A DC OPTIMIZATION APPROACH FOR CONSTRAINED BI-LEVEL HIERARCHICAL CLUSTERING WITH ℓ_1 NORM

Adugna Fita Gabissa

A Dissertation Submitted to the Department of Applied Mathematics, College of Applied Natural Science

Presented in Fulfillment of the Requirement for the Degree of Doctor of Philosophy in Applied Mathematics

Office of Graduate Studies

Adama Science and Technology University

December, 2024

Adama, Ethiopia

A DC OPTIMIZATION APPROACH FOR CONSTRAINED BI-LEVEL HIERARCHICAL CLUSTERING WITH ℓ_1 NORM

Adugna Fita Gabissa

Supervisor: Dr. Legesse Lemecha Obsu

Co-Supervisor: Dr. Wondimagegnehu Geremew

(Stockton University, USA)

A Dissertation Submitted to the Department of Applied Mathematics, College of Applied Natural Science

Presented in Fulfillment of the Requirement for the Degree of Doctor of Philosophy in Applied Mathematics (Optimization)

Office of Graduate Studies

Adama Science and Technology University

December, 2024

Adama, Ethiopia

Declaration

I hereby declare that this dissertation entitled "**A DC Optimization Approach for Constrained Bi-level Hierarchical Clustering with ℓ_1 norm**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials used for this dissertation have been duly acknowledged through appropriate citations.

Adugna Fita Gabissa	_____	_____
Name of student	Signature	Date

Recommendation

I, the supervisor of this dissertation, hereby certify that I have read and revised the dissertation entitled"**A DC Optimization Approach for Constrained Bi-level Hierarchical Clustering with ℓ_1 norm**" prepared under my guidance by **Adugna Fita Gabissa** submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in **Applied Mathematics**. Therefore, I recommend the submission of the dissertation to the department for further review and defense.

Major Supervisor	Signature	Date

Approval Sheet

I/we hereby certify that the recommendations and suggestions made by the board of examiners are appropriately incorporated into the final version of the dissertation entitle **A DC Optimization Approach for Constrained Bi-level Hierarchical Clustering with ℓ_1 norm by Adugna Fita Gabissa**.

Major Supervisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the dissertation open defense by **Adugna Fita Gabissa** have read and evaluated the dissertation entitled "**A DC Optimization Approach for Constrained Bi-level Hierarchical Clustering with ℓ_1 norm**" and examined the candidate during open defense. This is, therefore, to certify that the dissertation is accepted for partial fulfillment of the requirement of the degree of Doctor of Philosophy in **Applied Mathematics**.

Chairperson	Signature	Date
Internal Examiner 1	Signature	Date
Internal Examiner 2	Signature	Date
External Examiner 1	Signature	Date
External Examiner 2	Signature	Date

Finally, approval and acceptance of the dissertation is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the candidate's Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my supervisor Dr. Legesse Lemcha for making this research work possible. His guidance, valuable suggestions and great support has been a good influence for this work to be successfully done. He kindly read this final dissertation and offered valuable detailed advice on the organization and the theme of the whole dissertation. This dissertation is made possible through his help and support.

I would also like to thank my co-supervisor Dr. Wondimagegnehu Geremew for his guidance and support during my Ph.D studies.

Many thanks go to all my colleagues at Department of Applied Mathematics for the good working atmosphere as well as for their friendly and unreserved support.
I also very grateful to Adama Science and Technology University(ASTU) for providing an excellent platform for studies, research and opportunity for education.

Finally, special thanks to my wife Almaz Hailu, my sons Beka and Lemi for their emotional support and patience during these years where most of my time has been devoted to the research work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Algorithms

LIST OF ACRONYMS

aff(F)	Affine hull of F
co(F)	Convex hull of F
DC	Difference-of-Convex
DCA	Difference of Convex Algorithm
dom(f)	Domain of f
HQ	Headquarter
int(F)	Interior of F
NP	Non polynomial time
ri(F)	Relative interior of F

Abstract

Clustering problems often face challenges due to the non-smooth nature of distance measures like the ℓ_1 norm. To address this issue, we employ Nesterov's partial smoothing technique to approximate the ℓ_1 norm with a smoother function, thereby facilitating more effective mathematical optimization. This dissertation presents two main contributions. We model and solved clustering problems where nodes are identified as cluster centers to minimize the overall ℓ_1 distance between nodes within clusters. The proposed approach uses Nesterov's technique to smooth the ℓ_1 norm and improve optimization efficiency. We investigate two network topologies for bi-level hierarchical clustering. The first topology chooses cluster centers first, then a headquarter from cluster centers that functions as both a cluster center and a headquarter node that minimize the ℓ_1 distances within the network. The second topology locates a distinct headquarter node within an average of these cluster centers that minimizing the total network cost. In both models, we introduce a penalty parameter to ensure that the cluster centers and headquarter are adjusted to actual network nodes. As a result we employed three Difference of Convex Algorithm (DCA) based methods to find optimal cluster centers and headquarters efficiently. The numerical experiments demonstrate that our algorithms achieve high-quality solutions and significantly reduce iteration times by constraining the search space. While our methods show promising results, future work will address the remaining challenges of selecting optimal smoothing and penalty parameter. Beside, future work will focus on the challenge of applying non-smooth distance measures to large-scale geographic clustering applications such as urban planning and logistics.

Keywords. Clustering, DC programming, bi-level hierarchical, headquarter, Nesterov's smoothing, Sub-gradient, Fenchel conjugate.

CHAPTER 1

INTRODUCTION AND PRELIMINARIES

This chapter provides the background of the study and outlines the statement of the problem. In addition, it highlights the desired features in convex analysis and survey key terms and results for understanding later chapters of the dissertation. We mainly focus on convex optimization tools for non-smooth and non-convex problem solving.

1.1 Introduction

Clustering is a fundamental problem in unsupervised learning which has many applications in various domains. It is widely used in various fields, such as pattern recognition in machine learning, facility location for logistics, and image processing for data segmentation and many other areas of science and technology. It is an unsupervised partitioning technique dealing with the problems of organization of a collection of patterns into groups based on similarities or distance (Xu & Wunsch, 2005).

The similarity measure in clustering is essentially defined using different norms. Clustering problems with the similarity measure defined by the squared Euclidean norm are known as the minimum sum-of-squares clustering problems. Traditional clustering models, such as K-means and hierarchical clustering, may suffer from poor performance because of the non-smoothness of the models and the difficulties in finding global optimal solutions for such models. The clustering results are generally highly dependent on the initializations and the results could differ significantly with different initializations (Tran, 2020). There are many algorithms for solving such problems including heuristics such as the K-means algorithm and its modifications, meta-heuristics such as the simulated annealing, tabu search, genetic algorithms and optimization algorithms such as the branch and bound, cutting plane and interior point (see (Jain, 2010; Ordin & Bagirov, 2015; Teboulle, 2007) and references therein). But some of them are non polynomial time and difficult to check for optimality.

To overcome these problems (Sun et al., 2021) develop a semi-smooth Newton based augmented Lagrangian method for solving large-scale convex clustering problems. Extensive numerical experiments on both simulated and real data demonstrate that their algorithm is highly efficient and robust for solving large-scale problems. Moreover, the numerical results also show the superior performance and scalability of their algorithm comparing to the existing first-order methods. The complexity of modern networks such as communication networks, broadcasting networks, and distribution networks requires multilevel connectivity.

For efficiency purposes, the delivery company wants to identify a certain number of locations to serve as distribution centers for the delivery of supplies to the stores. At the same time, the company also wants to identify a location as a main distribution center, known as the headquarter, from which the other distribution centers receive their supplies. This is a typical description of a bi-level communication network, which can also be seen as a multi-facility location problem or as a bi-level hierarchical clustering problem (Nam et al., 2017).
Bi-level hierarchical clustering problem can be described mathematically as follows. Given m nodes $a^1, \ldots, a^m$ in $\mathbb{R}^n$, the objective is to choose k cluster centroids $a^{(1)}, \ldots, a^{(k)}$ and a headquarter $a^{(k+1)}$ from the given nodes in such a way that the total transportation cost of the tree formed by connecting the cluster centers to the total center, and the remaining nodes to the nearest cluster centers is minimized. The fact that the centers and the total center have to be among the existing nodes makes the problem a discrete optimization problem, which can be shown to be NP-hard (Geremew et al., 2018).
Given a finite number of data points with a measurement distance, a centroid-based clustering problem seeks a finite number of cluster centers with each data point assigned to the nearest cluster center in a way that a certain measurement distance is minimized. This leads to the optimization of non-smooth problems which is difficult to solve to optimality without the presence of convexity and smoothness.
The notion of sub-differential (collection of sub-gradients) for non-differentiable convex functions was independently introduced and developed by Moreau and Rockafellar who were both influenced by Fenchel, (B. Mordukhovich & Nam, 2014). Since then this notion has become one of the most central concepts of convex analysis and its various applications. The underlying difference between the standard derivative of a differentiable function and the sub-differential of a convex function at a given point is that the sub-differential is a set which reduces to a singleton (gradient) if the function is differentiable. Due to the set-valuedness of the sub-differential, deriving calculus rules for it is a significantly more involved task in comparison with classical differential calculus.
The first and the most important result of convex sub-differential calculus is the sub-differential sum rule, which was obtained at the very beginning of convex analysis and has been since known as the Moreau-Rockafellar theorem. The reader can find this theorem and other results of convex sub-differential calculus in finite-dimensional spaces in the now classical monograph by (Rockafellar, 1970)
Recently, non-smooth analysis and optimization have become increasingly important for applications to many new fields such as clustering, image processing, computational statistics, machine learning, computer vision and sparse optimization. While the objective function involved in those problems are most frequently non-differentiable. A natural way to cope with the non-differentiability in optimization is to approximate non-smooth objective functions by smooth functions that are favorable for applying smooth optimization schemes (Geremew et al., 2018). Along with the difficulty in dealing with non differentiability, another challenge in

modern optimization is to go from convexity to non-convexity as non-convex optimization techniques and algorithms allow us to solve more complex optimization problems arising naturally in many practical applications. Non-smoothness and non-convexity require new optimization methods capable of handling a broader class of functions and sets where convexity is not assumed. One of the most successful approaches to go beyond convexity is to consider the class of DC functions, where DC stands for difference of convex (Jain, 2010).
Although the role of DC functions had been known earlier in optimization theory, the first algorithmic approach was developed by Pham Dinh Tao in 1985. The algorithm introduced by Pham Dinh Tao for minimizing $f = g - h$, where $g, h : \mathbb{R}^n \to R$ are convex functions is called the DCA, which is based on sub-gradients of the function h and sub-gradients of the Fenchel conjugate of the function g. This approach leads to the nice and elegant concept of approximating a non-convex DC program by a sequence of convex ones for the construction of DCA. It turns out that with appropriate DC decompositions and suitably equivalent DC reformulations, DCA permits to recover most of standard methods in convex and non-convex programming. These theoretical and algorithmic tools, constituting the backbone of non-convex programming and global optimization (L. T. H. An & Tao, 2005).
Although convex and smooth optimization techniques and numerical algorithms have been the topics of extensive research for more than 50 years, solving large-scale optimization problems without the presence of convexity and smoothness remains a challenge. Many recent real world applications require solving optimization problems in which the objective function is non-convex and non-smooth.
This is a motivation to search for new optimization methods that are capable of handling broader classes of functions and sets where convexity and smoothness are not assumed. One such method is to consider the class of functions that can be represented as a difference of two convex functions, DC functions for short. Mathematical programming problems dealing with DC functions are called DC programming problems. It is known that the set of DC functions defined on a compact convex set of $\mathbb{R}^n$ is dense in the set of continuous functions on this set (Horst & Thoai, 1999). Therefore, in principle, every continuous function can be approximated by a DC function with any desired precision. Moreover, every C^2-function is a DC function. Although DC representations are available for important function classes, finding such a representation for an arbitrary DC function is a hard open problem (Horst & Thoai, 1999).
DC decomposition helps to apply the DCA algorithm that finds the minimum value of DC functions. Fortunately, the objective functions for the three models under consideration are DC functions. To address the lack of smoothness, we propose the use of a smoothing approximation technique commonly known as Nesterov's smoothing technique, and attempted to develop algorithms that can be implemented by writing MATLAB codes to find optimal value for the three proposed models.
Therefore, we proposed three different models that produces efficient DC algorithm by im-

posing constraints that keep the root (or headquarter) of the tree in the convex hull of the centers for bi-level hierarchal clustering models.

1.2 Statement of the Problem

One of the most well-known and commonly used algorithms for solving centroid-based clustering problems, like the models we are considering, is the K-means algorithm. Although the K-means algorithm is one of the simplest and cheapest algorithms, providing an easy way to classify a given data set through a certain number of clusters, it has a list of drawbacks: the K-means algorithm depends heavily on the initial choice of clusters.The K-means algorithm depends heavily on the initial choice of clusters, and there is no guarantee that it converges to a global optimal solution (L. An et al., 2007; Kashima et al., 2008). The other drawback is the result depend heavily on the distance measurement,the algorithm may not converge if a different distance function other than the Euclidean norm is used (Tran, 2020).
Our suggested approach was anticipated to address the drawbacks of the widely-used K-means technique and its variations for clustering. Furthermore, even though my dissertation focused on geographic locations, the concept may be applied to any numerical data of any size as long as we have a suitable distance measure. Furthermore, it is more practical to use the ℓ_1 norm in many real-world applications (Jörnsten, 2004; Zhang et al., 2012).
Consider either a wireless or cable computer. The connection among computers in the network is not necessarily direct; there are several bends and blockages on the way. So the use of Euclidean norm, which is a measure of the direct distance between two geographical locations, is an approximation of the reality. Depending on the application and the problem, there are several other distance measures that can give us a better approximation of the reality than the Euclidean distance. For instance, the city block distance (also known as Manhattan distance) is one example.
To address these limitations, this dissertation explores a novel DC optimization approach for constrained bi-level hierarchical clustering. Specifically, it introduces the use of the ℓ_1 norm for distance measures and applies Nesterov's smoothing technique to overcome the challenges of non-smooth clustering. By developing new algorithms based on Difference of Convex (DC) programming, this research aims to provide more robust and efficient clustering methods, particularly for large-scale, real-world data.

- Improving the drawback of k-mean algorithms and ℓ_2-norm clustering.
- Using ℓ_l norm to overcame the drawback of squared euclidean norm in clustering.
- How can we determine cluster centers and headquarter that are less sensitive to outliers?

1.3 Objectives of the Study

1.3.1 General Objective

The primary objective of this study is to investigate clustering and bi-level hierarchical clustering problems under constraints by enhancing existing models that utilize the ℓ_1 norm for distance measurement.

1.3.2 Specific Objectives

The specific objectives of the study are to:

1. formulate clustering problem with ℓ_1 norm and solve using DC optimization.
2. develop two distinct network topologies for bi-level hierarchical clustering models using ℓ_1 norm, with the goal of reducing computational complexity and improving clustering accuracy in non-smooth scenarios.
3. implement numerical simulation of the three models using MATLAB software.
4. test the constructed models using GPS coordinates of towns and cities that were gathered from some towns and cities in Oromia / Ethiopia.

1.4 Significance of the Study

One of the most well-known and commonly used algorithms for solving centroid-based clustering problems, like the models we are considering, is the K-means algorithm. Although the K-means algorithm is one of the simplest and cheapest algorithms, providing an easy way to classify a given data set, it has a list of drawbacks. In K-means clustering, using Manhattan distance instead of Euclidean distance can produce better results, especially when dealing with sparse high-dimensional data or when outliers are present.
Also, it's often preferred in text classification and document clustering due to its effectiveness with sparse vector spaces. The K-means algorithm does not always converge to a global optimal solution; it depends heavily on the initial cluster selection. Its results rely heavily on the distance measurement, and if a distance function other than the Euclidean norm is used, it might not converge.
The method we are proposed is expected to overcome many of the disadvantages of the K-means algorithm and its variants that are routinely used for clustering. From a practical standpoint, ℓ_1 regularization induces sparsity than ℓ_2 which tends to shrink coefficients evenly.
Therefore, ℓ_1 is useful for feature selection, as we can drop any variables associated with

coefficients that go to zero. Moreover, even though we considered geographical locations in this project, the idea can be extended to any numerical data of any dimension, as long as we have an appropriate distance measure (a measure of similarity or dissimilarity) among different observations. In addition, using ℓ_1 norm is more realistic in many application on ground.

1.5 The Scope of the Study

This study investigated clustering and bi-level hierarchical clustering problems through DC programing and DCA while the objective functions and constraints are non-smooth and non-convex and numerical algorithm implemented in $\mathbb{R}^2$.

1.6 Organizations of the dissertation

The dissertation is structured as follows: first, key concepts in convex analysis and DC programming are discussed, followed by the development of three new clustering models based on ℓ_1 norm. The effectiveness of these models is validated through numerical experiments using data from Oromia Regional State. In the chapter two we discuss full literature reviews. The methodology used in conducting this dissertation is presented in chapter three. In the fourth chapter we study clustering problem formulation and analysis to develop DCA algorithms that solve the problem based on ℓ_1 distance and Nesterov's smoothing technique. In chapter five and six we mainly focus on formulation and study of different network topologies of bi-level hierarchical clustering problems and developing DCA algorithms that solve the two new models based on Nesterov's smoothing technique and ℓ_1 distance. In final chapter, we end the dissertation with conclusions and future work.

1.7 Mathematical Preliminaries

In this section, we present some important definitions and theorems which are used in this dissertation as an input from existing literature. Convex analysis was developed by (Rockafellar, 1970) and (Moreau, 1963) independently in the 1960s. Their development of the sub-differential, which is a useful concept in non-smooth analysis and optimization theory, generalizes the idea of the derivative in classical calculus from differentiable functions to functions that are convex but not necessarily differentiable. This section provides the mathematical foundation for convex analysis that will be used throughout this dissertation. More details of convex analysis can be found in (J. B. Hiriart-Urruty & LemarÃl'chal, 2001; B. Mordukhovich & Nam, 2014; B. S. Mordukhovich, 2006; Nam et al., 2017; Rockafellar, 1970).

1.7.1 Basics of Convex Set

We begin with a systematic study of convex sets and functions with the discussion beginning with sets. Throughout this document we deal with extended-real-valued functions $f : \mathbb{R}^n \to (-\infty, \infty] = \bar{\mathbb{R}}$ and with the following arithmetic conventions: $(+\infty) - (+\infty) = +\infty$, $(+\infty) - \alpha = +\infty$ and $\alpha - (+\infty) = -\infty$, $\forall \alpha \in (-\infty, +\infty)$.
Throughout the document, consider the Euclidean space $\mathbb{R}^n$ of $n-$tuples of real numbers with the inner product,

$$\langle x, y \rangle := \sum_{i=1}^{n} x_i y_i \text{ for } x = (x_1, \ldots, x_n) \in \mathbb{R}^n \text{and } y = (y_1, \ldots, y_n) \in \mathbb{R}^n.$$

The Euclidean norm or ℓ_2 norm induced by this inner product is defined as usual by

$$||x||_2 := \sqrt{\sum_{i=1}^{n} x_i{}^2}.$$

And the ℓ_1 norm in $\mathbb{R}^n$ is defined as,

$$\|x\|_1 := \sum_{i=1}^{n} |x_i| \quad = \quad |x_1| + |x_2| + \cdots + |x_n|,$$

where we often identify each element $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ with the column $x = [x_1, \ldots, x_n]^T$.

