.02	0	16	54.32	54.35	1	0.39
3	10	4	73.12	73.12	TL	0	0	6	93.38	93.38	1	272.38	0	0	33	64.01	64.01	2	0.41
3	10	6	265.66	254.16	TL	3.25	0	40	29.08	25.02	3	296.13	3.25	0	33	171.13	162.60	4	0.39
3	12	2	1.00	13.14	TL	-10.73	0	8	1.77	14.00	1	290.55	-8.91	2.04	1	25.21	40.26	1	0.35
3	12	3	136.08	133.93	TL	17.77	16.70	2	70.21	68.66	1	315.47	0.92	0	2	67.28	65.76	1	0.34
3	12	4	54.92	54.92	TL	31.69	31.69	1	82.11	82.11	1	309.10	0	0	23	53.87	53.87	1	0.40
3	12	6	63.16	63.16	TL	0	0	50	0	0	50	478.59	0	0	35	125.29	125.29	1	0.45
Average			40.64	40.70	2018.63	9.52	10.00	23.55	62.05	62.59	8.85	190.49	-0.17	0.30	23.40	100.46	100.95	2.10	0.29

Table B.12: Comparison of LM_3, H_1, and H_2 for Simulated Data Sets with 200 Data Points and 4 Clusters

p	m	q	**LM_3**			**H_1**							**H_2**						
						best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$CPU\ s.$	$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
4	4	2	0	0	TL	12.24	12.24	1	20.27	20.27	1	112.30	0	0	11	138.94	138.94	1	0.16
4	4	3	0	0	1007.09	0	0	47	124.25	124.25	1	118.84	0	0	28	239.95	239.95	1	0.17
4	5	2	1.92	1.36	TL	5.92	5.33	2	59.84	58.96	1	200.57	0.56	0	1	51.74	50.90	2	0.20
4	5	3	0	0	1970.24	0	0	35	189.74	189.74	1	144.61	0	0	29	199.07	199.07	1	0.21
4	5	4	0	0	815.56	0	0	21	145.41	145.41	1	116.23	0	0	20	265.75	265.75	1	0.20
4	6	2	20.47	22.37	TL	18.55	20.43	1	61.03	63.58	1	201.71	-1.56	0	1	144.29	148.16	1	0.21
4	6	3	360.76	361.06	TL	-0.07	0	2	118.64	118.78	1	227.26	-0.07	0	17	148.81	148.97	1	0.22
4	6	4	0	0	4467.98	0	0	28	161.40	161.40	1	173.28	0	0	18	160.88	160.88	1	0.23
4	8	2	-0.52	20.03	TL	-17.12	0	4	-6.16	13.22	1	254.66	-16.55	0.69	1	30.52	57.48	2	0.27
4	8	3	67.77	67.77	TL	24.29	24.29	1	111.35	111.35	1	305.40	0	0	14	106.62	106.62	1	0.27
4	8	4	60.04	60.04	TL	0	0	40	124.05	124.05	1	586.81	0	0	1	157.68	157.68	1	0.35
4	8	6	30.16	30.16	TL	0	0	42	95.86	95.87	1	298.88	0	0	26	237.73	237.74	1	0.39
4	10	2	2.11	28.95	TL	-20.82	0	2	-12.08	11.04	1	457.22	-17.34	4.40	1	10.39	39.42	1	0.42
4	10	3	31.08	25.10	TL	54.60	47.55	1	77.62	69.53	1	372.41	4.78	0	3	68.24	60.57	1	0.43
4	10	4	111.18	111.18	TL	0	0	10	116.05	116.05	1	535.91	0	0	19	142.68	142.68	1	0.45
4	10	6	33.19	33.19	TL	0	0	44	139.27	139.27	1	515.73	2.47	2.47	4	210.52	210.52	1	0.41
4	12	2	19.68	58.84	TL	-24.66	0	1	-13.91	14.26	1	610.63	-22.36	3.05	1	14.19	51.56	1	0.49
4	12	3	249.14	249.37	TL	42.80	42.89	1	83.06	83.18	1	669.99	-0.07	0	14	110.89	111.03	1	0.46
4	12	4	208.95	208.95	TL	5.97	5.97	1	74.95	74.95	1	805.45	0	0	10	87.46	87.46	1	0.49
4	12	6	261.31	263.70	TL	-0.66	0	41	98.34	99.65	1	757.57	-0.66	0	2	146.03	147.65	1	0.45
Average			72.86	77.10	2065.22	5.05	7.94	16.25	88.45	91.74	1.00	373.27	-2.54	0.53	11.05	133.62	138.15	1.10	0.32