Definition 1.7.1. *The closed ball centered at $\bar{x}$ with radius $r > 0$ and the closed unit ball of $\mathbb{R}^n$ are defined,respectively, by*

$$\mathbb{B}(\bar{x}, r) := \{x \in \mathbb{R}^n : ||x - \bar{x}|| \leq r\} \text{ and } \mathbb{B} := \{x \in \mathbb{R}^n : ||x|| \leq 1\}.$$

Given two elements x and y in $\mathbb{R}^n$, define the interval / line segment

$$[x, y] := \{\lambda x + (1 - \lambda) y \mid \lambda \in [0, 1]\}.$$

Definition 1.7.2. *A subset Ω of $\mathbb{R}^n$ is called convex if $\lambda x + (1 - \lambda) y \in \Omega$ for all $x, y \in \Omega$ and $\lambda \in [0, 1]$.*

Given $\omega_1, ..., \omega_m \in \mathbb{R}^n$, the element $x = \sum_{i=1}^{m} \lambda_i \omega_i$, where $\sum_{i=1}^{m} \lambda_i = 1$ and $\lambda_i \geq 0$ for some $m \in \mathbb{N}$, is called a convex combination of $\omega_1, ..., \omega_m$.

Proposition 1.7.1. *A subset F of $\mathbb{R}^n$ is convex if and only if it contains all convex combinations of its elements.*

Proof. B. Mordukhovich & Nam (2014) Suppose that F contains all convex combination of its elements. Fix $x, y \in F$ and $\lambda \in (0, 1)$ then $\lambda x + (1 - \lambda) y \in F$ hence F is convex.

Suppose that F is convex, we show by induction that any convex combination $x = \sum_{i=1}^{m} \lambda_i \omega_i$ of elements in F is an element of F. This is conclusion follows directly from the definition for $m = 1,2$. Fix now a positive integer $m \geq 2$ and suppose that every convex combination of $k \in N$ elements from F, where $k \leq m$, belongs to F. Form the convex combination

$$y := \sum_{i=1}^{m+1} \lambda_i \omega_i, \quad \sum_{i=1}^{m+1} \lambda_i = 1, \quad \lambda_i \geq 0,$$

and observe that if $\lambda_{m+1} = 1$, then $\lambda_1 = \lambda_2 = \cdots = \lambda_m = 0$, so $y = \omega_{m+1} \in F$. In the case where $\lambda_{m+1} < 1$ we get the representations,

$$\sum_{i=1}^{m} \lambda_i = 1 - \lambda_{m+1} \text{ and } \sum_{i=1}^{m} \frac{\lambda_i}{1-\lambda_{m+1}} = 1,$$

which imply in turn the inclusion

$$z = \sum_{i=1}^{m} \frac{\lambda_i}{1-\lambda_{m+1}} \omega_i \in F.$$

It yields therefore the relationships

$$\begin{aligned} y &= (1-\lambda_{m+1}) \sum_{i=1}^{m} \frac{\lambda_i}{1-\lambda_{m+1}} \omega_i + \lambda_{m+1} \omega_{m+1}, \\ &= (1-\lambda_{m+1}) z + \lambda_{m+1} \omega_{m+1} \in F, \end{aligned}$$

this completes the proof of the proposition. □

Definition 1.7.3. *Let $F \subset \mathbb{R}^n$. The convex hull of F is defined and denoted by*

$$co(F) := \bigcap \{C : C \text{ is convex and } C \subset F\}.$$

Proposition 1.7.2. *Let $F \subset \mathbb{R}^n$. The convex hull $co(F)$ is the minimal convex set containing F.*

Proof. Define $\bar{C} := \{C : C \text{ is convex and } C \subset F\}$.Let $a, b \in co(F) = \bigcap \bar{C}$ then $a, b \in C$ for all $C \in \bar{C}$. Then for any $\lambda \in (0,1)$ we see $\lambda a + (1-\lambda)b \in C$. This shows $co(F)$ is convex. Further, for any $x \in co(F)$ we see $x \in C$ whenever $F \subset C$, by properties of set intersection. Thus, $co(F)$ is minimal. □

Lemma 1.7.1. *(Intersection of convex sets). Let $F_1, ..., F_m$ be convex subsets of $\mathbb{R}^n$,then $\bigcap_{i=1}^{m} F_i$ is convex.*

Proof. Let take $x, y \in \bigcap_{i=1}^{m} F_i$ and $\lambda \in [0,1]$, then $\lambda x + (1-\lambda)y \in F_i$ by convexity of each F_i for all $i = 1,2,...,m$. Thus, $\lambda x + (1-\lambda)y \in \bigcap_{i=1}^{m} F_i$. Therefore, intersection of convex sets is convex. □

1.7.2 Convex Functions

Convex functions play an important role in many fields of mathematics such as optimization, control theory, operations research,functional analysis etc. They have a lot of interesting and fruitful properties, such as continuity and differentiability properties or the fact that a local minimum becomes a global minimum.
In this section we deal with extended-real-valued functions $f : \mathbb{R}^n \to (-\infty, \infty] = \bar{\mathbb{R}}$.

Definition 1.7.4. *Let $\Omega \subset \mathbb{R}^n$ be a convex set and let $f : \Omega \to \mathbb{R}$ be a function. Then*

(i) A function f is said to be convex on Ω if,

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y),$$

for all $x, y \in \Omega$ and $\lambda \in [0,1]$, The function f is called concave if $-f$ is convex.
(ii) The set defined by,

$$\text{dom} f = \{x \in \mathbb{R}^n : f(x) < \infty\},$$

is called the effective domain of f.
(iii) A function f is said to be proper if dom $f \neq \emptyset$, that is, the function is not identically equal to $+\infty$ and $f(x) > -\infty$ for all $x \in \mathbb{R}^n$.

Example 1.7.1. *Let $\Omega = \mathbb{R}$ and let $f(x) = \|x\|$ for $x \in \mathbb{R}$. Then f is a convex function. Indeed, for any $x, y \in \mathbb{R}$ and $\lambda \in (0,1)$,*

$$\begin{aligned} f(\lambda x + (1-\lambda)y) &= \|\lambda x + (1-\lambda)y\| \\ &\leq \|\lambda x\| + \|(1-\lambda)y\| \\ &= \lambda \|x\| + (1-\lambda)\|y\| \\ &= \lambda f(x) + (1-\lambda) f(y), \end{aligned}$$

therefore, f is convex

Definition 1.7.5. *The epigraph of the function f is the set*

$$epi f = \{(x, \alpha) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq \alpha\}.$$

Proposition 1.7.3. *Let f be a function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$, then f is convex if and only if epi$f$ and dom$f$ are convex.*

Proof. ($\Rightarrow$) Suppose that f be convex and $(x_1, \alpha_1), (x_2, \alpha_2) \in$ epi$f$ that is $f(x_1) < \alpha_1$ & $f(x_2) < \alpha_2$.

Let also be $\lambda \in (0,1)$. From the convexity of f and the definition of the epigraph we have

$$\begin{aligned} f(\lambda x_1 + (1-\lambda)x_2) &\leq \lambda f(x_1) + (1-\lambda) f(x_2) \\ &\leq \lambda \alpha_1 + (1-\lambda)\alpha_2. \end{aligned}$$

This means,

$$(\lambda x_1 + (1-\lambda)x_2, \lambda \alpha_1 + (1-\lambda)\alpha_2) \in \text{epi} f,$$

i.e. epi$f$ is convex.
($\Leftarrow$) Suppose that epi$f$ be convex. Let us take $x_1, x_2 \in \text{dom} f$ and choose a, b such that $f(x_1) \leq a$ and $f(x_2) \leq b$, i.e. $(x_1, a), (x_2, b) \in \text{epi} f$. By the assumptions follows,

$$(\lambda(x_1, a) + (1-\lambda)(x_2, b)) \in epif \text{ for all } \lambda \in [0,1],$$

and this implies

$$\begin{aligned} f(\lambda x_1 + (1-\lambda)x_2) &\leq \lambda f(x_1) + (1-\lambda) f(x_2) \ \ f \text{ is convex} \\ &\leq \lambda \alpha_1 + (1-\lambda)\alpha_2. \end{aligned}$$

Therefore, $\lambda x_1 + (1-\lambda)x_2 \in \text{dom} f$,then domain of f is convex. □

Definition 1.7.6. *(Lower semi-continuous function). A function $f : \mathbb{R}^n \to [-\infty, \infty]$ is called lower semi-continuous (l.s.c.) at $x \in \mathbb{R}^n$ if For all sequence $\{x_n\}$ which converges to x,*

$$f(x) \leq \liminf f(x_n).$$

The function f is said to be lower semi-continuous, if it is l.s.c. at x, for all $x \in \mathbb{R}^n$.

Proposition 1.7.4. *(Epigraph characterization). Let $f : \mathbb{R}^n \to [-\infty, \infty]$ any function f is l.s.c. if and only if its epigraph is closed.*

Proof. If f is l.s.c., and if $(x_n, t_n) \in epif$ and $(x_n, t_n) \to (\bar{x}, \bar{t})$, then$\forall n, \bar{t} \geq f(x_n)$. Consequently,

$$\bar{t} = \lim_{n\to\infty} \inf t_n \geq \lim_{n\to\infty} \inf f(x_n) \geq f(\bar{x}).$$

Thus, $(\bar{x}; \bar{t}) \in epif$, and epi f is closed.
Conversely, if f is not l.s.c., there exists an $x \in \mathbb{R}^n$, and a sequence $\{x_n\} \to x$, such that $f(x) > \liminf f(x_n)$,i.e., there is an $\varepsilon > 0$ such that for all $n \geq 0$,

$$\inf_{k \geq n} f(x_k) \leq f(x) - \varepsilon.$$

Thus, for all $n, \exists k_n \geq k_{n-1}, f(x_{k_n}) \leq f(x) - \varepsilon$. We have built a sequence $\{w_n\} = (x_{kn}, f(x) - \varepsilon)$, each term of which belongs to epi$f$, and which converges to a limit $\bar{w} = (f(x) - \varepsilon)$ which is outside the epigraph. Consequently, epi$f$ is not closed. □

Proposition 1.7.5. *Let $(f_i)_{i\in I}$ be a family of convex functions $\mathbb{R}^n \to \bar{\mathbb{R}}$, with I any set of index set. Then the upper hull of the family $(f_i)_{i\in I}$ is convex.*

Proof. Let $f = \sup_{i\in I} f_i$ be the upper hull of the family. $epif = \bigcap_{i\in I} epif_i$. Indeed,

$$\begin{aligned}(x,t) \in epif, \forall i \in I,& \\ t \geq f_i(x) \quad &\Leftrightarrow \quad (x,t) \in epif_i, \\ &\Leftrightarrow \quad (x,t) \in \bigcap_{i\in I} epif_i\end{aligned}$$

but any intersection of convex sets is convex . This show that epi$f$ is convex, i.e. f is convex. □

Definition 1.7.7. *(Sub-level set)The sub-level set of a function $f : \mathbb{R}^n \to \mathbb{R}$, where $\alpha \in \mathbb{R}$,is the set of points in $\mathbb{R}^n$at which the function is smaller or equal to α,*

$$L_\alpha = \{x : f(x) \leq \alpha\}.$$

Lemma 1.7.2. *Sub-level sets of convex functions are convex.*

Proof. If $x, y \in \mathbb{R}^n$ such that $f(x) \leq \alpha$ and $f(y) \leq \alpha$ then for any $\lambda \in (0,1)$

$$\begin{aligned}f(\lambda x + (1-\lambda)y) \quad &\leq \quad \lambda f(x) + (1-\lambda) f(y) \text{ by convexity of } f \\ &\leq \quad \lambda\alpha + (1-\lambda)\alpha = \alpha,\end{aligned}$$

$\lambda x + (1-\lambda)y \in L_\alpha$. Thus, L_α is convex.

□

Definition 1.7.8. *Let $\Omega \subset \mathbb{R}^n$ be an open subset and a function $f : \Omega \to \mathbb{R}$ is Lipschitz-continuous if there exists a positive constant $\ell > 0$ such that,*

$$|f(x) - f(y)| \leq \ell ||x - y||, \forall x, y \in \Omega.$$

Remark 1.7.1. *A locally Lipschitz function $f : \Omega \to \mathbb{R}$ is continuous on Ω. We are finally ready to state the main result of this section, that is a convex function is continuous on the interior part of its domain.*

Theorem 1.7.1. *Let $\Omega \subset \mathbb{R}^n$ be a convex set. A convex function $f : \Omega \longrightarrow \mathbb{R}$ is locally Lipschitz on the interior int(Ω).*

Proof. Step 1. We claim that f is locally bounded at every interior point $x \in \text{int}(\Omega)$. Indeed, let us consider the vertexes of a regular convex polygon with center x, $\varepsilon > 0$ is sufficiently small such that

$$x_i := x + \varepsilon e_i \text{ and } x_{i+n} := x - \varepsilon e_i,$$

for $i = 1, ..., n$, where $\{e_1, ..., e_n\}$ is the usual orthonormal basis of $\mathbb{R}^n$.
Define,

$$C := \text{co}(x_1, ..., x_n, x_{n+1}, ..x_{2n}),$$

and notice that, for every $y \in C$, by convexity we have,

$$f(y) \leq \sum_{i=1}^{2n} f(\lambda_i x_i) \leq \sum_{i=1}^{2n} \lambda_i f(x_i) \leq \max_{i=1,...,2n} f(x_i) \sum_{i=1}^{2n} \lambda_i := M,$$

where $\lambda_i \geq 0$ and $\sum_{i=1}^{2n} \lambda_i = 1$ which gives a local upper bound. Similarly, given $y, y' \in C$, such that

$$x = \frac{y+y'}{2} \Rightarrow f(x) = f(\frac{y+y'}{2}) \leq \frac{1}{2} f(y) + \frac{1}{2} f(y')$$

It follows that

$$f(y) \geq 2f(x) - \frac{1}{2} f(y') \geq 2f(x) - M,$$

which gives a local lower bound.
Step 2. The function f is locally bounded around $\bar{x} \in int(\Omega)$, hence there exist $\varepsilon > 0$ and $M \in \mathbb{R}^+$ such that

$$|f(y)| \leq M \ \ \forall y \in \mathbb{B}(\bar{x}, 2\varepsilon),$$

let $x, y \in \mathbb{B}(\bar{x}, \varepsilon)$, and $\alpha := \|x - y\| \neq 0$. Clearly,

$$z := x + \frac{\varepsilon}{\alpha}(x - y) \in \mathbb{B}(\bar{x}, 2\varepsilon) \Rightarrow x = \frac{\alpha}{\alpha + \varepsilon} z + \frac{\varepsilon}{\alpha + \varepsilon} y,$$

and thus by convexity of f we have,

$$f(x) \leq \frac{\alpha}{\alpha + \varepsilon} f(z) + \frac{\varepsilon}{\alpha + \varepsilon} f(y).$$

It follows that

$$\begin{aligned} f(x) - f(y) &\leq \frac{\alpha}{\alpha + \varepsilon} [f(z) - f(y)] \\ &\leq \frac{\alpha}{\alpha + \varepsilon} |f(z) - f(y)| \\ &\leq \frac{2M}{\varepsilon} \alpha = \frac{2M}{\varepsilon} ||x - y||, \end{aligned}$$

which means that $f(x) - f(y) \leq \ell ||x - y||$. The argument is symmetric with respect to x and

y ,and therefore we also have that $f(y)-f(x) \leq \ell||x-y||$. which implies $|f(x)-f(y|) \leq \ell||x-y||$, where $\ell = \frac{2M}{\varepsilon}$. □

1.7.3 Sub-differential of Convex Function

Definition 1.7.9. *A function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is Fréchet differentiable at $\bar{x} \in int(domf)$ if there exists an element $v \in \mathbb{R}^n$ such that,*

$$\lim_{x \to \bar{x}} \frac{f(x) - f(\bar{x}) - \langle v, x - \bar{x} \rangle}{||x - \bar{x}||} = 0.$$

Then v is called the Fréchet derivative of f at $\bar{x}$ and is denoted by $\nabla f(\bar{x})$.

Definition 1.7.10. *A vector $v \in \Omega \subset \mathbb{R}^n$ is a sub-gradient of a convex function $f : \Omega \to \bar{\mathbb{R}}$ at $\bar{x} \in dom(f)$ if it satisfies the inequality,*

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x} \rangle \ \ for\ all \ \ x \in \mathbb{R}^n.$$

The collection of sub-gradients is called the sub-differential of f at $\bar{x}$ and is denoted by $\partial f(\bar{x}) = \{v \in \mathbb{R}^n \mid v \text{ satisfies definition } 1.7.10 \}$.If $\bar{x} \notin \text{dom}(f)$, we set $\partial f(\bar{x}) = \emptyset$.

Remark 1.7.2. *If $f : \mathbb{R}^n \to \mathbb{R}$ is Fréchet differentiable at $\bar{x}$, then $\partial f(\bar{x}) = \{\nabla f(\bar{x})\}$.*

Proposition 1.7.6. *Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is convex and differentiable at $\bar{x} \in dom(f)$. Then $f(x) \geq f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x} \rangle$ for all $x \in \mathbb{R}^n$ with $\partial f(\bar{x}) = \{\nabla f(\bar{x})\}$.*

Proof. Since f is differentiable at $\bar{x}$, for any $\varepsilon > 0$ there exists $\delta > 0$ such that

$$-\varepsilon||x - \bar{x}|| \leq f(x) - f(\bar{x}) - \langle \nabla f(\bar{x}), x - \bar{x} \rangle \leq \varepsilon||x - \bar{x}||, \text{whenever} ||x - \bar{x}|| \leq \delta.$$

Define further the function

$$\psi(x) := f(x) - f(\bar{x}) - \langle \nabla f(\bar{x}), x - \bar{x} \rangle + \varepsilon||x - \bar{x}||,$$

for which $\psi(x) \geq \psi(\bar{x}) = 0$ whenever $x \in \mathbb{B}(\bar{x}, \delta)$. It follows from the convexity of ψ that $\psi(x) \geq \psi(\bar{x})$ for $x \in \mathbb{R}^n$ and thus,

$$\langle \nabla f(\bar{x}), x - \bar{x} \rangle) \leq f(x) - f(\bar{x}) + \varepsilon||x - \bar{x}||, \forall x \in \mathbb{R}^n.$$

Letting $\varepsilon \to 0^+$ gives us the inequality and shows that $\nabla f(\bar{x}) \in \partial f(\bar{x})$.
To show the remaining part, pick $v \in \partial f(\bar{x})$ then,

$$\langle v, x - \bar{x} \rangle \leq f(x) - f(\bar{x}), \forall x \in \mathbb{R}^n.$$

From the differentiability of f at $\bar{x}$ we have,

$$\langle v - \nabla f(\bar{x}), x - \bar{x} \rangle \leq \varepsilon ||x - \bar{x}||, \text{whenever} ||x - \bar{x}|| \leq \delta,$$

and so $||v - \nabla f(\bar{x})|| \leq \varepsilon$. This yields $v = \nabla f(\bar{x})$ since $\varepsilon > 0$ is arbitrary, the claimed relationship $\partial f(\bar{x}) = \{\nabla f(\bar{x})\}$. □

Theorem 1.7.2. *Let $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ be a convex function. The sub-differential $\partial f(\bar{x})$ is nonempty,convex and compact at every point $\bar{x} \in$ int(dom f).*

Proof. We first prove the convexity of $\partial f(\bar{x})$. Let $v_1, v_2 \in \partial f(\bar{x})$, and let $\lambda \in (0,1)$. then we have

$$f(x) \geq f(\bar{x}) + \langle v_1, x - \bar{x} \rangle \text{ and } f(x) \geq f(\bar{x}) + \langle v_2, x - \bar{x} \rangle, \forall x \in \mathbb{R}^n.$$

We multiply the first inequality by λ, the second by $(1-\lambda)$, and we sum them to find that

$$f(x) \geq f(\bar{x}) + \langle \lambda v_1 + (1-\lambda) v_2, x - \bar{x} \rangle$$

which implies $\partial f(\bar{x})$ is convex.
To show the boundedness and closedness. Let $(v_k) \in \partial f(\bar{x}), k \in \mathbb{N}$ be a converging sequence, and let $v := \lim_{k \to +\infty} v_k$.Then we have,

$$f(x) \geq f(\bar{x}) + \langle v_k, x - \bar{x} \rangle, \forall x \in \mathbb{R}^n,$$

and taking the limit for $k \to +\infty$ yields

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x} \rangle, \forall x \in \mathbb{R}^n,$$

thus $\partial f(\bar{x})$ is closed.
To prove the boundedness, from (Theorem 1.7.1) f is locally Lipschitz and suppose Lipschitz constant is m, hence for $\varepsilon > 0$ and $x \in \mathbb{B}(\bar{x}, \varepsilon)$ we have that,

$$\langle v, x - \bar{x} \leq f(x) - f(\bar{x}) \leq m ||x - \bar{x}||.$$

let $x := \bar{x} + \varepsilon \frac{v}{||v||} \in \mathbb{B}(\bar{x}, \varepsilon)$, and we have

$$\varepsilon ||v|| = \langle v, \varepsilon \frac{v}{||v||} \rangle \leq m ||\varepsilon \frac{v}{||v||}|| \Rightarrow ||v|| \leq m,$$

which means that $\partial f(\bar{x})$ is bounded, and thus compact. □

Example 1.7.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be defined by $f(x) = ||x||$, then*

$$\partial f(x) = \begin{cases} \mathbb{B}, & \text{if} x = 0, \\ \frac{x}{||x||}, & \text{otherwise.} \end{cases}$$

The equality is routine if $x \neq 0$ because the function is differentiable in this case. Our main concern is calculating the sub-differential at $x = 0$.
Let $v \in \partial f(0)$, then by definition we have $\langle v, x \rangle \leq ||x||$ for all $x \in \mathbb{R}^n$. This implies that

$$\begin{aligned} \langle v, v \rangle &\leq ||v||, \\ ||v|| &\leq 1, \text{ so } v \in \mathbb{B}. \end{aligned}$$

The opposite inclusion follows from the Cauchy-Schwartz inequality.
Let $v \in \mathbb{B}$,

$$\begin{aligned} \langle v, x - 0 \rangle = \langle v, x \rangle &\leq ||v||||x||, \\ &\leq ||x|| = f(x) - f(0) \forall x \in \mathbb{R}^n. \end{aligned}$$

Therefore $v \in \partial f(0)$ and we have $\partial f(0) = \mathbb{B}$.