Table B.13: Comparison of H_1 and H_2 for Simulated Data Sets with 500 Data Points and 2 Clusters

p	m	q	**H_1**							**H_2**						
			best			*worst*			*CPU s.*	*best*			*worst*			*CPU s.*
			$\%Gap_M$	$\%Gap_B$	*Hits*	$\%Gap_M$	$\%Gap_B$	*Hits*		$\%Gap_M$	$\%Gap_B$	*Hits*	$\%Gap_M$	$\%Gap_B$	*Hits*	
2	4	2	-0.08	0	50	-0.08	0	50	155.05	-0.08	0	49	71.93	72.06	1	1.41
2	4	3	0	0	50	0	0	50	109.04	0	0	50	0	0	50	1.33
2	5	2	-0.02	0	50	-0.02	0	50	244.49	-0.02	0	49	55.91	55.94	1	1.58
2	5	3	0	0	49	113.87	113.87	1	211.90	0	0	50	0	0	50	1.60
2	5	4	0	0	50	0	0	50	152.49	0	0	50	0	0	50	1.48
2	6	2	-0.32	0	43	80.51	81.08	1	373.24	-0.32	0	20	97.67	98.30	2	1.67
2	6	3	0	0	23	58.58	58.59	1	282.18	0	0	2	8.42	8.42	7	1.77
2	6	4	0	0	50	0	0	50	199.68	0	0	50	0	0	50	1.65
2	8	2	11.73	14.34	3	26.42	29.38	1	549.80	-2.29	0	12	35.81	38.99	1	2.38
2	8	3	0	0	35	93.93	93.93	1	546.94	0	0	47	108.53	108.53	2	2.31
2	8	4	0	0	49	109.98	109.98	1	478.54	0	0	50	0	0	50	2.34
2	8	6	0	0	50	0	0	50	355.29	0	0	50	0	0	50	2.25
2	10	2	18.77	20.34	1	19.62	21.19	19	708.13	-1.30	0	22	30.69	32.41	1	2.76
2	10	3	0	0	43	47.88	47.88	2	809.37	0	0	27	50.44	50.44	1	2.37
2	10	4	0	0	13	67.81	67.81	10	705.12	8.56	8.56	46	67.08	67.08	1	2.91
2	10	6	0	0	50	0	0	50	623.68	0	0	49	6.48	6.48	1	2.63
2	12	2	16.74	16.79	2	38.79	38.85	1	1092.70	-0.05	0	16	37.60	37.67	1	3.02
2	12	3	17.51	18.27	1	23.26	24.06	3	856.07	-0.64	0	21	37.88	38.77	1	2.90
2	12	4	0	0	28	56.10	56.10	3	839.08	0	0	49	42.43	42.43	1	3.27
2	12	6	0	0	49	104.35	104.35	1	952.49	0	0	50	0	0	50	3.07
Average			3.22	3.49	34.45	42.05	42.35	19.75	512.26	0.19	0.43	37.95	32.54	32.88	18.55	2.23

Table B.14: Comparison of H_1 and H_2 for Simulated Data Sets with 500 Data Points and 3 Clusters