Proposition 1.7.7. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and $\bar{x} \in dom f$ be a local minimizer of f. Then f attains its global minimum at this point.*

Proof. By our assumptions, for some $\delta > 0$ we have,

$$f(\bar{x}) \leq f(y), \forall y \in \mathbb{B}(\bar{x}, \delta).$$

Let $x \in \mathbb{R}^n$ and $z \in \mathbb{B}(\bar{x}, \delta)$ for $\lambda \in (0,1)$,

$$\begin{aligned} z = \lambda x + (1 - \lambda)\bar{x} \in \mathbb{B}(\bar{x}, \delta) & \\ f(\bar{x}) \leq f(z) &= f(\lambda x + (1 - \lambda)\bar{x}) \\ &\leq \lambda f(x) + (1 - \lambda) f(\bar{x}) \\ \lambda f(\bar{x}) &\leq \lambda f(x), \end{aligned}$$

which implies $f(\bar{x}) \leq f(x)$ for all $x \in \mathbb{R}^n$. □

Proposition 1.7.8. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and $\bar{x} \in dom f$. Then f attains its local/global minimum at $\bar{x}$ if and only if $0 \in \partial f(\bar{x})$.*

Proof. First suppose f attains its global minimum at $\bar{x}$. Then

$$0 = \langle 0, x - \bar{x} \rangle \leq f(x) - f(\bar{x}) \ \ \forall x \in \mathbb{R}^n,$$

which implies $0 \in \partial f(\bar{x})$. by the definition of sub-differential.
The sufficient condition follows by a similar argument.let $0 \in \partial f(\bar{x})$. then by sub-differential rule

$$f(x) \geq f(\bar{x}) + \langle 0, x - \bar{x} \rangle \ \ \forall x \in \mathbb{R}^n,$$

this inequality implies $f(\bar{x}) \leq f(x) \ \ \forall x \in \mathbb{R}^n$. □

Definition 1.7.11. *The intersection of any collection of affine sets is an affine set and the affine hull of* Ω *is defined by*

$$aff(\Omega) := \bigcap \{C \mid C \text{is affine and}, \Omega \subset C\}.$$

Definition 1.7.12. *We say that* $x \in ri(\Omega)$ *if there exists* $\delta > 0$ *such that*

$$\mathbb{B}(x, \delta) \cap aff(\Omega) \subset \Omega,$$

where $\mathbb{B}(x, \delta)$ *denotes the closed ball centered at* x *with radius* δ.

Definition 1.7.13. *Let* Ω *be a nonempty convex subset of* $\mathbb{R}^n$. *Then the normal cone to the set*Ω *at* $\bar{x} \in \Omega$ *is defined by*

$$N(\bar{x}, \Omega) := \{v \in \mathbb{R}^n \mid \langle v, x - \bar{x} \rangle \leq 0, \forall x \in \Omega\}.$$

In the case where $\bar{x} \notin \Omega$ *we define* $N(\bar{x}, \Omega) := \emptyset$.

Let $\delta_\Omega(x)$ denote the indicator function of set Ω at x, i.e.,

$$\delta_\Omega(x) := \begin{cases} 0 & \text{if } x \in \Omega, \\ +\infty & \text{if } x \notin \Omega. \end{cases}$$

Then by definition, it is easy to see that

$$\partial \delta_\Omega(\bar{x}) = N(\bar{x}, \Omega).$$

Proposition 1.7.9. *Let* $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ *be a convex function, and let* $\bar{x} \in dom(f)$. *Then we have*

$$\partial f(\bar{x}) = \{v \in \mathbb{R}^n \mid (v, -1) \in N((\bar{x}, f(\bar{x})); epi(f))\}.$$

Proof. Fix any sub-gradient $v \in \partial f(\bar{x})$ then from Definition 1.7.10 that

$$\langle v, x - \bar{x} \rangle \leq f(x) - f(\bar{x}), \forall x \in \mathbb{R}^n. \tag{1.1}$$

To show that $(v, -1) \in N((\bar{x}, f(\bar{x})); epi(f))$, fix any $(x, \lambda) \in epi(f)$ and observe that due to $\lambda \geq f(x)$ we have the relationships,

$$\begin{aligned} \langle (v, -1), (x, \lambda) - (\bar{x}, f(\bar{x})) \rangle &= \langle v, x - \bar{x} \rangle + (-1)(\lambda - f(\bar{x})) \\ &= \langle v, x - \bar{x} \rangle - (\lambda - f(\bar{x})) \\ &\leq \langle v, x - \bar{x} \rangle - (f(x) - f(\bar{x})) \leq 0. \end{aligned}$$

where the last inequality holds by(1.1). To verify the opposite inclusion in, take $(v,-1) \in N((\bar{x},f(\bar{x}));epi(f))$ and fix any $x \in dom(f)$. Then $(x,f(x)) \in epi(f)$ and hence,

$$\langle (v,-1),(x,f(x))-(\bar{x},f(\bar{x}))\rangle \leq 0,$$

which in turn implies the inequality

$$\langle v,x-\bar{x}\rangle \leq (f(x)-f(\bar{x})) \leq 0.$$

Thus $v \in \partial f(\bar{x})$, which completes the proof. □

Theorem 1.7.3. *(B. Mordukhovich & Nam, 2014) Let $f_i : \mathbb{R}^n \to \bar{\mathbb{R}}$ be proper extended real-valued convex functions and the intersection of relative interior of domain $\bigcap_{i=1}^m ri(dom(f_i)) \neq \emptyset$ where $m \geq 2$. Then for all $\bar{x} \in \bigcap_{i=1}^m dom(f_i)$,*

$$\partial\left(\sum_{i=1}^m f_i(\bar{x})\right) = \sum_{i=1}^m \partial f_i(\bar{x}). \tag{1.2}$$

Proof. Consider first the case of $m=2$ and Let $v_1 \in \partial f_1(\bar{x})$ and $v_2 \in \partial f_2(\bar{x})$. By the definition of sub-gradient,

$$f_1(x)-f_1(\bar{x}) \geq \langle v_1,x-\bar{x}\rangle, \forall x \in \mathbb{R}^n \text{and}, f_2(x)-f_2(\bar{x}) \geq \langle v_2,x-\bar{x}\rangle, \forall x \in \mathbb{R}^n.$$

By adding this two inequalities, one gets

$$(f_1+f_2)(x)-(f_1+f_2)(\bar{x}) \geq \langle v_1+v_2,x-\bar{x}\rangle, \forall x \in \mathbb{R}^n,$$

so $v_1+v_2 \in \partial(f_1+f_2)(\bar{x})$.
This implies, $\partial(f_1+f_2)(\bar{x}) \supset \partial f_1(\bar{x})+\partial f_2(\bar{x})$.
For the other direction, pick any $v \in \partial(f_1+f_2)(\bar{x})$. Then we have

$$\langle v,x-\bar{x}\rangle \leq (f_1+f_2)(x)-(f_1+f_2)(\bar{x}) \forall x \in \mathbb{R}^n. \tag{1.3}$$

Define the following convex subsets of $\mathbb{R}^{n+2}$ by

$$\Omega_1 := \{(x,\lambda_1,\lambda_2) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \mid \lambda_1 \geq f_1(x)\} = epi(f_1) \times \mathbb{R},$$
$$\Omega_2 := \{(x,\lambda_1,\lambda_2) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \mid \lambda_2 \geq f_2(x)\} = epi(f_2) \times \mathbb{R}.$$

We can easily verify by (1.3) and the normal cone definition that

$$(v,-1,-1) \in N((\bar{x},f_1(\bar{x}),f_2(\bar{x}));\Omega_1 \cap \Omega_2).$$

let us check that $ri(\Omega_1)\cap ri(\Omega_2)\neq\emptyset$. Indeed, we get

$$ri(\Omega_1)=\{(x,\lambda_1,\lambda_2)\in\mathbb{R}^n\times\mathbb{R}\times\mathbb{R}\,|\,x\in ri(domf_1),\lambda_1>f_1(x)\}=ri(epi(f_1))\times\mathbb{R},$$
$$ri(\Omega_2)=\{(x,\lambda_1,\lambda_2)\in\mathbb{R}^n\times\mathbb{R}\times\mathbb{R}\,|\,x\in ri(domf_2),\lambda_2>f_2(x)\}=ri(epi(f_2))\times\mathbb{R}.$$

Then choosing $z\in ri(dom(f_1))\cap ri(dom(f_2))$, it is not hard to see that

$$(z,f_1(z)+1,f_2(z)+1)\in ri(\Omega_1)\cap ri(\Omega_2)\neq\emptyset,$$

set intersection gives us

$$N((\bar{x},f_1(\bar{x}),f_2(\bar{x}));\Omega_1\cap\Omega_2)=N((\bar{x},f_1(\bar{x}),f_2(\bar{x}));\Omega_1)+N((\bar{x},f_1(\bar{x}),f_2(\bar{x}));\Omega_2).$$

It follows from the structures of the sets Ω_1 and Ω_2 that

$$(v,-1,-1)=(v_1,-\sigma_1,0)+(v_2,0,-\sigma_2)$$

with $(v_1,-\sigma_1)\in N((\bar{x},f_1(\bar{x}));epi(f_1))$ and $(v_2,\sigma_2)\in N((\bar{x},f_2(\bar{x}));epi(f_2))$. Thus

$$v=v_1+v_2,\sigma_1=\sigma_2=1,$$

and we have by Proposition 1.7.9 that $v_1\in\partial f_1(\bar{x})$ and $v_2\in\partial f_2(\bar{x})$. This ensures the inclusion $\partial(f_1+f_2)(\bar{x})\subset\partial f_1(\bar{x})+\partial f_2(\bar{x})$ and hence verifies (1.1) in the case of $m=2$.
To complete the proof of the theorem in the general case of $m>2$, is using induction. □

Lemma 1.7.3. *Let Ω be a convex set in $\mathbb{R}^n$. Then $int(\Omega)=ri(\Omega)$ provided that $int(\Omega)\neq\emptyset$. Furthermore, $N(\bar{x},\Omega)=\{0\}$ if $\bar{x}\in int(\Omega)$.*

Proof. Suppose that $int(\Omega)\neq\emptyset$ and check that $aff(\Omega)=\mathbb{R}^n$. Let picking $\bar{x}\in int(\Omega)$ and fixing $x\in\mathbb{R}^n$,find $t>0$ with $tx+(1-t)\bar{x}=\bar{x}+t(x-\bar{x})\in int(\Omega)\subset$ aff(Ω). It yields

$$x=\frac{1}{t}(tx+(1-t)\bar{x})+(1-\frac{1}{t})\bar{x}\in\text{aff}(\Omega),$$

which justifies the claimed statement due to the definition of relative interior.
To verify the second statement take $v\in N(\bar{x};\Omega)$ with $\bar{x}\in int(\Omega)$ and get

$$\langle v,x-\bar{x}\rangle\leq 0,\forall x\in\Omega.$$

Choosing $\delta>0$ such that $\bar{x}+tv\in\mathbb{B}(\bar{x},\delta)\subset\Omega$ for $t>0$ sufficiently small, gives us

$$\langle v,\bar{x}+tv-\bar{x}\rangle=t\|v\|^2\leq 0,$$

which implies $v=0$ and thus completes the proof. □

1.7.4 On DC Programming

Convexity is a pleasant feature of functions, however under basic algebraic operations like minimum or scalar multiplication, it is not retained. This serves as inspiration to look for novel optimization techniques that can handle larger classes of functions and sets for which convexity is not required. Thinking about the class of functions representable as differences of two convex functions is one of the most effective ways to get beyond convexity. Such functions are referred to as DC functions, where DC stands for difference of convex. It was first considered by Alexandrov (1940, 1950) and Landis (1951), and some time later by (Hartman, 1959) who recognized that the class of DC functions has many nice algebraic properties; see, (Hartman, 1959).

This wide range of applications is to be expected, since some important classes of non-convex functions may be represented as DC functions. For instance, twice continuously differentiable functions on any convex subset of $\mathbb{R}^n$ (Hartman, 1959) and continuous piecewise linear functions (Abbaszadehpeivasti et al., 2024; Melzer, 1986) may be written as DC functions. Furthermore, every continuous function on a compact and convex set can be approximated by a DC function (Horst & Thoai, 1999; Tuy et al., 1998). We refer the interested reader to (J.-B. Hiriart-Urruty, 1985) and (Tuy et al., 1998) for more information on DC representable functions. In this subsection we introduce DC functions and a number of important properties of this class under operations which are commonly considered in non-convex optimization problems. More details can be found in (Horst, 2000; B. S. Mordukhovich & Nam, 2022; Tuy & Tuy, 2016) in the proofs and other properties. We first begin with the definition of DC functions.

Definition 1.7.14. *A convex function $f(x) : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ is proper if $f(x) < +\infty$ for at least one x and $f(x) > -\infty, \forall x \in \mathbb{R}^n$.*

Definition 1.7.15. *An extended real valued function $f : \mathbb{R}^n \to (-\infty, \infty]$, is called a DC function, if it can be represented as a difference of two convex function g and h as,*

$$\textit{Minimize} \quad f(x) := g(x) - h(x), \; x \in \mathbb{R}^n, \tag{1.4}$$

where $g : \mathbb{R}^n \longrightarrow (-\infty, \infty]$ is a proper closed (lower semi-continuous) convex function and $h : \mathbb{R}^n \longrightarrow \mathbb{R}$ is convex. If $\Omega \subset \mathbb{R}^n$ be nonempty closed and convex set than a real-valued function $f : \Omega \longrightarrow \mathbb{R}$ is called DC on Ω.

The convex constrained DC program

$$\inf\{f(x) = g(x) - h(x) : x \in \Omega\}. \tag{1.5}$$

can be formulated as a standard unconstrained DC program as:

$$\min\{(g + \delta_\Omega)(x) - h(x) : x \in \mathbb{R}^n\}, \tag{1.6}$$

where δ_Ω is an indicator function of Ω. Clearly since δ_Ω is convex, both $g+\delta_\Omega$ and h are convex functions.

Proposition 1.7.10. *Under the convention $\infty-\infty=\infty$ and the assumption that the solution set of (1.4) is non-empty (implying that $\varnothing \neq dom(g) \subset dom(h)$), if $\bar{x}$ is a local minimizer of $f(x)$, then $\partial h(\bar{x}) \subset \partial g(\bar{x})$.*

Proof. Fix any $v \in \partial h(\bar{x})$. Linearize the concave part of $f(x)$ we have,

$$\psi(x) = g(x) - h(\bar{x}) - \langle v, x-\bar{x}\rangle. \tag{1.7}$$

Point $\bar{x}$ is a globally optimal solution of the convex problem (1.7). Applying standard necessary and sufficient optimality conditions one obtains that $0 \in \partial\psi(\bar{x})$. Which implies that $0 \in \partial g(\bar{x}) - v$, one gets that $v \in \partial g(\bar{x})$,as v is arbitrary implies the desired result. □

Any point $\bar{x}$ that satisfies the condition $\partial h(\bar{x}) \subset \partial g(\bar{x})$ is called a stationary point of (1.4) and any point $\bar{x}$ such that $\partial h(\bar{x}) \cap \partial g(\bar{x}) \neq \varnothing$ is called a critical point of (1.4). If $h(x)$ is further differentiable, the stationary points and the critical points coincide. A DC optimization problem, and it can be solved by *the Difference of Convex Algorithm* introduced by (T. P. An L.T.H., 1997) and (Tao & An, 1998). a DC optimization problem, and it can be solved by DCA or Algorithm 1.

Algorithm 1: DCA algorithm 1

```
Input  : x_0 ∈ ℝ^n, N ∈ ℕ,
while stopping is not reached do
    for k = 1, ..., N do
        Find y_k ∈ ∂h(x_{k-1}),
        Find x_k ∈ ∂g*(y_k),
    end
end
Output: x_N.
```

The function g^* referred in DCA is the Fenchel Conjugate of g, and it is defined as:

$$g^*(y) = \sup\{\langle y,x\rangle - g(x) \mid x \in \mathbb{R}^n\}, y \in \mathbb{R}^n. \tag{1.8}$$

Conjugate functions are always closed and convex (regardless of the properties of g).
Note that in the case where g is a proper function, i.e.,

$$\text{dom}(g) := \{x \in \mathbb{R}^n \mid g(x) < \infty\} \neq \emptyset,$$

the Fenchel conjugate $g^* : \mathbb{R}^n \to \bar{\mathbb{R}}$ is always convex and is an extended-real-valued convex function. Suppose further that g is convex and lower semi-continuous, then the Fenchel-

Moreau Theorem states that $(g^*)^* = g$, see, (Zalinescu, 2002). Based on this theorem,we have the following relation between the sub-gradients of g and its Fenchel conjugate g^*,

$$x \in g^*(y) \Leftrightarrow y \in g(x).$$

Theorem 1.7.4. *Rockafellar (1970) Let $g : \mathbb{R}^n \to (-\infty, \infty]$ be convex and fix $x \in \mathbb{R}^n$. Then the following statements are equivalent:*

1. $\mathbf{x} \in \partial g^*(\mathbf{y})$.
2. $\mathbf{y} \in \partial g(\mathbf{x})$.
3. $g(\mathbf{x}) + g^*(\mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$.

Proof. 1. Suppose g is convex and closed then from $g(\mathbf{x}) + g^*(\mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$ we have,

$$\begin{aligned} \langle \mathbf{x}, \mathbf{y} \rangle - g(x) &= \sup \{ \langle \mathbf{y}, \mathbf{z} \rangle - g(z) \}, \forall z \in \mathbb{R} \\ \langle \mathbf{x}, \mathbf{y} \rangle - g(x) &= \langle \mathbf{y}, \mathbf{z} \rangle - g(z), \forall z \in \mathbb{R} \\ g(z) &\geq g(x) + y^T (z - x), \forall z \in \mathbb{R} \end{aligned}$$

this implies $y \in \partial g(\mathbf{x})$.

2. Suppose $y \in \partial g(\mathbf{x})$ then,

$$\begin{aligned} g^*(y) &= \sup_u \{ y^T u - g(u) \} \\ &= y^T x - g(x) \\ g^*(v) &= \sup_v \{ y^T v - g(v) \} \\ &\geq v^T x - g(x). \end{aligned}$$

Adding zero to the right hand side we have

$$\begin{aligned} &= v^T x - y^T x - g(x) + y^T x \\ &= x^T (v - y) + y^T x - g(x) \\ &= x^T (v - y) + g^*(y) \\ g^*(v) &\geq g^*(y) + x^T (v - y) \\ \text{this implies } &x \in \partial g^*(\mathbf{y}). \end{aligned}$$

□

Proposition 1.7.11. *Given a convex function $g : \mathbb{R}^n \to \mathbb{R}$. Then $g^* : \mathbb{R}^n \to \bar{\mathbb{R}}$ is also a convex function and lower semi-continuous.*

Proof. (i) Let be $y_1, y_2 \in \mathbb{R}^n$ and $\lambda \in [0,1]$. Concerning the convexity of g^* we have,

$$\begin{aligned} g^*(\lambda y_1 + (1-\lambda)y_2) &= \sup_{x\in\mathbb{R}^n}\{\langle \lambda y_1 + (1-\lambda)y_2, x\rangle - g(x)\} \\ &\leq \lambda \sup_{x\in\mathbb{R}^n}\{\langle y_1, x\rangle - g(x)\} + (1-\lambda)\sup_{x\in\mathbb{R}^n}\{\langle y_2, x\rangle - g(x)\} \\ &= \lambda g^*(y_1) + (1-\lambda)g^*(y_2), \end{aligned}$$

this implies that $g^* : \mathbb{R}^n \to \bar{\mathbb{R}}$ is also a convex function.
(ii) Let $y_n \to y \in \mathbb{R}^n$. Applying Young's inequality, we obtain

$$\begin{aligned} g^*(y_n) &= \sup\{\langle y_n, x\rangle - g(x)\}, \text{for all } \ x \in \mathbb{R}^n. \\ &\leq \langle y_n, x\rangle - g(x), \text{for all } \ x \in \mathbb{R}^n. \end{aligned}$$

This follows

$$\begin{aligned} \liminf_{n\to\infty} g^*(y_n) &\geq \liminf_{n\to\infty}\{\langle y_n, x\rangle - g(x)\} \\ &= \langle y, x\rangle - g(x), \text{for all } \ x \in \mathbb{R}^n. \end{aligned}$$

Hence,

$$\liminf_{n\to\infty} g^*(y_n) \geq \sup\{\langle y, x\rangle - g(x)\} = g^*(y)$$

therefore, g is lower semi-continuous. □

Remark 1.7.3. *If $g : \mathbb{R}^n \to \bar{\mathbb{R}}$ a proper convex and lower semi-continuous function. Then g^* is a proper functional.*
Because g is proper, it follows the existence of an $x \in \mathbb{R}^n$ such that $g(x) < \infty$ and so

$$g^*(y) = \sup_{x_1\in\mathbb{R}^n}\{\langle y, x_1\rangle - g(x_1)\} \geq \langle y, x\rangle - g(x) > -\infty, \textit{for all } \ y \in \mathbb{R}^n.$$

Proposition 1.7.12. *Let $g : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper lower semi-continuous convex function.Then $\partial g(\mathbb{R}^n) := \bigcup_{x\in\mathbb{R}^n} \partial g(x) = dom \ \partial(g^*) := \{y \in \mathbb{R}^n \mid \partial g^*(y) \neq \emptyset\}$.*

Proof. Let $x \in \mathbb{R}^n$ and $y \in \partial g(x)$. Then $x \in \partial g^*(y)$ which implies $\partial g^*(y) \neq \emptyset$, and so $y \in$ dom∂g^*. □

Example 1.7.3. *Minimize $f(x) = x^4 - 3x^2 - x, x \in \mathbb{R}$ then $f(x) = g(x) - h(x)$, where $g(x) = x^4$*

and $h(x) = 3x^2 + x$. *Applying DCA we have*

$$\begin{aligned} \partial h(x) &= \{6x+1\}, \partial g^*(y) \\ &= \{argmin_{x\in\mathbb{R}^n}(x^4 - yx)\} \\ &= \left\{\sqrt[3]{\frac{y}{4}}\right\}, x_{k+1} = \sqrt[3]{\frac{6x_k+1}{4}}. \end{aligned}$$

solution using Algorithm 1 is in Figure 1.1.

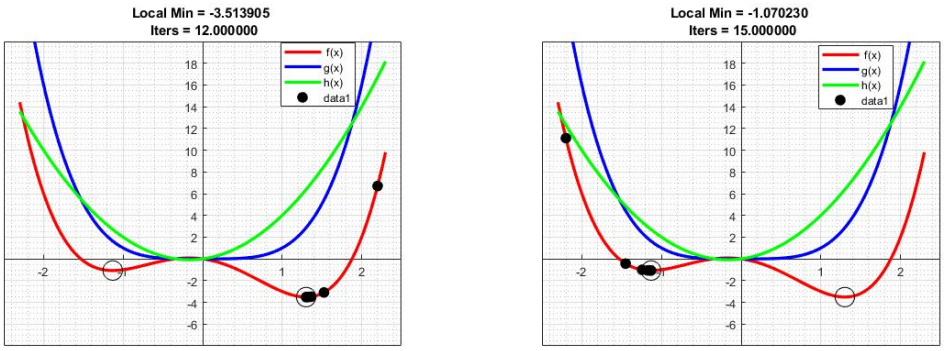


Figure 1.1: DCA algorithm depends on initial values

Definition 1.7.16. *We say that a function* $g : \mathbb{R}^n \to \bar{\mathbb{R}}$ *is coercive if*

$$\lim_{||x||\to\infty} \frac{g(x)}{||x||} = \infty.$$

We also say that g is level-bounded if for any $\alpha \in \mathbb{R}$, the level set $g^{-1}((-\infty, \alpha])$ is bounded. That is if g is a coercive function, then the sub-level $L_\alpha := \{x \in \mathbb{R}^n : g(x) \leq \alpha$ is bounded and for every $\alpha \in \mathbb{R}$. Indeed, we notice that g is coercive which implies $\forall\, \alpha \in \mathbb{R}\ \exists r > 0 : g(x) \geq \alpha$ for any $x \in \{x \in \mathbb{R}^n : ||x|| \geq r\}$ and, in particular, we have $L_\alpha \subset \mathbb{B}(0, r)$ for some $r > 0$.
In conclusion, if Ω is closed and g is continuous (or lower semi-continuous), then we simply need to solve the minimization problem,

$$\min\{g(x) : x \in \Omega \cap L_\alpha\},$$

which admits a solution by Weierstrass theorem, as the intersection $\Omega \cap L_\alpha$ is compact.

Proposition 1.7.13. *Let* $g : \mathbb{R}^n \to \bar{\mathbb{R}}$ *be a proper lower semi-continuous convex function. Suppose that* g *is coercive and level-bounded. Then* $dom(\partial g^*) = \mathbb{R}^n$. *In particular,*$dom(g^*) = \mathbb{R}^n$.

Proof. We want to show that the set $\partial g(\mathbb{R})$ is closed. Fix any sequence $v_k \in \partial g(\mathbb{R})$ that converges to v. For each $k \in N$, choose $x_k \in \mathbb{R}^n$ such that $v_k \in \partial g(x_k)$. Thus,

$$\langle v_k, x - x_k \rangle \leq g(x) - g(x_k), \forall x \in \mathbb{R}. \tag{1.9}$$

This implies

$$g(x_k) - \langle v_k, x_k \rangle \leq g(x) - \langle v_k, x \rangle, \forall x \in \mathbb{R}.$$

In particular, we can fix $\bar{x} \in \text{dom} g$ and use the fact that v_k is bounded to find a constant $l_0 \in \mathbb{R}$ such that

$$g(x_k) - \langle v_k, x_k \rangle \leq g(\bar{x}) - \langle v_k, \bar{x} \rangle \leq l_0, \forall k \in \mathbb{N}. \tag{1.10}$$

Let us now show that x_k is bounded. By contradiction, assume that this is not the case. Without loss of generality, we can assume that $\lim_{k \to \infty} ||x_k|| = \infty$. By the coercive property of g,

$$\lim_{||x|| \to \infty} \frac{g(x_k) - \langle v_k, x_k \rangle}{||x_k||} = \infty.$$

This is a contradiction to (1.10), so x_k is bounded. We can assume without loss of generality that x_k converges to $a \in \mathbb{R}^n$. Then it follows from (1.9) by passing the limit that

$$\langle v, x - a \rangle \leq g(x) - g(a), \forall x \in \mathbb{R}.$$

This implies $v \in \partial g(a) \subset \partial g(\mathbb{R}^n)$, and hence $\partial g(\mathbb{R}^n)$ is closed. By Proposition 1.7.12,

$$\mathbb{R}^n = \partial g(\mathbb{R}^n) = \text{dom} \partial (g^*),$$

which completes the proof. □

Based on the proposition below, we see that in the case where we cannot find x_k or y_k exactly for Algorithm 1, we can find them approximately by solving a convex optimization problem.

Proposition 1.7.14. *Let $g, h : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a proper lower semi-continuous convex function. Then $v \in \partial g^*(y)$ if and only if*

$$v \in \textit{argmin} \{ g(x) - \langle y, x \rangle \mid x \in \mathbb{R}^n \}. \tag{1.11}$$

Moreover, $w \in \partial h(x)$ if and only if

$$w \in \textit{argmin} \{ h^*(x) - \langle y, x \rangle \mid y \in \mathbb{R}^n \}. \tag{1.12}$$

Proof. Suppose that 1.11. Let $\varphi(x) := g(x) - \langle y, x\rangle, x \in \mathbb{R}^n$.
Then $0 \in \partial\varphi(v)$, implies

$$0 \in \partial g(v) - y,$$

$$y \in \partial g(v).$$

Suppose that 1.12. Then $0 \in \partial\psi(w)$, where $\psi(y) := h^*(y) - \langle x, y\rangle, y \in \mathbb{R}^n$. This implies $0 \in \partial h^*(w) - x$, $x \in \partial h^*(w)$, or, $x \in \partial h^*(w), \Leftrightarrow w \in \partial h(x)$. □

Based on Proposition 1.7.14, we derive an alternative version of the following DCA Algorithm 2.

Algorithm 2: DCA algorithm 2

Input : $x_0 \in \mathbb{R}^n, N \in \mathbb{N}$,
while *stopping is not reached* **do**
  for $i = 1, ..., N$ **do**
    Find $y_k \in \partial h(x_{k-1})$ or find y_k approximately by solving the problem,

$$\text{minimize} \quad \psi_k(y) := h^*(y) - \langle y, x_{k-1}\rangle, y \in \mathbb{R}^n,$$

    Find $x_k \in \partial g^*(y_k)$ or find x_k approximately by solving the problem,

$$\text{minimize} \quad \phi_k(x) := g(x) - \langle x, y_k\rangle, x \in \mathbb{R}^n.$$

  end
end
Output: x_N.

Definition 1.7.17. *A function $h : \mathbb{R}^n \to \mathbb{R}$ is called ρ-convex ($\rho \geq 0$) if the function defined by $k(x) := h(x) - \frac{\rho}{2}||x||^2, x \in \mathbb{R}^n$, is convex. If there exists $\rho > 0$ such that h is ρ-convex, then h is called strongly convex.*

Lemma 1.7.4. *Let $A \in \mathbb{R}^{m\times n}$ and given a nonempty closed bounded convex subset $F \subset \mathbb{R}^m$. Suppose that $\psi : \mathbb{R}^m \to \mathbb{R}$ is a strongly convex function with parameter $\rho > 0$. Then the function $f : \mathbb{R}^m \to \mathbb{R}$ defined by*

$$f(x) := \max\{\langle Ax, u\rangle - \psi(u) \mid u \in F\},$$

is well-defined and Frechet differentiable with Lipschitz gradient with constant $\ell = \frac{||A||^2}{\rho}$.

Proof. Bajaj et al. (2022) The sub-differential of the function f is given by

$$\partial f(x) = \left\{A^T u(x)\right\} \quad \forall x \in \mathbb{R}^n,$$

where $u(x) \in R^m$ denotes the unique element such that the maximum is attained in the definition of $f(x)$. Let $g_x(u) = -\langle Ax, u\rangle + \psi(u) + \delta(x, F)$. Then

$$0 \in \partial g_x(u(x)) = -Ax + \partial g(u(x)),$$

where $g(x) := h(x) + \delta(x; F)$ is strongly convex with parameter ρ. It follows that

$$Ax \in \partial g(u(x)). \tag{1.13}$$

The function g is strongly convex with parameter ρ, so its sub-differential is strongly monotone in the sense that

$$\rho||x_1 - x_2||^2 \leq \langle w_1 - w_2, x_1 - x_2\rangle \quad \text{whenever } w_i \in \partial g(x_i),$$

then using Eq.(2.6)

$$\begin{aligned} \rho||u(x_1) - u(x_2)||^2 &\leq \langle A(x_1) - A(x_2), u(x_1) - u(x_2)\rangle \\ &= \langle x_1 - x_2, A^T(u(x_1)) - A^T(u(x_2))\rangle \\ &\leq ||x_1 - x_2||||A^T(u(x1)) - A^T(u(x2))||, \end{aligned}$$

It implies that

$$\begin{aligned} ||A^T(u(x_1)) - A^T(u(x^2))||^2 &\leq ||A^T||^2||u(x_1) - u(x_2)||^2 \\ &\leq \frac{||A||^2}{\rho}||x_1 - x_2||||A^T(u(x_1)) - A^T(u(x_2))||. \end{aligned}$$

This implies

$$||A^T(u(x_1)) - A^T(u(x_2))|| \leq \frac{||A||^2}{\rho}||x_1 - x_2||.$$

Thus, the sub-differential mapping $\partial\rho(.)$ is continuous, and so f is Fre'chet differentiable with $\partial f(x) = \{\nabla f(x)\} = \{A^T u(x)\}$. In addition,

$$||\nabla\rho(x_1) - \nabla\rho(x_2)|| \leq \frac{||A||^2}{\rho}||x_1 - x_2||.$$

Which Lipschitz continuous and complete the proof. □

Note that any $C^{1,1}$ function is also C^1 (a continuously differentiable function). Following the definition of a $C^{1,1}$ function, we shall provide relevant propositions.

Proposition 1.7.15. *Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a $C^{1,1}$ function (not necessarily convex) with Lips-*

chitz continuous gradient and Lipschitz constant ℓ. Then,

$$f(y) \leq f(x) + \langle \nabla f(x), y-x \rangle + \frac{\ell}{2}||x-y||^2, \textit{for all } x,y \in \mathbb{R}^n.$$

Proof. For all $x,y \in \mathbb{R}^n$, define $\varphi(t) := f(x+t(y-x)), t \in \mathbb{R}$. Then φ is differentiable on $\mathbb{R}$. Observe that

$$\varphi(0) = f(x), \varphi(1) = f(y), \varphi'(t) = \langle \nabla f(x+t(y-x)), y-x \rangle.$$

By the Fundamental Theorem of Calculus we have

$$\varphi(1) - \varphi(0) = \int_0^1 \varphi'(t)dt = \int_0^1 \langle \nabla f(x+t(y-x)), y-x \rangle dt.$$

Hence

$$\begin{aligned} f(y) - f(x) &= \int_0^1 \langle \nabla f(x+t(y-x)) - \nabla f(x) + \nabla f(x), y-x \rangle dt \\ &= \int_0^1 \langle \nabla f(x+t(y-x)) - \nabla f(x), y-x \rangle dt + \langle \nabla f(x), y-x \rangle. \end{aligned}$$

Rearranging the above equation by moving $f(x)$ to the right-hand side gives us

$$\begin{aligned} f(y) &= f(x) + \langle \nabla f(x), y-x \rangle + \int_0^1 \langle \nabla f(x+t(y-x)) - \nabla f(x), y-x \rangle dt \\ &\leq f(x) + \langle \nabla f(x), y-x \rangle + \int_0^1 ||\nabla f(x+t(y-x)) - \nabla f(x)||.||y-x|| dt \\ &\leq f(x) + \langle \nabla f(x), y-x \rangle + \int_0^1 \ell t ||y-x||^2 dt \\ &\leq f(x) + \langle \nabla f(x), y-x \rangle + \frac{\ell}{2}||y-x||^2. \end{aligned}$$

Which completes the proof. □

As mentioned earlier, any $C^{1,1}$ function is also C^1. We provide the following relevant propositions associated with C^1 functions.

Proposition 1.7.16. *Suppose that $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is a C^1 function. Then f is convex if and only if*

$$\langle \nabla f(y), x-y \rangle \leq f(x) - f(y), \textit{for all } x,y \in \mathbb{R}^n.$$

Proof. Fix any $x,y \in \mathbb{R}^n$ and $t \in (0,1)$. Then we have

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y),$$

which is equivalent to

$$f(y+t(x-y)) \leq f(y)+t[f(x)-f(y)],$$

Thus,

$$\frac{f(y+t(x-y))-f(y)}{t} \leq f(x)-f(y)$$

Letting $t \longrightarrow 0^+$, gives us

$$\langle \nabla f(y), x-y \rangle \leq f(x)-f(y).$$

Conversely, for any $x, y \in \mathbb{R}^n$ and $t \in (0,1)$, define $z_t := tx+(1-t)y$. Then we will get that

$$\langle \nabla f(z_t), x-z_t \rangle \leq f(x)-f(z_t) \text{ and } \langle \nabla f(z_t), y-z_t \rangle \leq f(y)-f(z_t).$$

It follows that

$$t\langle \nabla f(z_t), x-z_t \rangle \leq tf(x)-tf(z_t),$$

$$(1-t)\langle \nabla f(z_t), y-z_t \rangle \leq (1-t)f(y)-(1-t)f(z_t).$$

Adding these inequalities gives us

$$0 \leq tf(x)+(1-t)f(y)-f(z_t).$$

Therefore, $f(z_t) \leq tf(x)+(1-t)f(y)$, and thus f is convex. □

Corollary 1.7.1. *Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a C^1 function. Then f is convex if and only if it is monotone in the sense that*

$$\langle \nabla f(x)-\nabla f(y), x-y \rangle \geq 0 \text{ for all } x, y \in \mathbb{R}^n.$$

Proof. Suppose that f is convex. Let us show that ∇f is monotone. For all $x, y \in \mathbb{R}^n$ we have

$$\langle \nabla f(x), y-x \rangle \leq f(y)-f(x),$$

$$\langle \nabla f(y), x-y \rangle \leq f(x)-f(y).$$

Adding these inequalities, we get

$$\langle \nabla f(x)-\nabla f(y), x-y \rangle \geq 0.$$

Thus, ∇f is monotone. To prove the converse, fix any $x, y \in \mathbb{R}^n$ and let $z_t := tx+(1-t)y$.