p	m	q	$\mathbf{H_1}$							$\mathbf{H_2}$						
			best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
3	4	2	-0.19	0	43	105.98	106.39	1	455.33	-0.19	0	34	150.57	151.06	1	1.30
3	4	3	0	0	50	0	0	50	403.33	0	0	41	142.05	142.05	4	1.32
3	5	2	28.11	33.46	7	44.66	50.70	2	635.37	-4.01	0	17	44.07	50.09	3	1.61
3	5	3	0	0	50	0	0	50	505.72	0	0	37	265.56	265.56	1	1.62
3	5	4	0	0	50	0	0	50	499.16	0	0	40	151.06	151.06	1	1.47
3	6	2	33.47	34.65	7	48.79	50.10	6	689.60	-0.88	0	20	75.99	77.55	1	1.64
3	6	3	-0.02	0	50	-0.02	0	50	694.91	-0.02	0	39	115.19	115.22	1	1.60
3	6	4	0	0	48	100.06	100.06	2	934.02	0	0	40	100.92	100.92	9	1.60
3	8	2	-2.90	1.03	17	10.63	15.10	4	1538.36	-3.88	0	2	43.87	49.68	1	2.53
3	8	3	38.80	38.80	12	71.78	71.78	1	1587.45	0	0	27	78.76	78.76	1	2.43
3	8	4	0	0	47	144.09	144.09	1	1465.25	0	0	39	144.26	144.26	1	2.16
3	8	6	0	0	46	108.42	108.42	1	1102.96	0	0	34	204.22	204.22	1	2.18
3	10	2	-9.11	0	8	5.86	16.47	3	1846.98	-9.11	0	2	31.16	44.30	1	2.88
3	10	3	12.19	13.69	2	25.12	26.79	1	1991.55	-1.32	0	10	51.78	53.80	1	2.77
3	10	4	0	0	14	161.04	161.04	1	2247.66	0	0	7	95.09	95.09	1	2.89
3	10	6	0	0	48	169.27	169.27	1	1821.77	0	0	38	170.88	170.88	2	2.62
3	12	2	-3.29	0	3	4.74	8.30	2	2333.18	1.85	5.31	1	44.07	48.97	1	3.20
3	12	3	22.11	22.31	1	29.34	29.56	1	2565.61	-0.17	0	10	45.22	45.46	2	2.96
3	12	4	46.23	46.23	10	79.99	79.99	3	2958.08	0	0	1	60.56	60.56	1	2.83
3	12	6	0	0	45	127.72	127.72	2	2602.28	0	0	40	90.38	90.38	1	3.31
Average			8.27	9.51	27.90	61.87	63.29	11.60	1443.93	-0.89	0.27	23.95	105.28	106.99	1.75	2.25

Table B.15: Comparison of H_1 and H_2 for Simulated Data Sets with 500 Data Points and 4 Clusters

p	m	q	**H_1**							**H_2**						
			best			*worst*			*CPU s.*	*best*			*worst*			*CPU s.*
			$\%Gap_M$	$\%Gap_B$	*Hits*	$\%Gap_M$	$\%Gap_B$	*Hits*		$\%Gap_M$	$\%Gap_B$	*Hits*	$\%Gap_M$	$\%Gap_B$	*Hits*	
4	4	2	0	0	18	70.54	70.54	1	1106.35	0	0	23	110.09	110.09	1	1.40
4	4	3	0	0	49	183.92	183.92	1	742.62	0	0	33	79.56	79.56	9	1.25
4	5	2	-0.46	0	14	148.18	149.32	1	1419.11	-0.46	0	30	128.61	129.66	1	1.61
4	5	3	0	0	43	186.28	186.28	1	1316.71	0	0	38	142.43	142.43	2	1.59
4	5	4	0	0	22	131.23	131.23	1	1091.17	0	0	29	251.80	251.80	1	1.52
4	6	2	23.64	26.51	1	64.01	67.80	2	1838.96	-2.26	0	15	150.78	156.59	1	1.67
4	6	3	0	0	42	150.88	150.89	1	1917.00	0	0	32	132.94	132.94	1	1.65
4	6	4	0	0	30	153.40	153.40	1	1700.05	0	0	27	264.17	264.17	1	1.58
4	8	2	35.83	39.41	1	56.94	61.08	1	2607.32	-2.57	0	3	66.81	71.21	1	2.50
4	8	3	52.99	52.99	1	79.42	79.42	1	2694.85	0	0	17	108.67	108.67	1	2.28
4	8	4	0	0	44	73.26	73.26	1	3488.13	0	0	32	168.87	168.87	1	2.20
4	8	6	0	0	38	167.30	167.30	2	2611.03	0	0	29	245.43	245.43	1	2.31
4	10	2	10.28	19.42	12	16.99	26.69	1	4428.64	-7.65	0	1	40.94	52.62	1	2.74
4	10	3	32.12	33.18	1	80.54	81.99	1	4402.19	-0.80	0	12	82.95	84.42	1	2.61
4	10	4	0	0	9	138.28	138.28	2	3915.51	0	0	26	118.91	118.91	1	2.81
4	10	6	0	0	35	141.54	141.55	4	4097.90	0	0	3	194.40	194.40	1	2.92
4	12	2	-21.38	0	2	-9.85	14.66	1	6442.01	-14.96	8.17	1	5.56	34.27	1	3.78
4	12	3	35.80	31.46	1	58.67	53.60	1	4349.86	3.30	0	16	45.04	40.40	1	3.00
4	12	4	26.58	26.58	1	96.36	96.36	2	5884.97	0	0	13	110.22	110.22	1	3.09
4	12	6	0	0	45	105.05	105.05	1	5484.87	0	0	1	156.54	156.54	1	2.99
Average			9.77	11.48	20.45	104.65	106.63	1.35	3076.96	-1.27	0.41	19.05	130.24	132.66	1.45	2.27