Using a similar technique as in the proof of Proposition 1.7.15, we have

$$\begin{aligned} f(x)-f(y) &= \int_0^1 \langle \nabla f(z_t)-\nabla f(y)+\nabla f(y), x-y\rangle dt \\ &= \int_0^1 \langle \nabla f(z_t)-\nabla f(y), x-y\rangle dt + \langle \nabla f(y), x-y\rangle. \end{aligned}$$

We next prove that $\langle \nabla f(z_t)-\nabla f(y), x-y\rangle \geq 0$ for all $t \in [0,1]$. If $t=0$, then $z_t = y$, and hence

$$\langle \nabla f(z_t)-\nabla f(y), x-y\rangle = 0.$$

If $0 < t \leq 1$, then $z_t - y = t(x-y)$ and $x-y = \frac{1}{t}(z_t-y)$. By the monotonicity of ∇f, we have

$$\langle \nabla f(z_t)-\nabla f(y), x-y\rangle = \frac{1}{t}\langle \nabla f(z_t)-\nabla f(y), z_t-y\rangle \geq 0.$$

Thus, $f(x)-f(y) \geq \langle \nabla f(y), x-y\rangle$ and Proposition 1.7.16 tells us that f is a convex function on $\mathbb{R}^n$. □

We now conclude this subsection with a proposition, linking a C^1 function with the previous propositions and corollary.

Proposition 1.7.17. *Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a C^1 function. The following assertions are equivalent:*

1. *f is strongly convex with the parameter ρ.*
2. *For all $x, y \in \mathbb{R}^n$, we have the inequality*

$$\langle \nabla f(y), x-y\rangle \leq f(x)-f(y)-\frac{\rho}{2}||x-y||^2.$$

3. *The gradient of the function f is strongly monotone in the sense that*

$$\langle \nabla f(x)-\nabla f(y), x-y\rangle \geq ||x-y||^2 \text{ for all } x, y \in \mathbb{R}^n.$$

Proposition 1.7.18. *Let $h : \mathbb{R}^n \to \bar{\mathbb{R}}$ be $\rho-$convex with $\bar{x} \in domh$. Then $v \in \partial h(\bar{x})$ if and only if*

$$\langle v, x-\bar{x}\rangle + \frac{\rho}{2}||x-\bar{x}|| \leq h(x)-h(\bar{x}). \tag{1.14}$$

Proof. Let $k : \mathbb{R}^n \to \bar{\mathbb{R}}$ be the convex function with $k(x) = h(x) - \frac{\rho}{2}||x||^2$. For $v \in \partial h(\bar{x})$, one has $v \in \partial \varphi(\bar{x})$, where $\varphi(x) = k(x) + \frac{\rho}{2}||x||^2$ for $x \in \mathbb{R}^n$. By the sub-differential sum rule,

$$v \in \partial k(\bar{x}) + \rho\bar{x},$$

which implies $v-\rho\bar{x} \in \partial k(\bar{x})$. Then

$$\langle v-\rho\bar{x}, x-\bar{x}\rangle \leq k(x)-k(\bar{x}), \forall x \in \mathbb{R}^n.$$

It follows that

$$\begin{aligned}
\langle v, x-\bar{x}\rangle &\leq \rho\langle \bar{x}, x\rangle - \rho\langle \bar{x}, \bar{x}\rangle + h(x) - \frac{\rho}{2}||x||^2 - (h(\bar{x}) - \frac{\rho}{2}||x||^2) \\
&\leq h(x) - h(\bar{x}) - \frac{\rho}{2}(||x||^2 - 2\langle x, \bar{x}\rangle + ||\bar{x}||^2) \\
&= h(x) - h(\bar{x}) - \frac{\rho}{2}||x-\bar{x}||^2.
\end{aligned}$$

This implies 1.14 and completes the proof. □

Theorem 1.7.5. *(Nam et al., 2017) Let f be as defined in problem (1.4) and let $\{x_k\}$ be a sequence generated by the DCA. Suppose that g and h are ρ_1 and ρ_2 convex respectively, then at every iteration number k of the DCA, we have*

$$f(x_{k+1}) \leq f(x_k) - \frac{\rho_1+\rho_2}{2}||x_{k+1}-x_k||^2, \forall k \in N. \tag{1.15}$$

Proof. Since $y_k \in \partial h(x_k)$, then one has

$$\langle y_k, x-x_k\rangle + \frac{\rho_2}{2}||x-x_k||^2 \leq h(x) - h(x_k) \text{for all} x \in \mathbb{R}^n.$$

In particular, h

$$\langle y_k, x_{k+1}-x_k\rangle + \frac{\rho_2}{2}||x_{k+1}-x_k||^2 \leq h(x_{k+1}) - h(x_k).$$

In addition,$x_{k+1} \in \partial g^*(y_k)$, and so $y_k \in \partial g(x_{k+1})$, which similarly implies

$$\langle y_k, x_k-x_{k+1}\rangle + \frac{\rho_1}{2}||x_k-x_{k+1}||^2 \leq g(x_k) - g(x_{k+1}).$$

Adding these inequalities gives (1.15).

$$f(x_{k+1}) \leq f(x_k) - \frac{\rho_1+\rho_2}{2}||x_{k+1}-x_k||^2, \forall k \in N.$$

□

Theorem 1.7.6. *(Nam et al., 2017) Let f be defined by (1.4) and $\{x_k\}$ is a sequence created by DCA. Then functional value $\{f(x_k)\}$ is a decreasing sequence. If f is bounded from below and g is lower semi-continuous, and that g is ρ_1-convex and h is ρ_2- convex with $\rho_1+\rho_2 > 0$. If $\{x_k\}$ is bounded, then all sub-sequential limits of $\{x_k\}$ converge to a stationary point of f.*

Lemma 1.7.5. *Suppose that $h : \mathbb{R}^n \to \mathbb{R}$ is a convex function. If $w_k \in \partial h(x_k)$ and $\{x_k\}$ is a bounded sequence, then $\{w_k\}$ is also bounded.*

Proof. Fix any point $\bar{x} \in \mathbb{R}^n$. By theorem 1.7.1 h is locally Lipschitz continuous around $\bar{x}$ then there exist $\ell > 0$ and $\delta > 0$ such that

$$|h(x) - h(y)| \leq \ell ||x - y||, \text{whenever } \; x, y \in \mathbb{B}(\bar{x}; \delta).$$

This implies that $||w|| \leq \ell$ whenever $w \in \partial h(u)$ for $u \in \mathbb{B}(\bar{x}; \frac{\delta}{2})$. Indeed,

$$\langle w, x - u \rangle \leq h(x) - h(u), \text{for all } \; x \in \mathbb{R}^n.$$

Choose $\rho > 0$ sufficiently small such that $\mathbb{B}(u; \rho) \subset \mathbb{B}(\bar{x}; \delta)$. Then
$\langle w, x - u \rangle \leq h(x) - h(u) \leq \ell ||x - u||$ whenever $||x - u|| \leq \rho$.
Thus, $||w|| \leq \ell$. For a contradiction, suppose now that $\{w_k\}$ is not bounded. Then we can assume without loss of generality that $||w_k|| \to \infty$. Since $\{x_k\}$ is bounded, it has a subsequence $\{x_{kp}\}$ that converges to $x_0 \in \mathbb{R}^n$. Let $\ell > 0$ be a Lipschitz constant of h around x_0. By the observation above, $||w_k|| \leq \ell$ for sufficiently large p. This is a contradiction. □

Definition 1.7.18. *We say that an element $\bar{x} \in \mathbb{R}^n$ is a stationary point of the function f defined by (1.3) if* $\partial g(\bar{x}) \bigcap \partial h(\bar{x}) \neq \emptyset$.

Obviously, in the case where both g and h are differentiable, $\bar{x}$ is a critical point of f if and only if $\bar{x}$ satisfies the Fermat rule $\nabla f(\bar{x} = 0$. The theorem below provides a convergence result for the DCA. It can be taken directly from Tao & An (1998).

Theorem 1.7.7. *Consider the function f defined by (1.3) and the sequence $\{x_k\}$ generated by the Algorithm 1. Then $\{f(x_k)\}$ is a decreasing sequence. Suppose further that f is bounded from below, that g is lower semi-continuous, and that g is ρ_1-convex and h is ρ_2-convex with $\rho_1 + \rho_2 > 0$. If $\{x_k\}$ is bounded, then every sub-sequential limit of the sequence $\{x_k\}$ is a stationary point of f.*

Proof. It follows from (1.15) that $\{f(x_k)\}$ is a decreasing sequence so it converges to real number since f is bounded from below. Then $f(x_k) - f(x_{k+1}) \to 0$ as $k \to \infty$, and so using (1.15) again yields $||x_{k+1} - x_k|| \to 0$. Suppose that $x_{k\ell} \to x^*$ as $\ell \to \infty$. By definition,

$$y_k \in \partial g(x_{k+1}) \text{for all } \; k \in \mathbb{N}.$$

Since $\{x_k\}$ is bounded, by Lemma 1.7.5, $\{y_k\}$ is also a bounded sequence. By extracting a further subsequence, we can assume without loss of generality that $y_{k-\ell} \to y^*$ as $\ell \to \infty$. Since $y_{k\ell} \in \partial h(x_{k\ell})$ for all $\ell \in \mathbb{N}$, one has

$$y^* \in \partial h(x^*).$$

Indeed, by the definition,

$$\langle y_{k_\ell}, x - x_{k_\ell} \rangle \leq h(x) - h(x_{k_\ell}) \text{for all} x \in \mathbb{R}^n, \ell \in \mathbb{N}. \tag{1.16}$$

Thus,

$$\langle y_{k_\ell}, x^* - x_{k_\ell} \rangle \leq h(x^*) - h(x_{k_\ell}).$$

Then $h(x_{k_\ell}) \leq \langle y_{k_\ell}, x_{k_\ell} - x^* \rangle + h(x^*)$, and hence limsup $h(x_{k_\ell}) \leq h(x^*)$. By the lower semi-continuity of h, one has that $h(x_{k_\ell}) \to h(x^*)$. Letting $l \to \infty$ in (1.15) gives $y^* \in \partial h(x^*)$. Since $||x_{k+1} - x_k|| \to 0$ and $x_{k_\ell} \to x^*$, one has $x_k \to x^*$. From the relation $y_{k_\ell} \in \partial g(x_{k_\ell} + 1)$, one has $y^* \in \partial g(x^*)$ by a similar argument. Therefore, x^* is a stationary point of f. □

Deeper studies of the convergence of this algorithm and its generalizations involving the Kurdyka-Lojasiewicz (KL) inequality are given in (L. An et al., 2018; N. An & Nam, 2017).

Lemma 1.7.6. *The sequence of $\{f(x^k)\}$ generated by DCA algorithm1 is non-increasing and convergent.*

Proof. Consider DC program in (1.4), let x^k be given and take $v^k \in \partial h(x^k)$. Then by convexity of h, the linearizion of the concave part is give as:

$$h(x^{k+1}) \geq h(x^k) + \langle v^k, x^{k+1} - x^k \rangle. \tag{1.17}$$

If x^k is a local minimizer of f , then it is a globally optimal solution of the convex equation in(1.18). Then solve,

$$x^{k+1} \in \arg\min_x \{g(x) - \langle v^k, x - x^k \rangle\}. \tag{1.18}$$

Then, for every $k \in \mathbb{N}$, we have

$$\begin{aligned} f(x^{k+1}) &= g(x^{k+1}) - h(x^{k+1}) \\ &\leq g(x^{k+1}) - h(x^k) - \langle v^k, x^{k+1} - x^k \rangle \text{ by majorization(1.17)} \\ &\leq g(x^k) - h(x^k) \text{ by minimization of (1.18)} \\ &= f(x^k). \end{aligned}$$

which implies that $\{f(x^k)\}$ is non-increasing. The non-emptiness of the solution set of (1.4) implies that f is lower bounded. Then the convergence of $\{f(x^k)\}$ is followed by the non-increasing and the lower boundedness of $\{f(x^k)\}$. □

Lemma 1.7.7. *(L. T. H. An & Tao, 2005; Niu, 2022) If either g or h is strongly convex over Ω (i.e., $\rho_g + \rho_h > 0$), then*

- *(sufficiently decrease property)*

$$f(x^k) - f(x^{k+1}) \geq \frac{\rho_g + \rho_h}{2}\|x^k - x^{k+1}\|^2, \forall k = 1,2,\ldots. \tag{1.19}$$

- *(square summable property)*

$$\sum_{k\geq 0} \|x^{k+1} - x^k\|^2 < \infty,$$

hence $\|x^{k+1} - x^k\| \to 0, k \to \infty$.

Proof. (Descent property): For every $k = 0,1,\ldots,$ by the first order optimality condition to the convex problem

$$x^{k+1} \in \arg\min\{g(x) - \langle v^k, x\rangle \mid x \in \Omega\},$$

where $v^k \in \partial h(x^k)$, we have

$$0 \in \partial g(x^{k+1}) + N_\Omega(x^{k+1}) - v^k.$$

Thus

$$v^k \in \partial h(x^k) \cap (\partial g(x^{k+1}) + N_\Omega(x^{k+1})).$$

By the strong-convexity of h and $v^k \in \partial h(x^k)$, then

$$h(x^{k+1}) \geq h(x^k) + \langle v^k, x^{k+1} - x^k\rangle + \frac{\rho_h}{2}\|x^{k+1} - x^k\|^2. \tag{1.20}$$

By the strong-convexity of g we have

$$g(x^k) \geq g(x^{k+1}) + \langle v, x^k - x^{k+1}\rangle + \frac{\rho_g}{2}\|x^k - x^{k+1}\|^2. \tag{1.21}$$

for all $v \in \partial g(x^{k+1})$. Taking $w \in N_\Omega(x^{k+1})$ and $v = v^k - w$, then (1.21) turns to

$$g(x^k) + \langle v^k - v, x^{k+1} - x^k\rangle - \frac{\rho_g}{2}\|x^{k+1} - x^k\|^2 \geq g(x^{k+1}). \tag{1.22}$$

By $w \in N_\Omega(x^{k+1})$, then

$$\langle w, x - x^{k+1}\rangle \leq 0, \forall x \in \Omega.$$

For every $k = 1,2,\ldots,$ we have $x^k \in \Omega$. Taking $x = x^k$ in the above inequality, we have

$$\langle w, x^k - x^{k+1}\rangle \leq 0. \tag{1.23}$$

It follows that for all $k = 1, 2, \ldots$

$$\begin{aligned} f(x^k + 1) &= g(x^{k+1}) - h(x^{k+1}) \\ &\leq g(x^{k+1}) - \left(h(x^k) + \langle v^k, x^{k+1} - x^k \rangle + \frac{\rho_h}{2}\|x^{k+1} - x^k\|^2\right) (1.21) \\ &\leq g(x^k) - h(x^k) + \langle w, x^k - x^{k+1} \rangle - \frac{\rho_g + \rho_h}{2}\|x^{k+1} - x^k\|^2 (1.22) \\ &\leq f(x^k) - \frac{\rho_g + \rho_h}{2}\|x^{k+1} - x^k\|^2, (1.23) \end{aligned} \tag{1.24}$$

which leads to the required inequality.
(Square summable property): Taking the sum of (1.19) for $k = 1, \ldots N(\geq 1)$, we have

$$\begin{aligned} \sum_{k=1}^{N} \frac{\rho_g + \rho_h}{2}\|x^k - x^{k+1}\|^2 &\leq \sum_{k=1}^{N} f(x^k) - f(x^k + 1) \\ &= f(x^1) - f(x^{N+1}) \\ &\leq f(x^1) - \min\{f(x) : x \in \Omega\} = f(x^1) - f^*, \end{aligned}$$

where the last inequality holds for any $N \geq 1$. Taking $N \to \infty$, and by the lower boundedness of f over Ω, then

$$\sum_{k\geq 1}^{N} \frac{\rho_g + \rho_h}{2}\|x^k - x^{k+1}\|^2 < \infty.$$

It follows immediately by $\rho_g + \rho_h > 0$ and the finiteness of $\|x^0 - x^1\|$ that

$$\sum_{k\geq 0}^{N} \|x^k - x^{k+1}\|^2 < \infty,$$

and as a consequence, $\|x^k - x^{k+1}\| \to 0$ as $k \to \infty$. □

Remark 1.7.4. *$\|x^k - x^{k+1}\| \to 0$ can be also derived from the sufficiently descent property and the convergence of $\{f(x^k)\}$ as*

$$0 \leq \frac{\rho_g + \rho_h}{2}\|x^k - x^{k+1}\|^2 \leq f(x^k) - f(x^{k+1}) \underrightarrow{k \to \infty} 0.$$

Remark 1.7.5. *Without the assumption $\rho_g + \rho_h > 0$, the sequence $\{\|x^k - x^{k+1}\|\}$ may not converge to 0. Consider the following counter example:*

Example 1.7.4. *Let $g(x) = \sup\{-x, 0, x - 1\}, h(x) = \sup\{-x, 0\}$ and $\Omega = \mathbb{R}$. The functions g and h are piecewise linear and convex, but neither strongly convex nor strictly convex. Starting DCA from an initial point $x^0 \in (0, 1)$, we get $\partial h(x^0) = \{0\}$, and $x^1 \in \arg\min\{g(x)\} = [0, 1]$. Hence, DCA could generate a sequence $\{x^k\} \subset (0, 1)$ such as $\{x^k\} = \{0.1, 0.9, 0.1, 0.9, \ldots\}$. Hence $\{\|x^k - x^{k+1}\|\}$ is a constant sequence $\{0.8, 0.8, \ldots\}$ whose limit is nonzero, and thus we don't have the square summable property. Note that the se-*

quence $\{f(x^k)\}$ is the constant zero-sequence which is convergent but without verifying the sufficiently descent property.

Corollary 1.7.2. *(***DCA Convergence rate***) If either g or h is strongly convex over Ω (i.e., $\rho_g + \rho_h > 0$), then, after N iterations of DCA, we have*

$$\frac{1}{N+1}\sum_{k=0}^{N}\|x^{k+1}-x^k\|^2 \leq \frac{2(f(x^0)-f^*)}{(\rho_g+\rho_h)N+1} = O\left(\frac{1}{N}\right),$$

where $f^* = \lim_{k\to\infty} f(x^k)$, and

$$\min_{0\leq k\leq N}\|x^{k+1}-x^k\| \leq \sqrt{\frac{2(f(x^0)-f^*)}{(\rho_g+\rho_h)N+1}} = O\left(\frac{1}{\sqrt{N}}\right).$$

Proof. We get from Lemma 1.7.7 that

$$\sum_{k=0}^{N}\frac{\rho_g+\rho_h}{2}\|x^k-x^{k+1}\|^2 \leq f(x^1)-\min\{f(x):x\in\Omega\} = f(x^1)-f^*.$$

Hence,

$$\sum_{k=0}^{N}\|x^{k+1}-x^k\|^2 \leq \frac{2(f(x^0)-f^*)}{(\rho_g+\rho_h)}.$$

It follows immediately that

$$\frac{1}{N+1}\sum_{k=0}^{N}\|x^{k+1}-x^k\|^2 \leq \frac{2(f(x^0)-f^*)}{(\rho_g+\rho_h)N+1} = O\left(\frac{1}{N}\right),$$

and

$$\min_{0\leq k\leq N}\|x^{k+1}-x^k\| \leq \sqrt{\frac{2(f(x^0)-f^*)}{(\rho_g+\rho_h)N+1}} = O\left(\frac{1}{\sqrt{N}}\right).$$

□

Remark 1.7.6. *Corollary 1.7.2 indicates that the convergence rate depends on $f(x^0)-f^*$ (i.e., the initial point) and $\rho_g+\rho_h$ (i.e., the quality of the DC decomposition). A smaller $f(x^0)-f^*$ and a larger $\rho_g+\rho_h$ lead to a faster convergence. The convergence rate is sub-linear.*

1.7.5 The Point-wise Maximum, Minimum, and Convergence of the DCA

The maximum function is defined as the point-wise maximum of convex functions. For $j = 1, 2, \ldots, n$, let functions $f_j : \mathbb{R}^n \to \mathbb{R}$ be closed and convex. Then the maximum function

$$f(x) := \max_{j=1,\ldots,n} f_j(x) = \max\{f_1(x), f_2(x), \ldots, f_n(x)\},$$

is also closed and convex.
On the other hand, the minimum function $f(x)$, defined by

$$f(x) := \min_{j=1,\ldots,n} f_j(x) = \min\{f_1(x), f_2(x), \ldots, f_n(x)\},$$

may not be convex. However, it can always be represented as a difference of two convex functions as follows:

$$\min_{i=1,\ldots,m} f_i(x) = \sum_{i=1}^{m} f_i(x) - \max_{t=1,\ldots,m} \sum_{i=1, i\neq t}^{m} f_i(x). \tag{1.25}$$

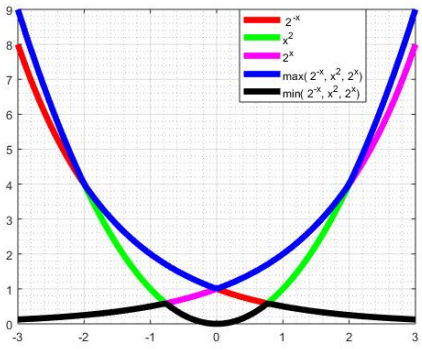


Figure 1.2: Point-wise minimum and maximum of convex functions.

Lemma 1.7.8. *(B. Mordukhovich & Nam, 2014) Let $f_1, f_2, \ldots, f_m : \mathbb{R}^n \to \bar{\mathbb{R}}$ be proper extended real-valued convex functions and let $f(x) = \max\{f_1(x), f_2(x), ..., f_m(x)\}$. if $x \in \bigcap_{i=1}^m int(dom(f_i))$, then*

$$\partial f(x) = co\left(\bigcup_{i \in I(x)} \partial f_i(x)\right),$$

where $I(x) = \{i \mid f_i(x) = f(x), i = 1, 2, ..., m\}$.

Proof. Let f be the maximum function given in Theorem 1.7.3 then we have

$$epi(f) = \bigcap_{i=1}^{m} epi(f_i).$$

$ri(epi(f_i)) = \{(x,\lambda) \mid x \in ri(dom(f_i)), \lambda > f_i(x)\} = f(x,\lambda) \mid x \in int(dom(f_i)), \lambda > f_i(x)\}$,which imply that $(\bar{x}, f(\bar{x}) + 1) \in \bigcap_{i=1}^m int(epi(f_i)) = \bigcap_{i=1}^m ri(epi(f_i))$. Furthermore, since $f_i(\bar{x}) < f(\bar{x}) = \bar{\alpha}$ for any $i \notin I(\bar{x})$, there exists a neighborhood U of $\bar{x}$ and $\varepsilon > 0$ such that $f_i(x) < \alpha$ whenever $(x, \alpha) \in U \times (\bar{\alpha} - \varepsilon, \bar{\alpha} + \varepsilon)$. It follows that $(\bar{x},) \in \text{int}(epi(f_i))$, and so $N((\bar{x}, \bar{\alpha}), epi(f_i)) = \{(0,0)\}$ for such induces i. Thus

$$N((\bar{x}, f(\bar{x})); epi(f)) = \sum_{i=1}^{m} N((\bar{x}, \bar{\alpha}); epi(f_i)) = \sum_{x \in I(\bar{x})} N((\bar{x}, f_i(\bar{x})); epi(f_i)).$$

Picking now $v \in \partial f(\bar{x})$, we have by Proposition 1.7.9 that $(v, -1) \in N((\bar{x}, f(\bar{x})); epif)$,which allows us to find $(v_i, -\lambda_i) \in N((\bar{x}; f_i(\bar{x})); epif_i)$ for $i \in I(\bar{x})$ such that

$$(v, -1) = \sum_{i \in I(\bar{x})} (v_i, -\lambda_i).$$

This yields $\sum_{i \in I(\bar{x})} \lambda_i = 1, \lambda_i \geq 0, v = \sum_{x \in I(\bar{x})} v_i, and v_i \in \lambda_i \partial f_i(\bar{x})$ Thus $v = \sum_{i \in I(\bar{x})} \lambda_i u_i$, where $u_i \in \partial f_i(\bar{x})$ and $\sum_{i \in I(\bar{x})} \lambda_i = 1$. This verifies that

$$v \in co\left(\bigcup_{i \in I(x)} \partial f_i(x)\right).$$

The opposite inclusion in the maximum rule follows from

$$\partial f_i(\bar{x}) \subset \partial f(\bar{x}), \forall i \in I(\bar{x}).$$

□

1.7.6 Nesterov's Smoothing Approximation of the ℓ_1 Norm

In this subsection, the more natural approach to solve (1.4) is to find a smooth approximation f_γ to f that is ℓ-smooth. If the approximation is good enough, then we can simply apply DCA to f_γ to achieve desirable performance. The objective is now simple: design a good approximation f_γ of f so that minimizing f_γ is as close to solving (1.4) as possible. There are many ways to find a smooth approximation of f_γ, but we will focus on Nesterov's technique. Assume that f from (1.4) can be represented by

$$f(x) = \max\{\langle Ax, u\rangle - \phi(u) : u \in F\}, \tag{1.26}$$

where $\phi : F \subset \mathbb{R}^n \to (-\infty, \infty]$ is proper, lower semi-continuous convex function and F is a convex, compact set. Nesterov proposes the following smooth approximation of f

$$f_\gamma(x) = \max_{u \in F}\{\langle Ax, u\rangle - \phi(u) - \gamma d(u)\}, \tag{1.27}$$

where $d(u)$ is called a proximity function that satisfies the following assumptions:

- $d(u) : \mathbb{R}^m \longrightarrow \mathbb{R}$ is continuous and strongly convex on F with parameter $\rho = 1$,
- $\text{dom}(\phi) \subset \text{dom}(d)$,
- $d(u) \geq 0, \forall u \in \text{dom}(\phi)$,
- $\min_{u \in F} d(u) = 0$ and $\gamma > 0$.

Before we discuss the properties of f_γ in (1.27), let us first examine the assumption in (1.26). Recall that if f is convex, lower semi-continuous, and proper, i.e. $\min_x f(x) = f_* > -\infty$, then $(f^*)^* = f$ where f^* denotes the Fenchel conjugate (or Fenchel transform). We know previously that f^* is convex. Thus, if we could show that $\text{dom}(f^*)$ is convex, compact, then f always has at least one representation of the form in (1.26). There are numerous assumptions that ensure this condition. For example, if f itself is Lipschitz continuous, then dom f^* is compact (see (Rockafellar, 1970)). That is, the assumptions made previously are not particularly stringent. We proceed to proving properties of the smooth approximation (1.27)

Definition 1.7.19. *(B. Mordukhovich & Nam, 2014) Let F be a nonempty closed subset of $\mathbb{R}^n$ and let $x \in \mathbb{R}^n$.*

1. *Define the distance between x and set F by*

$$d(x; F) = \inf\{\|x - w\| \mid w \in F\}.$$

2. *The set of all Euclidean projection from x to F is defined by*

$$P(x,F) = \{w \in F \mid d(x;F) = \|x-w\|\}.$$

It is well-known that $P(x;F)$ is nonempty when $F \subset \mathbb{R}^n$ is closed. If we assume in addition that F is convex, than $P(x,F)$ is a singleton. Moreover, we can show that if F is a convex set and $w \in P(x;F)$, then $x-w \in N(w,F)$; see, e.g.,(B. Mordukhovich & Nam, 2014; Rockafellar, 1970) and the references therein.

Proposition 1.7.19. *(M. N. Nguyen et al., 2019) Given any $a \in \mathbb{R}^n$ and $\gamma > 0$, A Nesterov smoothing approximation of $\varphi(x) = ||x-a||_1$ has the representation*

$$\varphi_\gamma(x) := \frac{1}{2\gamma}\|x-a\|^2 - \frac{\gamma}{2}\left[d(\frac{x-a}{\gamma};F)\right]^2.$$

where F is the closed unit box of $\mathbb{R}^n$, i.e., $F := \{v = (v^1,...,v^n) \in \mathbb{R}^n \mid -1 \le v_i \le 1$ for $i = 1,...,n.\}$
Moreover,

$$\nabla\varphi_\gamma(x) = P\left(\frac{x-a}{\gamma};F\right) \text{ and } \varphi_\gamma(x) \le \varphi(x) \le \varphi_\gamma(x)) + \frac{\gamma}{2}\|F\|^2, \tag{1.28}$$

where $\|F\| := \sup\{\|q\| \mid q \in F\}$.

Proof. The function φ can be represented as

$$\varphi(x) = ||x-a||_1 = \sup\{\langle x-a,u\rangle \mid u \in F\}.$$

Using the prox-function $d(x) = \frac{1}{2}||x||^2$ in (Nam et al., 2017), one obtains a smooth approximation of φ given by,

$$\begin{aligned}
\varphi_\gamma(x) &:= \sup\{\langle x-a,u\rangle - \frac{\gamma}{2}||u||^2 \mid u \in F\} \\
&= \sup\{-\frac{\gamma}{2}(||u||^2 - \frac{2}{\gamma}\langle x-a,u\rangle) \mid u \in F\} \\
&= \sup\{-\frac{\gamma}{2}||u - \frac{1}{\gamma}(x-a)||^2 + \frac{1}{2\gamma}||x-a||^2 \mid u \in F\} \\
&= \frac{1}{2\gamma}||x-a||^2 - \frac{\gamma}{2}\inf\{||u - \frac{1}{\gamma}(x-a)||^2 \mid u \in F\} \\
&= \frac{1}{2\gamma}||x-a||^2 - \frac{\gamma}{2}[d(\frac{x-a}{\gamma};F)]^2.
\end{aligned}$$

The estimation of $\varphi_\gamma(x) \le \varphi(x) \le \varphi_\gamma(x) + \frac{\gamma}{2}\|F\|^2$ is straightforward.

Obviously,

$$\varphi_\gamma(x) = \sup\left\{\langle x-a,u\rangle - \frac{\gamma}{2}||u||^2 \mid u \in F\right\} \leq \sup\{\langle x-a,u\rangle \mid u \in F\} = \varphi(x)$$

for all $x \in R^n$.
In addition,we have

$$\begin{aligned}
\varphi(x) &= \sup\{\langle x-a,u\rangle \mid u \in F\} \\
&= \sup\left\{\langle x-a,u\rangle - \frac{\gamma}{2}||u||^2 + \frac{\gamma}{2}||u||^2 \mid u \in F\right\} \\
&\leq \sup\left\{\langle x-a,u\rangle - \frac{\gamma}{2}||u||^2 \mid u \in F\right\} + \sup\left\{\frac{\gamma}{2}||u||^2 \mid u \in F\right\} \\
&= \varphi_\gamma(x) + D, \text{ for } D = \sup\left\{\frac{\gamma}{2}||u||^2 \mid u \in F\right\}.
\end{aligned}$$

□

Proposition 1.7.20. *(Nam et al., 2017) Let F be a nonempty closed set in $\mathbb{R}^n$ (not necessarily convex). Define the function*

$$\varphi_F(x) := \sup\{\langle 2x,w\rangle - \|w\|^2 : w \in F\} = 2\sup\{\langle x,w\rangle - \|w\|^2 : w \in F\}.$$

Then we have the following conclusions:

1. *The function φ_F is always convex. If we assume in addition that F is convex, then φ_F is differentiable with*
$$\nabla\varphi_F(x) = 2P(x,F).$$
2. *The function $f(x) := [d(x,F)]^2$ is a DC function with*
$$f(x) = \|x\|^2 - \varphi_F(x) \text{ for all } x \in \mathbb{R}^n.$$

Proof. (i) We have the following obvious representation for the squared distance function from x to F:

$$\begin{aligned}
[d(x,F)]^2 &= \inf\{\|x-w\|^2 : w \in F\} \\
&= \inf\{\|x\|^2 - 2\langle x,w\rangle + \|w\|^2 : w \in F\} \\
&= \|x\|^2 + \inf\{\|w\|^2 - 2\langle x,w\rangle : w \in F\} \\
&= \|x\|^2 - \sup\{\langle 2x,w\rangle - \|w\|^2 : w \in F\}.
\end{aligned}$$

It follows from the representation of $[d(x,F)]^2$ above that

$$[d(x,F)]^2 = \|x\|^2 - \varphi_F(x). \tag{1.29}$$

The function φ_F is convex because it is the supremum of a family of convex function. Note that, the squared distance function $\psi(x) := [d(x,F)]^2$ is differentiable with $\nabla\psi(x) = 2[x - P(x;F)]$ see,(B. Mordukhovich & Nam, 2014). Then the function φ_F is differentiable with

$$\nabla\varphi_F(x) = 2x - 2[x - P(x,F)] = 2P(x;F),$$

which completes the proof.
(ii) From (1.29) we see that f is a DC function (even if F is non-convex). □

Example 1.7.5. *Consider the function $f(x) = \|x\|$ which is not differentiable at $x = 0$ and can be represented as*

$$f(x) = \max\{ux \mid u \in [-1,1]\}, x \in \mathbb{R}.$$

Using the prox-function $q(u) = \frac{1}{2}u^2$ gives us

$$f_\gamma(x) = \max\left\{ux - \frac{\gamma}{2}u^2 \mid u \in [-1,1]\right\}.$$

And

$$u_\gamma = \begin{cases} -1, x \le -\gamma, \\ \frac{x}{\gamma}, -\gamma < x < \gamma, \\ 1, x \ge \gamma. \end{cases}$$

Therefore,

$$f_\gamma(x) = \begin{cases} -x - \frac{\gamma}{2}, x \le -\gamma, \\ \frac{x^2}{2\gamma}, -\gamma < x < \gamma, \\ x - \frac{\gamma}{2}, x \ge \gamma. \end{cases}$$

In this case the Lipschitz constant of $\nabla f_\gamma(x)$ is $L_\gamma = \frac{1}{\gamma}$.

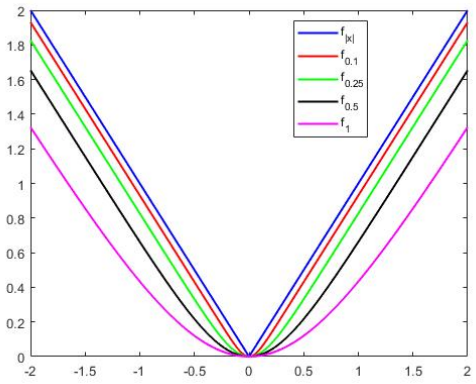


Figure 1.3: The smooth graph of $f = |x|$ for different values of γ.

CHAPTER 2

LITERATURE REVIEW

Clustering problems are inherently discrete and combinatorial, presenting significant challenges due to their non-convex and non-smooth nature. One effective approach to navigate these complexities is through the class of DC functions, which stands for the difference of convex functions. Although the utility of DC functions has been recognized in optimization theory for some time, the first algorithmic framework was introduced by Pham Dinh Tao in 1985. His algorithm, designed for minimizing $f = g - h$ (where, $g, h : \mathbb{R}^n \to \mathbb{R}$ are convex functions), is known as the DCA. This method uses the sub-gradients of h and the Fenchel conjugate of g, facilitating the approximation of a non-convex DC program through a sequence of convex problems.

Despite its innovations, the DCA's convergence properties can be sensitive to DC decomposition of the functions involved. In particular, there are cases where the algorithm converge to suboptimal solutions, raising questions about its robustness in diverse applications. Future research should explore the optimal DC decomposition which is an open problem and investigate alternative methods that could offer better guarantees under certain conditions.

Over the past four decades, various methods have been developed to tackle non-smooth unconstrained optimization problems, notably in data mining. Among these are the bundle method and its variations (Hiriart-Urruty et al., 2004; Kiwiel, 1990; Lemarechal & Zowe, 1994), as well as algorithms based on smoothing techniques (Polak & Royset, 2003) and random sub-gradient sampling (Burke et al., 2002). However, many of these approaches require the computation of at least one sub-gradient in each iteration, which can be challenging for practical problems. This limitation highlights the need for derivative-free methods, which do not rely on explicit gradient calculations. In 2008, Bagirov (A. M. Bagirov et al., 2008) introduced a novel derivative-free method for solving unconstrained non-smooth optimization problems, based on the concept of discrete gradients. While this approach offers a promising alternative, its effectiveness compared to traditional gradient-based methods remains under explored. A comparative analysis of convergence rates, robustness to noise, and computational efficiency across various problem instances would provide valuable insights into the practical applicability of these methods in high-dimensional settings.

Numerous studies on clustering have been conducted in this area, including the minimum sum of squares clustering problem (A. Bagirov et al., 2016). In this work, a new version of the k-means algorithm, known as the global k-means algorithm, was developed because the original k-means algorithm and its variations are sensitive to the choice of starting points and inefficient for solving clustering problems in large data sets. It is an incremental algorithm

that dynamically adds one cluster center at a time and uses each data point as a candidate for the k^{th} cluster center. A starting point for the k^{th} cluster center in this algorithm is computed by minimizing an auxiliary cluster function.

Another work by (Xavier & Xavier, 2011) proposed resolution method, called Hyperbolic Smoothing, adopts a smoothing strategy using a special C^{∞} differentiable class function. The final solution is obtained by solving a sequence of low dimension differentiable unconstrained optimization subproblems which gradually approach the original problem.

The combination of Nesterov's smoothing technique (Nesterov, 2005, 2018) , DC programming and DCA (THI & DINH, 2018) introduced an efficient platform to study non-convex and non-smooth problem in optimization. DCA has applied to facility location, compressed sensing, and imaging, supply chain management and telecommunication successfully (Nam et al., 2017; L. T. H. Nguyen P.A., 2020.; THI & DINH, 2018). The challenging non-smoothing in clustering is made easier by Nesterov's smoothing approaches, but choosing the appropriate smoothing parameter requires research because it affects the DCA's convergence and speed of convergence.

In 2003, Bagirov (A. Bagirov et al., 2003) introduced three models of hierarchical clustering based on the Euclidean norm and employed the derivative-free method developed in (A. Bagirov, 1999) to solve the problem in $\mathbb{R}^2$. But this method is not suitable for large-scale settings in high dimensions. The DCA was successfully applied in (T. P. An L.T.H., 1997) to solve these models in which the similarity measure is defined by the squared Euclidean distance. Although the DCA in (T. P. An L.T.H., 1997) provides an effective way to solve the bi-level hierarchical clustering in high dimensions, it has not been used to solve the original model defined by the Euclidean distance measurement proposed in (A. Bagirov et al., 2003) which is not suitable for the resulting DCA according to the authors of (T. P. An L.T.H., 1997).

In 2022, solving multifacility location problems by DC algorithms were studied in (Bajaj et al., 2022). This problem was continuation of their recent effort in applying continuous optimization techniques to study optimal multicast communication networks modeled as bi-level hierarchical clustering problems with ℓ_2-norm (Geremew et al., 2018; Nam et al., 2018). A similar problem was also investigated in (L. An & Minh, 2007) using different approach. The significant difference between problem studied in (Bajaj et al., 2022) and (L. An & Minh, 2007) is a squared Euclidean norm is used in the former while Euclidean norm is applied in the later case.