Table B.16: Comparison of H_1 and H_2 for Simulated Data Sets with 1000 Data Points and 2 Clusters

p	m	q	**H_1**							**H_2**						
			best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
2	4	2	-0.02	0	37	59.09	59.11	13	644.94	-0.02	0	17	66.92	66.95	1	6.77
2	4	3	0	0	50	0	0	50	391.25	0	0	50	0	0	50	5.98
2	5	2	-0.22	0	50	-0.22	0	50	902.34	-0.22	0	31	22.00	22.26	1	7.64
2	5	3	0	0	50	0	0	50	629.09	0	0	50	0	0	50	7.37
2	5	4	0	0	50	0	0	50	497.03	0	0	50	0	0	50	6.44
2	6	2	-0.02	0	50	-0.02	0	50	702.28	-0.02	0	48	59.70	59.73	2	7.70
2	6	3	-0.05	0	50	-0.05	0	50	932.35	-0.05	0	41	8.62	8.67	7	7.02
2	6	4	0	0	50	0	0	50	729.56	0	0	50	0	0	50	7.62
2	8	2	8.08	10.37	5	28.40	31.11	1	1150.39	-2.07	0	8	32.63	35.44	2	10.38
2	8	3	-0.04	0	33	88.03	88.10	7	1835.99	-0.04	0	46	84.39	84.46	1	9.89
2	8	4	0	0	50	0	0	50	1383.59	0	0	50	0	0	50	8.83
2	8	6	0	0	50	0	0	50	940.39	0	0	50	0	0	50	9.13
2	10	2	16.66	17.74	13	17.69	18.79	5	1494.65	-0.93	0	17	27.76	28.95	10	10.80
2	10	3	-0.01	0	35	41.25	41.27	5	2092.51	-0.01	0	39	41.48	41.50	1	11.08
2	10	4	0	0	12	60.87	60.87	2	1456.84	0	0	21	57.92	57.92	2	11.67
2	10	6	0	0	50	0	0	50	1622.21	0	0	50	0	0	50	10.31
2	12	2	15.06	15.06	10	31.71	31.71	1	2345.57	0	0	9	33.25	33.25	1	13.34
2	12	3	22.03	22.33	5	31.11	31.44	6	2181.98	-0.25	0	30	35.80	36.14	1	12.35
2	12	4	0	0	26	63.91	63.91	1	2857.47	0	0	44	69.95	69.95	1	13.46
2	12	6	0	0	48	96.16	96.16	1	2535.18	0	0	2	4.02	4.02	48	12.05
Average			3.07	3.27	36.20	25.90	26.12	27.10	1366.28	-0.18	0	35.15	27.22	27.46	21.40	9.49

Table B.17: Comparison of H_1 and H_2 for Simulated Data Sets with 1000 Data Points and 3 Clusters

p	m	q	**H_1**							**H_2**						
			best			*worst*			*CPU s.*	*best*			*worst*			*CPU s.*
			$\%Gap_M$	$\%Gap_B$	*Hits*	$\%Gap_M$	$\%Gap_B$	*Hits*		$\%Gap_M$	$\%Gap_B$	*Hits*	$\%Gap_M$	$\%Gap_B$	*Hits*	
3	4	2	-0.14	0	50	-0.14	0	50	1206.39	-0.14	0	25	171.25	171.63	1	7.23
3	4	3	0	0	50	0	0	50	1001.50	0	0	35	257.61	257.61	2	5.38
3	5	2	28.81	30.44	8	37.10	38.83	12	1822.33	-1.25	0	18	43.71	45.53	2	6.62
3	5	3	0	0	50	0	0	50	1438.85	0	0	40	117.83	117.83	1	7.11
3	5	4	0	0	50	0	0	50	1281.89	0	0	38	156.83	156.83	1	6.36
3	6	2	43.78	45.73	1	46.31	48.30	1	1970.56	-1.34	0	25	74.01	76.38	1	7.17
3	6	3	-0.12	0	50	-0.12	0	50	2154.87	-0.12	0	34	113.99	114.24	1	7.26
3	6	4	0	0	50	0	0	50	2065.63	0	0	38	203.90	203.90	1	7.48
3	8	2	23.77	25.55	3	26.27	28.09	1	4038.01	-1.42	0	15	44.32	46.40	1	10.63
3	8	3	0	0	1	81.54	81.54	1	3797.14	0	0	38	77.19	77.19	1	10.28
3	8	4	0	0	45	170.21	170.21	1	3668.14	0	0	38	173.33	173.33	1	9.01
3	8	6	0	0	44	232.92	232.92	6	3272.20	0	0	32	233.76	233.76	1	9.46
3	10	2	-10.53	0	2	0.89	12.76	1	5380.27	-10.53	0	1	17.05	30.82	2	11.98
3	10	3	60.32	61.84	1	77.49	79.16	1	5942.79	-0.94	0	31	53.50	54.95	1	10.42
3	10	4	0	0	43	137.46	137.46	1	5455.62	0	0	34	130.11	130.11	1	10.26
3	10	6	0	0	38	184.08	184.08	2	5559.02	0	0	33	185.14	185.14	1	9.87
3	12	2	-7.73	0	11	-2.74	5.41	1	7052.24	-2.74	5.41	1	38.60	50.21	1	14.72
3	12	3	19.90	20.85	12	27.81	28.82	2	6084.52	-0.78	0	9	43.17	44.30	1	11.27
3	12	4	42.45	36.27	1	79.60	71.81	1	6810.91	4.54	0	30	65.43	58.25	1	12.16
3	12	6	0	0	39	-93.91	-93.91	1	TL	0	0	38	101.62	101.62	2	13.43
Average			10.03	11.03	27.45	50.24	51.27	16.60	3684.36	-0.74	0.27	27.65	115.12	116.50	1.20	9.40