DC algorithm has laid an efficient foundation in solving different non-convex problems in clustering, see for example in (L. An et al., 2007, 2014; Barbosa & Villas-Boas, 2013; Nam et al., 2017) and references cited therein. The authors in (Geremew et al., 2018; Nam et al., 2018) implemented a new way on Nesterov's and DC Algorithm to overcome the problem in (A. Bagirov et al., 2003). They considered two different models of multicast networks by identifying a certain number of nodes as cluster centers, and at the same time, locating

a particular node that serves as a total center so as to minimize the total transportation cost throughout the network. The idea of using Nesterov's smoothing techniques overcomes the drawback of applying DCA stated in (L. An & Minh, 2007) as the model is not appropriate to implement the DCA.

Clustering problems with the ℓ_1-norm have attracted less attention than those with the squared Euclidean norm. In some applications clustering algorithms with the ℓ_1-norm produce easy interpreting results than those with the squared ℓ_2-norm. The former algorithms are more preferred in high dimensional data mining applications (Aggarwal et al., 2001) and they are also more robust to outliers (Zhang et al., 2012). To the best of our knowledge, clustering problem with the ℓ_1-norm was first considered in (Carmichael & Sneath, 1969) (see also (Kaufman & Rousseeuw, 2009)). The k-median algorithm was proposed in (Späth, 1976) and the ISODATA algorithm was introduced in (Jajuga, 1987). The X-means algorithm, introduced in (Pelleg et al., 2000), allows one to use the ℓ_1-norm based similarity measure. The local search optimization based clustering algorithm is presented (Sabo, 2014). The optimization algorithm using smoothing techniques was introduced in (A. M. Bagirov & Mohebi, 2016) and the non-smooth optimization clustering algorithm was proposed in (A. M. Bagirov & Mohebi, 2015). A DC Optimization Algorithm for Clustering Problems with ℓ_1-norm was introduced by (A. Bagirov & Taheri, 2017), the main contributions of their work was development of optimality conditions for the clustering problem using the DC representation of its objective function and a hyperbolic partial smoothing technique for approximation of the clustering functions. Numerical results demonstrate that the proposed algorithm was efficient for clustering on data sets containing hundreds of thousands of data points. However, it also had some limitations.

In most applied problems, ℓ_1 distance measures give a better approximation of the reality than Euclidean distance due to the obstacle during measurement. In this dissertation, we further study bi-level hierarchical clustering models proposed in (Geremew et al., 2018; Nam et al., 2018) by modifying the objective function and constraints using ℓ_1 norm. Since the distance is measured by the ℓ_1 norm , we employed Nesterov's partial smoothing techniques and appropriate DC decomposition that helps to apply DC Algorithm (DCA). Besides, the change made to the objective function and constraints, the centers and headquarter are forced to lie on nodes in the dataset and headquarter is also forced to lie in the convex hull of the selected centers that minimize the overall distance of the network. For this, we used penalty parameters to convert constrained problem to unconstrained one.

CHAPTER 3

RESEARCH METHODOLOGY

This chapter describes an overview of the study area, study period, data sources, data analysis methodologies, mathematical procedures, and a detailed description of the modified model employed to fulfill the objectives of the study.

3.1 Study Area

The study of the proposed dissertation was conducted in ASTU. To validate our models for proximity, some information about the locations (longituid and latitude) of towns and cities was gathered from Oromia/ Ethiopia.

3.2 Study Period

The study carried out in Adama Science and Technology University(ASTU) Adama, Ethiopia, from October 2017 to October 2024.

3.3 Source of Data

The study used three different data types. First we used a randomly generated artificial data set to test the algorithm. Second, we used a GPS data collected from some towns and cities in Ethiopia to apply the algorithm to real data sets. Third, we used secondary public data sets published to compare our methods with other clustering algorithm.

3.4 Data Analysis

In the proposed models analysis, we focused on both analytical and numerical approaches in the examination of the suggested models. In order to examine the data, we implemented a modified DC algorithm using the MATLAB software. The resulting outcomes are shown graphically in tables and graphs.

3.5 Mathematical Procedure

The cost functions under study are neither smooth nor convex. Although convex and smooth optimization techniques and numerical algorithms have been the topics of extensive research for more than 50 years, solving large-scale optimization problems without the presence of convexity and smoothness remains a challenge. So, the widely available constrained convex optimization solvers can not be used to solve the clustering optimization problems under study. Instead, we applied Nesterov's smoothing technique to find a smooth approximation of the Manhattan norm. In the process, we gained partial smoothness, but convexity of the cost function is lost. However, the smoothed cost function can be expressed as a difference of two convex functions. The good news is that, the DC (Difference of Convex) representation of the partially smoothed cost function can be manipulated by a well known local search method called the DCA (Difference of Convex Algorithm). To implement DCA we calculated gradient and sub-gradients of convex functions. Specially we used the notion of point-wise maximum functions and its sub-gradient calculations.
The smoothed approximation of those three models under investigation was addressed by a DCA-based algorithm that we developed. The chosen cluster center was pushed to a real node by setting the penalty parameter to push the search space to a practical location. We used MATLAB to create the algorithm and ran it with multiple starting points (seeds). Next, we selected the least expensive cost. There is no assurance that the algorithm converged to the global minimal cost because the DCA is a local search technique. Nonetheless, we believe that our approach offers a reliable approximation of the global solution.

3.6 Problems Formulation

Clustering is typically a partition of N data points into k groups (clusters) such that the points in each group are more similar to each other than to points in other groups. To define our problems, consider a set A of m data points, that is $A = \left\{a^i \in \mathbb{R}^n : i = 1,...,m\right\}$ and k variable cluster centers denoted by $x^1,...,x^k$. We modeled a clustering model and two different network topologies of bi-level hierarchal clustering problem by choosing k separate centers for clustering problem. Now we formulate the clustering model as

$$\begin{aligned} &\text{Minimize} \quad \left(\sum_{i=1}^{m} \min_{j=1,...,k} \|x^j - a^i\|_1\right) \\ &\text{subject to} \quad \sum_{j=1}^{k} \min_{i=1,...m} \|x^j - a^i\|_1 = 0, \;\; x^1,...,x^k \in \mathbb{R}^n. \end{aligned}$$

For bi-level hierarchal clustering models we choose k separate centers and a headquarters .

Other members of the data is assigned to one of the centers based on the ℓ_1 norm between the data points and centers. In model I, nodes are grouped in to k variable centers by minimizing the ℓ_1- norm from all node to the k centers. Then headquarter is defined to lie in convex hull of x^j for $j = 1,..,k$. i.e $\bar{x} = \sum_{j=1}^{k} \lambda_j x^j$ where $\lambda_j \geq 0,\ \sum_{j=1}^{k} \lambda_j = 1$. This constraint limits the search region for headquarter to the convex hull of centers.

A node that minimize the ℓ_1 distance will serve as the headquarter. Since $\bar{x}$ is artificial center, penalty is used to push the headquarter to lie on one of the nodes that minimize the total distance. Thus we formulate the bi-level hierarchal clustering models as follows:

MODEL I:

$$\begin{aligned} \text{Minimize} \quad & \left(\sum_{i=1}^{m} \min_{j=1,\ldots,k+1} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - \bar{x}\|_1 \right) \\ \text{subject to} \quad & \sum_{j=1}^{k+1} \min_{i=1,\ldots m} \|x^j - a^i\|_1 = 0. \end{aligned}$$

MODEL II:

$$\begin{aligned} \text{Minimize} \quad & \left(\sum_{i=1}^{m} \min_{j=1,\ldots,k} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - \bar{x}\|_1 \right) \\ \text{subject to} \quad & \sum_{j=1}^{k+1} \min_{i=1,\ldots m} \|x^j - a^i\|_1 = 0. \end{aligned}$$

where,

$$\bar{x} = \frac{1}{k} \sum_{j=1}^{k} x^j.$$

In model II, nodes are grouped into k variable centers by minimizing the ℓ_1 distances from all node to k centers. Then a headquarter is a center that minimize the overall distance of the network and in the convex hull of centers. The basic difference in Model I and Model II is that in Model I the headquarter is also serve as a cluster center. The fact that the centers and the headquarters have to be among the existing nodes make the problems a discrete optimization problem which is NP-hard.

In this dissertation, we propose the use of artificial centers (geographical locations that are not necessarily existing nodes), and formulate three continuous optimization models whose solutions can be used to find optimal cluster centers and headquarters.

CHAPTER 4

CONSTRAINED CLUSTERING

4.1 A DC Optimization Approach for Constrained Clustering with ℓ_1 Norm.

4.1.1 Problems Formulation

Clustering is typically a partition of N data points into k groups (clusters) such that the points in each group are more similar to each other than to points in other groups. To define our problems, consider a set A of i data points, that is $A = \left\{a^i \in \mathbb{R}^n : i = 1,...,m\right\}$ and k variable cluster centers denoted by $x^1,...,x^k$. We modeled the clustering problem by choosing k separate centers from the data points that minimize the overall distance. Other members of the data will be assigned to one of the centers based on the ℓ_1- norm between the data points and centers. In the model, nodes are grouped in to k variable centers by minimizing the ℓ_1 norm from all node to the k centers. Then we define clustering model as

$$\text{Minimize} \left(\sum_{i=1}^{m} \min_{j=1,...,k} \|x^j - a^i\|_1 \right) \tag{4.1}$$

where the distance measure is the ℓ_1-norm defined as

$$||x||_1 = \sum_{j=1}^{n} |x_j|.$$

In addition, to make sure that the k centers are selected from the existing nodes, we need to add the following constraint to the optimization problem formulated in (4.1):

$$\sum_{j=1}^{k} \min_{i=1,...,m} \|x^j - a^i\|_1 = 0. \tag{4.2}$$

The constraints given in (4.2) helps to push the artificial center to the closest node. Now we formulate the clustering model as

$$\begin{aligned} \text{Minimize} \quad & \left(\sum_{i=1}^{m} \min_{j=1,\ldots,k} \|x^j - a^i\|_1 \right) \\ \text{subject to} \quad & \sum_{j=1}^{k} \min_{i=1,\ldots m} \|x^j - a^i\|_1 = 0, \;\; x^1,\ldots,x^k \in \mathbb{R}^n. \end{aligned} \tag{4.3}$$

To solve the problem given in (4.3) we apply penalty method with parameter $\tau > 0$. Then we can express the minimization problem as unconstrained problem as follows:

$$\text{Minimize} \sum_{i=1}^{m} \min_{j=1,\ldots,k} \|x^j - a^i\|_1 + \tau \sum_{j=1}^{k} \min_{i=1,\ldots,m} \|x^j - a^i\|_1, \;\; x^1,\ldots,x^k \in \mathbb{R}^n \tag{4.4}$$

Since the pointwise minimum of convex functions may not be convex, we can suitably write as difference of convex functions. as given in (4.5).

$$\min_{i=1,\ldots,m} f_i(x) = \sum_{i=1}^{m} f_i(x) - \max_{t=1,\ldots,m} \sum_{i=1, i\neq t}^{m} f_i(x). \tag{4.5}$$

Rewriting the minimum of convex function in (4.4) as sum and maximum of convex functions we have

$$\begin{aligned} f(x^1,x^2,\ldots,x^k) \;\; = \;\; & \sum_{i=1}^{m}\sum_{j=1}^{k} \|x^j - a^i\|_1 - \sum_{i=1}^{m} \max_{t=1,\ldots,k} \sum_{j=1, j\neq t}^{k} \|x^j - a^i\|_1 \\ & + \tau \sum_{i=1}^{m}\sum_{j=1}^{k} \|x^j - a^i\|_1 - \tau \max_{t=1,\ldots,k} \sum_{j=1, j\neq t}^{k} \|x^j - a^i\|_1 \end{aligned}$$

Writing as DC function it becomes

$$\begin{aligned} f(x^1,x^2,\ldots,x^k) \;\; = \;\; & (1+\tau) \sum_{i=1}^{m}\sum_{j=1}^{k} \|x^j - a^i\|_1 - \sum_{i=1}^{m} \max_{t=1,\ldots,k} \sum_{j=1, j\neq t}^{k} \|x^j - a^i\|_1 \\ & - \tau \sum_{j=1}^{k} \max_{r=1,\ldots,m} \sum_{i=1, i\neq r}^{m} \|x^j - a^i\|_1 \end{aligned} \tag{4.6}$$

where the corresponding convex functions g and h are given as

$$g(x^1,x^2,\ldots,x^k) = (1+\tau) \sum_{i=1}^{m}\sum_{j=1}^{k} \|x^j - a^i\|_1$$

$$h(x^1,x^2,...,x^k)=\sum_{i=1}^{m}\max_{t=1,...,k}\sum_{j=1,j\neq t}^{k}\|x^j-a^i\|_1+\tau\sum_{j=1}^{k}\max_{r=1,...,m}\sum_{i=1,i\neq r}^{m}\|x^j-a^i\|_1$$

Since f is DC function based on ℓ_1 smoothing studied in (M. N. Nguyen et al., 2019), we give a Nesterov's approximation of $||x-a||_1$ as define in Proposition 1.7.19 :

$$\|x-a\|_1=\frac{\gamma}{2}\left[\|\frac{x-a}{\gamma}\|^2-[d(\frac{x-a}{\gamma};F)]^2\right]$$

and a Nesterov smoothed objective function in (4.6) is as follows.

$$\begin{aligned}f_\gamma(x^1,x^2,...,x^k) &= \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}\|\frac{x^j-a^i}{\gamma}\|^2\\ &-\frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}[d(\frac{x^j-a^i}{\gamma};F)]^2\\ &-\sum_{i=1}^{m}\max_{t=1,...,k}\sum_{j=1,j\neq t}^{k}\|x^j-a^i\|_1-\tau\sum_{j=1}^{k}\max_{r=1,...,m}\sum_{i=1,i\neq r}^{m}\|x^j-a^i\|_1)\end{aligned}\tag{4.7}$$

The aim is to minimize the smoothed problem f_γ as:

$$\text{Minimize}\quad\left\{f_\gamma(x^1,x^2,...,x^k)=g_\gamma(x^1,x^2,...,x^k)-h_\gamma(x^1,x^2,...,x^k)\right\}$$

where

$$g_\gamma(x^1,x^2,...,x^k)=\frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}\|\frac{x^j-a^i}{\gamma}\|^2.\tag{4.8}$$

$$\begin{aligned}h_\gamma(x^1,x^2,...,x^k) &= \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}[d(\frac{x^j-a^i}{\gamma};F)]^2+\sum_{i=1}^{m}\max_{t=1,...,k}\sum_{j=1,j\neq t}^{k}\|x^j-a^i\|_1\\ &+\tau\sum_{j=1}^{k}\max_{r=1,...,m}\sum_{i=1,i\neq r}^{m}\|x^j-a^i\|_1.\end{aligned}$$

The next proposition is a direct consequence of Proposition1.7.19.

Proposition 4.1.1. *(Geremew et al., 2018) Given any $\lambda>0$ and $\gamma>0$, the functions f and f_γ satisfy*

$$f_\gamma(x^1,\ldots,x^k)\leq f(x^1,\ldots,x^k)\leq f_\gamma(x^1,\ldots,x^k)+mk(1+\frac{\lambda}{2})\gamma\parallel F\parallel^2,\ \ \forall x^1,\ldots,x^k\in\mathbb{R}^n.$$

In what follows we will prove that each of the functions f and f_γ admits an absolute minimum in $(R^n)^k$.

Theorem 4.1.1. *Given any $\lambda > 0$ and $\gamma > 0$, each of the functions f and f_γ has an absolute minimum in $(R^n)^k$.*

Proof. (Geremew et al., 2018) Let us show that for any $\alpha \in \mathbb{R}$, the sub-level set $L_\alpha := \{(x^1,\ldots,x^k) \mid f(x^1,\ldots,x^k) \leqslant \alpha\}$is bounded in $(R^n)^k$. Since $0 \in int(F)$, there exists $r > 0$ such that $B(0;r) \subset F$. Consequently,

$$r||x|| = \sup\{\langle x,u\rangle | u \in B(0;r)\} \leq \sup\{\langle x,u\rangle | u \in F\} = \| x^j - a^i \|$$

for all $x \in \mathbb{R}^n$.
From the definition of the function f , we have

$$\begin{aligned}
&\left\{(x^1,\ldots,x^k) \in (R^n)^k \mid f(x^1,\ldots,x^k) \leqslant \alpha\right\} \\
\subset &\left\{(x^1,\ldots,x^k) \in (R^n)^k \mid \min_{i=1,\ldots,m} \sum_{j=1}^{k} \| x^j - a^i \| \leqslant \alpha\right\} \\
\subset &\left\{(x^1,\ldots,x^k) \in (R^n)^k \mid \min_{i=1,\ldots,m} \sum_{j=1}^{k} \| x^j - a^i \| \leqslant \frac{\alpha}{r}\right\} \\
\subset &\bigcup_{i=1}^{m} \left\{(x^1,\ldots,x^k) \mid \varphi_i(x^1,\ldots,x^k) \leqslant \frac{\alpha}{r}\right\},
\end{aligned}$$

where $\varphi_i(x^1,\ldots,x^k) := \sum_{j=1}^{k} \|x^j - a^i\|$. Observe that for each $i = 1,\ldots,m$, one has the inclusion

$$\left\{(x^1,\ldots,x^k) \mid \varphi_i(x^1,\ldots,x^k) \leqslant \frac{\gamma}{r}\right\} \subset \{(x^1,\ldots,x^k) \mid \sum_{j=1}^{k} \| x^j \| \leqslant \frac{\alpha}{r} + k\|a^i\|\}$$

Thus, L_γ is a bounded set as it is contained in the union of a finite number of bounded sets in $(R^n)^k$. As f is a continuous function, it has an absolute minimum in $(R^n)^k$. Let $\alpha_\gamma := mk(1+\frac{\lambda}{2})\gamma\|F\|^2$. It follows from Proposition 4.1.1 that for any $\gamma \in \mathbb{R}$,

$$\left\{(x^1,...,x^k) \in (R^n)^k \mid f_\gamma(x^1,\ldots,x^k) \leq \alpha\right\}$$

$$\subset \left\{(x^1,\ldots,x^k) \in (R^n)^k \mid f(x^1,\ldots,x^k) \leq \alpha_\gamma + \alpha\right\}.$$

It follows that the sub-level set $\left\{(x^1,\ldots,x^k) \in (R^n)^k \mid f_\gamma(x^1,\ldots,x^k) \leqslant \alpha\right\}$ is also bounded, and hence f_γ has an absolute minimum in $(R^n)^k$. □

To facilitate the gradient and sub-gradient calculations for the DCA, we will introduce a data matrix A and a variable matrix X. The data matrix A is formed by putting each a^i , $i = 1,\ldots,m$, in the i^{th} row, i.e.,

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

Similarly, if $x^1, \ldots, x^k$ are the k cluster centers, then the variable X is formed by putting each $x^j, j = 1, \ldots, k$, in the j^{th} row, i.e.,

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kn} \end{pmatrix}$$

With these notations, the decision variable X of the optimization problem belongs to $\mathbb{R}^{k\times n}$, the linear space of $k \times n$ real matrices. Hence, we will assume that $\mathbb{R}^{k\times n}$ is equipped with the inner product,

$$\langle X, A\rangle = \text{ trace}(X^T A) = \sum_{i=1}^{n}\sum_{j=1}^{k} x_{ij}a_{ij}.$$

And the Frobenius norm on $\mathbb{R}^{k\times m}$ is given by

$$\|A\|_F = \sqrt{\langle A, A\rangle} = \sqrt{\sum_{j=1}^{k}\langle a^j, a^j\rangle} = \sqrt{\sum_{j=1}^{k}||a^j||^2} \tag{4.9}$$

To solve the smoothed problem,let us start by computing the gradient of the first part of the DC decomposition of g_γ in equation (4.8). Then it becomes

$$g_\gamma(x^1, x^2, ..., x^k) = \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}\|\frac{x^j - a^i}{\gamma}\|^2.$$