Table B.18: Comparison of H_1 and H_2 for Simulated Data Sets with 1000 Data Points and 4 Clusters

p	m	q	**H_1**							**H_2**						
			best			*worst*			$CPU\ s.$	*best*			*worst*			$CPU\ s.$
			$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$		$\%Gap_M$	$\%Gap_B$	$Hits$	$\%Gap_M$	$\%Gap_B$	$Hits$	
4	4	2	-0.09	0	7	70.91	71.05	4	2781.26	-0.09	0	23	145.57	145.78	1	6.73
4	4	3	0	0	36	205.99	205.99	2	1852.29	0	0	30	287.80	287.80	2	5.75
4	5	2	3.97	4.56	23	63.05	63.98	1	3828.12	-0.56	0	27	113.11	114.32	1	7.71
4	5	3	0	0	42	188.69	188.69	1	3213.07	0	0	39	203.59	203.59	1	7.14
4	5	4	0	0	46	192.99	192.99	1	2383.87	0	0	38	284.71	284.71	2	6.54
4	6	2	25.01	26.41	1	64.94	66.79	1	4585.72	-1.11	0	10	145.37	148.12	1	7.20
4	6	3	-0.04	0	38	178.84	178.95	1	4063.95	-0.04	0	29	213.19	213.32	1	7.44
4	6	4	0	0	29	114.42	114.42	1	3367.08	0	0	34	218.65	218.65	1	6.62
4	8	2	23.25	24.29	1	152.42	154.54	1	TL	-0.83	0	1	91.49	93.10	1	10.11
4	8	3	35.90	35.92	1	101.33	101.36	1	6558.40	-0.01	0	27	100.61	100.63	1	10.23
4	8	4	0	0	34	124.16	124.16	1	TL	0	0	25	146.83	146.83	1	10.44
4	8	6	0	0	37	171.19	171.19	1	5599.69	0	0	30	248.24	248.24	1	9.26
4	10	2	7.02	7.66	3	16.70	17.40	1	TL	-0.60	0	1	46.68	47.56	1	11.61
4	10	3	-0.75	0	2	89.09	90.52	1	TL	-0.75	0	6	77.42	78.76	1	9.83
4	10	4	0	0	8	101.84	101.84	1	TL	0	0	4	106.17	106.17	1	10.64
4	10	6	0	0	39	3.61	3.61	1	TL	0	0	33	225.06	225.06	2	10.02
4	12	2	-21.04	0	1	-13.89	9.06	1	TL	-15.96	6.43	1	11.10	40.70	1	14.72
4	12	3	48.41	49.17	1	66.99	67.85	1	TL	-0.51	0	1	57.72	58.53	1	13.07
4	12	4	41.04	41.04	2	90.67	90.67	1	TL	0	0	23	108.62	108.62	1	12.03
4	12	6	0	0	23	105.16	105.16	1	TL	0	0	27	180.57	180.57	1	12.63
Average			8.13	9.45	18.70	104.46	106.01	1.20	3823.35	-1.02	0.32	20.45	150.62	152.55	1.15	9.49