The partials of g_γ with respect to X can be written as

$$\begin{aligned} g_\gamma(x^1, x^2, ..., x^k) &:= \frac{(1+\tau)}{2\gamma}\sum_{i=1}^{m}\sum_{j=1}^{k}||x^j - a^i||^2 \\ &= \frac{(1+\tau)}{2\gamma}\sum_{i=1}^{m}\sum_{j=1}^{k}\left[\|x^j\|^2 - 2\langle x^j, a^i\rangle + \|a^i\|^2\right] \\ &= \frac{(1+\tau)}{2\gamma}\left[m\|X\|_F^2 - 2\langle X, E_{km}A\rangle + k\|A\|_F^2\right] \end{aligned}$$

where $E_{km} \in \mathbb{R}^{k\times m}$ is a matrix of all ones.. Then, $g_{1\gamma}$ is smooth and its partial gradients are:

$$\begin{aligned}\nabla_x g_\gamma(x^1,x^2,...,x^k) &= \frac{(1+\tau)}{\gamma}[mX - E_{km}A] \\ &= \frac{(1+\tau)}{\gamma}[mX - M], \text{where } M = E_{km}A.\end{aligned}$$

Next,we focus on $X \in \partial g^*(Y)$ where g^* is a Fenchel conjugate defined in (1.8) and can be calculated using the fact that $X \in \partial g^*(Y) \Leftrightarrow Y \in \partial g(X)$. Since g_γ is differentiable, it is expressed as

$$\nabla_x g_\gamma(x^1,x^2,...,x^k) = Y.$$

And now one can drive X from the following equation as

$$Y = \frac{(1+\tau)}{\gamma}[mX - M]$$

Now we can express X as

$$X = \frac{\gamma Y}{m(1+\tau)} + \frac{M}{m}.$$

To compute the sub-gradient of convex function h_γ in (4.9). First we express h_γ as sum of convex functions as in (4.10)

$$h_\gamma(x^1,x^2,...,x^k) = h_{1\gamma}(x^1,x^2,...,x^k) + h_{2\gamma}(x^1,x^2,...,x^k) + h_{3\gamma}(x^1,x^2,...,x^k) \tag{4.10}$$

where

$$\begin{aligned}h_{1\gamma}(x^1,x^2,...,x^k) &= \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}[d(\frac{x^j-a^i}{\gamma};F)]^2, \\ h_{2\gamma}(x^1,x^2,...,x^k) &= \sum_{i=1}^{m}\max_{t=1,...,k}\sum_{j=1,j\neq t}^{k}\|x^j-a^i\|_1, \\ h_{3\gamma}(x^1,x^2,...,x^k) &= \tau\sum_{j=1}^{k}\max_{r=1,...,m}\sum_{i=1,i\neq r}^{m}\|x^j-a^i\|_1.\end{aligned}$$

Since we use ℓ_1 norm, the sub-gradient $Y \in \partial h_\gamma(X)$ for the case where F is the closed unit box in $\mathbb{R}^n$ defined as $F = \{(v_1,...,v_n) \in \mathbb{R}^n \mid 1 \le v_k \le 1, k = 1,...,n\}$.
For a given $v \in \mathbb{R}$ we define

$$\text{sign(v)} = \begin{cases} 1 & if \quad v > 0, \\ 0 & if \quad v = 0, \\ -1 & if \quad v < 0, \end{cases}$$

the sub-gradient of $f(x) = \|x\|_1$ at $x \in \mathbb{R}^n$ are $s_i = \text{sign}(x)$ if $x_i \neq 0$ and $s_i \in [-1,1]$ if $x_i = 0$. Let us find the gradient of a smooth function h_1 in (4.10) with respect to X. As all function in sum is convex, sub-gradient of h_γ is computed using sub-differential sum and maximum rule, given in (B. Mordukhovich & Nam, 2014). Hence

$$h_{1\gamma} = \frac{(1+\tau)\gamma}{2} \sum_{i=1}^{m} \sum_{j=1}^{k} \left[d(\frac{x^j - a^i}{\gamma}; F) \right]^2.$$

Thus, the partial of $h_{1\gamma}$ at X and λ is given as:

$$\frac{\partial h_{1\gamma}}{\partial x^j}(x^1, x^2, ..., x^k) = (1+\tau) \sum_{i=1}^{m} \left[\frac{x^j - a^i}{\gamma} - P(\frac{x^j - a^i}{\gamma}; F) \right], \tag{4.11}$$

for all $j = 1, ..., k$. Note that, $\frac{\partial h_{1\gamma}}{\partial x^j}$ is a $k \times n$ matrix.

The projection in (4.11) is the Euclidean projection from $v \in \mathbb{R}^n$ onto a unit closed box F is defined as,

$$P(v, F) = \max(-e, \min(v, e)).$$

Where $e \in \mathbb{R}^n$ is a vector with one in each coordinate and zero elsewhere.
On the other hand, the third kinds $h_{2\gamma}$ and $h_{3\gamma}$, are non-smooth since we use ℓ_1 norm. Then, we can compute a sub-gradient for $h_{2\gamma}$ and $h_{2\gamma}$, based on the sub-differential formula for maximum functions, as demonstrated below. For instance, we consider

$$\begin{aligned} h_{2\gamma}(x^1, x^2, ..., x^k) &= \sum_{i=1}^{m} \max_{t=1,...,k} \sum_{j=1, j\neq t}^{k} ||x^j - a^i||_1 \\ &= \sum_{i=1}^{m} \max_{t=1,...,k} \left(\sum_{j=1}^{k} ||x^j - a^i||_1 - ||x^t - a^i||_1 \right) \end{aligned}$$

For $i = 1, ..., m$, select an index $t(i)$ so that

$$\max_{t=1,...,k} \left(\sum_{j=1}^{k} ||x^j - a^i||_1 - ||x^t - a^i||_1 \right) = \sum_{j=1}^{k} ||x^j - a^i||_1 - ||x^{t(i)} - a^i||_1$$

Now let $U = (u_{ji})$ be a signed block matrix with $u_{ji} = \text{sign}(x^j - a^i)$ is row vector and U^i be i^{th} column block matrix of U. Then $\partial h_{2\gamma}$ at X is

$$\frac{\partial h_{2\gamma}}{\partial x^j} := \sum_{i=1}^{m} \left(U^i - e_{t(i)} u_{t(i)i} \right)$$

for $e_{t(i)}$ a column vector of k coordinates with one at the $t(i)^{th}$ place and zero otherwise.

Similarly the sub-gradient of $h_{3\gamma}$ is given by

$$\begin{aligned} h_{3\gamma}(x^1,x^2,...,x^k) &= \tau\sum_{j=1}^{k}\max_{r=1,...,m}\sum_{i=1,i\neq r}^{m}||x^j-a^i||_1 \\ &= \sum_{j=1}^{k}\max_{r=1,...,m}\left(\sum_{i=1}^{m}||x^j-a^i||_1-||x^j-a^r||_1\right) \end{aligned}$$

for all $j=1,...,k$, select $r(j)$ so that

$$\max_{r=1,...,m}\left(\sum_{i=1}^{m}||x^j-a^i||_1-||x^j-a^r||_1\right)=\sum_{i=1}^{m}||x^j-a^i||_1-||x^j-a^{r(i)}||_1.$$

Let $S\in\mathbb{R}^{k\times n}$ be a matrix with j^{th} row $\sum_{i=1}^{m}s_{ji}-s_{jr(j)}$, then $\partial h_3\gamma$ at X is

$$\frac{\partial h_{3\gamma}}{\partial x^j}:=\tau S.$$

From the sub-gradient calculated above we have,

$$Y=\frac{\partial h_{1\gamma}}{\partial x^j}+\frac{\partial h_{2\gamma}}{\partial x^j}+\frac{\partial h_{3\gamma}}{\partial x^j}.$$

Now we have brought all necessary point such as gradient and sub-gradient to formulate the DCA algorithm that solve the problem as shown in Algorithm 3.

Algorithm 3: DCA algorithm for Clustering

Input : $A,X_0,\tau_0,\gamma_0,\sigma_1,\sigma_2,N\in\mathbb{N}$.

while *stopping is not reached* **do**

  for $t=1,\ldots,N$ **do**

    Search $Y_k\in\partial h_\gamma(X_{k-1})$

    $X_k\leftarrow\frac{\gamma Y_k}{m(1+\tau)}+\frac{M}{m}$

  end

end

Output: X_N

4.2 Simulation Results

The numerical simulation was done on a laptop with MATLAB software by considering artificial data. We used a continuous formulation of discrete problem. As a result it became non-smooth and non-convex, on which Nesterov's smoothing and DC-based algorithms were implemented.

During numerical simulation different parameters were used, among those, we used a large growing τ. In addition penalty and smoothing parameter are updated throughout the iteration

as follows : $\tau_{i+1} = \sigma_1 \tau_i, \sigma_1 > 1$, and $\gamma_{i+1} = \sigma_2 \gamma_i, \sigma_2 \in (0,1)$. We selected starting penalty parameter ($\tau_0 = e^{-6}$) and initial smoothing parameter $\gamma_0 = 1$. Besides, we used $\sigma_1 = 16e^9$ as penalty parameter's growth factor and $\sigma_2 = 0.75$ smoothing parameter's decay factor after testing with different values.
For the implementation of the algorithms, we used a randomly selected starting cluster centers (see Figure 4.1, 4.2 and 4.3).
Figure 4.1a, 4.1b, 4.1c and 4.1d shows different cluster assignment with the same centers and the same optimal cost is observed which indicates that the algorithm give choice for the decision maker. Since the algorithms are modified DCA, there is no guarantee that our algorithms converge to a global optimal solution. However, for small datasets the algorithm converges 100 % of the time to a global optimal solution (see Table 4.1, 4.2 and 4.3).

For the following tables, we conducted an experiment with fixed iteration numbers for each data set and initial cluster centers were randomly selected from the datasets and the same initial smoothing and penalty parameters works for the three datasets. The cost is obtained by minimizing equation (4.4).

Table 4.1: Ten iterations for the 15 points test data set and 3 clusters.

For $\gamma_0 = 0.5$, $\tau_0 = 10^{-6}$, $\sigma_1 = 1e+12$ and $\sigma_2 = 0.75$

Dataset	Cost	Time	Iteration	Centers	Data size
Test data	50.000000	1.637672	40	3	15
Test data	50.000000	1.911503	40	3	15
Test data	50.000000	1.398233	40	3	15
Test data	50.000000	1.305427	40	3	15
Test data	50.000000	1.454305	40	3	15
Test data	50.000000	1.404200	40	3	15
Test data	50.000000	1.321447	40	3	15
Test data	50.000000	1.727873	40	3	15
Test data	50.000000	1.519150	40	3	15
Test data	50.000000	1.476923	40	3	15

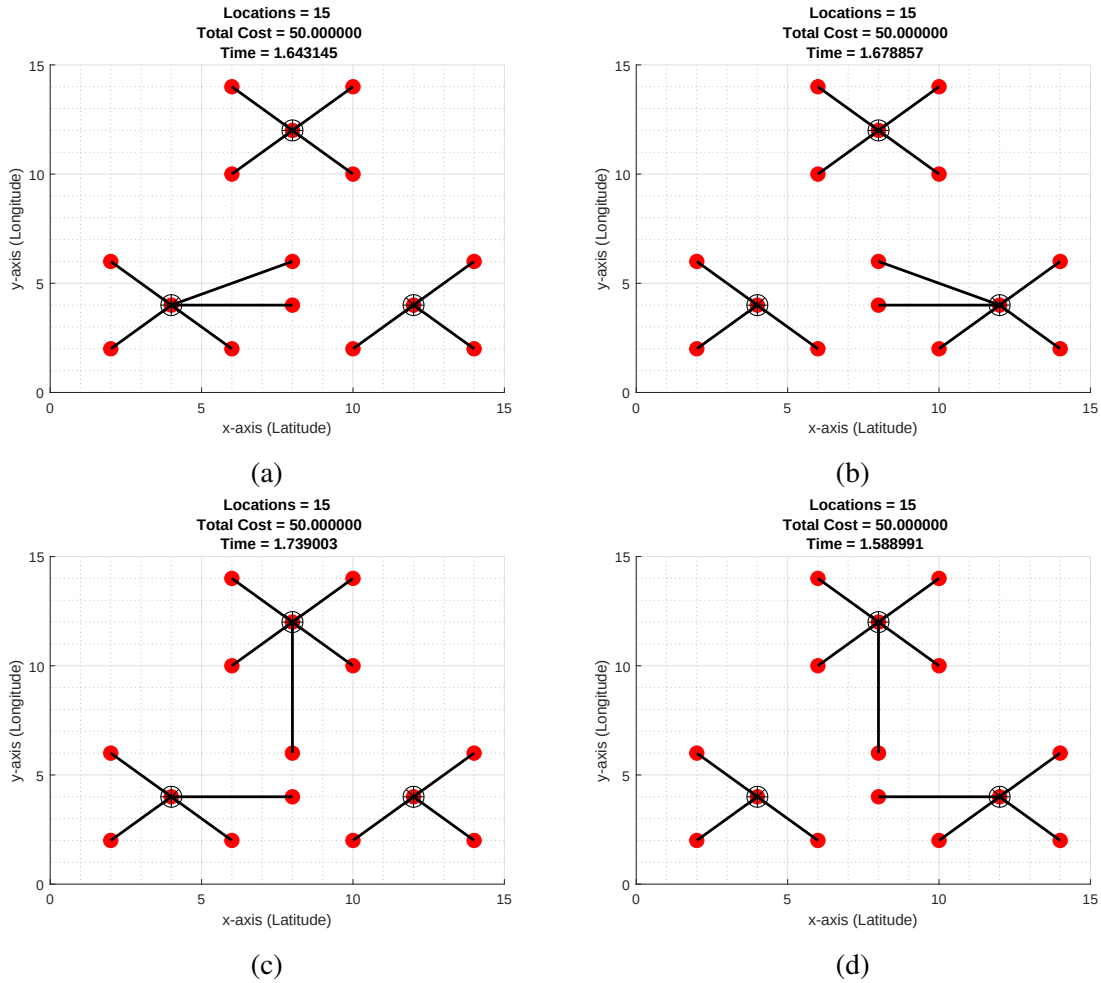


Figure 4.1: Multiple optimal solution of DCA Algorithm 3 , with 15 nodes and 3 clusters.

The optimal centers are

$$X = \begin{pmatrix} 8.0000 & 12.0000 \\ 4.0000 & 4.0000 \\ 12.0000 & 4.0000 \end{pmatrix}$$

Table 4.2: Ten iterations of 65 Oromia cities and towns in Ethiopia with 5 clusters.

For $\gamma_0 = 0.5$, $\tau_0 = 10^{-6}$, $\sigma_1 = 1e+12$ and $\sigma_2 = 0.75$

Dataset	Cost	Time	Iteration	Centers	Data size
A 65 points data	62.060009	5.598458	100	5	65
A 65 points data	62.060009	6.392423	100	5	65
A 65 points data	62.730003	5.920898	100	5	65
A 65 points data	62.730003	5.882201	100	5	65
A 65 points data	62.730003	5.679614	100	5	65
A 65 points data	62.730003	5.679614	100	5	65
A 65 points data	62.730003	5.170506	100	5	65
A 65 points data	62.730003	5.917282	100	5	65
A 65 points data	62.730003	5.598369	100	5	65
A 65 points data	62.730003	5.943033	100	5	65

Table 4.3: Ten iterations for a 1000 nodes artificial dataset with 5 clusters.

For $\gamma_0 = 0.5$, $\tau_0 = 10^{-6}$, $\sigma_1 = 1e+12$ and $\sigma_2 = 0.75$

Dataset	Cost	Time	Iteration	Centers	Data size
A 1000 nodes data	1989.530365	51.366254	200	5	1000
A 1000 nodes data	1950.954552	59.168751	200	5	1000
A 1000 nodes data	1950.954552	57.988467	200	5	1000
A 1000 nodes data	1950.954552	57.806795	200	5	1000
A 1000 nodes data	1950.954552	59.289988	200	5	1000
A 1000 nodes data	1950.748696	56.299496	200	5	1000
A 1000 nodes data	1950.948696	59.274801	200	5	1000
A 1000 nodes data	1950.618696	54.829517	200	5	1000
A 1000 nodes data	1950.948696	61.652984	200	5	1000
A 1000 nodes data	1954.476597	55.122954	200	5	1000

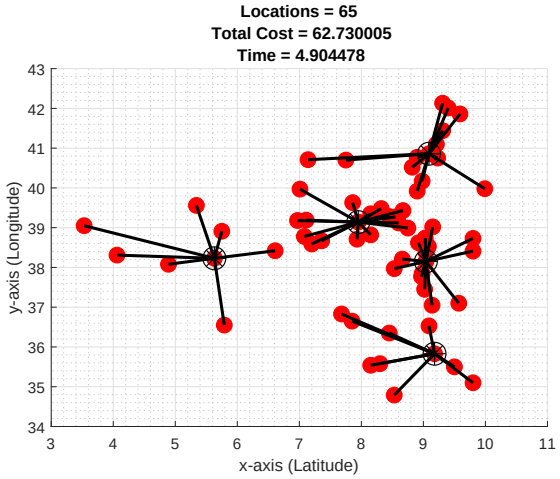


Figure 4.2: Result of DCA algorithm 3, 65 town in Oromia region of Ethiopia with 5 clusters.

The optimal centers are

$$X = \begin{pmatrix} 7.9500 & 39.1400 \\ 8.4500 & 36.3500 \\ 9.0900 & 40.8600 \\ 5.6300 & 38.2300 \\ 9.0400 & 38.3900 \end{pmatrix}$$

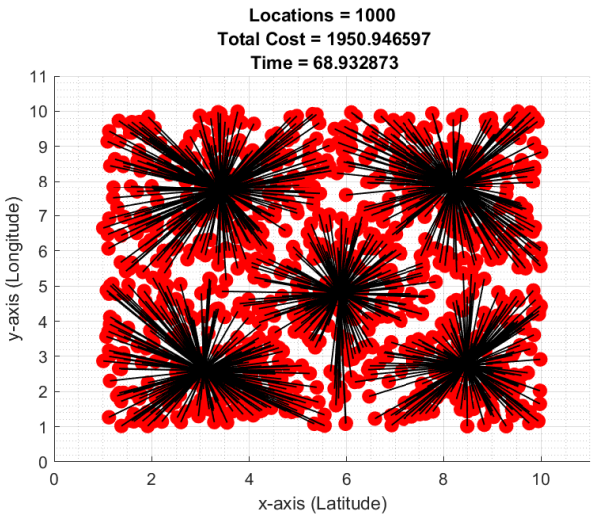


Figure 4.3: Optimal solution and CPU time of DCA algorithm 3, with 1000 nodes and 5 clusters.

The total cost $f(X) = 1954.401747$, with centers

$$X = \begin{pmatrix} 8.2370 & 7.8794 \\ 3.0467 & 2.7415 \\ 3.4140 & 7.7048 \\ 8.5211 & 2.7217 \\ 5.8614 & 4.7893 \end{pmatrix}$$

CHAPTER 5

HIERARCHICAL CLUSTERING MODEL I

5.1 Bi-level Hierarchical Clustering Model I

To define our problems, consider a set A of m data points, that is $A = \{a^i \in R^n : i = 1,\ldots,m\}$ and k variable cluster centers denoted by $x^1,\ldots,x^k$. We model a two-level hierarchical clustering problem by choosing k separate cluster centers from which one is the headquarter that serves the centers. Other members of the data will be assigned to one of the cluster based on the ℓ_1- norm between the data points and centers. Thus, nodes are grouped into k variable centers by minimizing the ℓ_1 distances from all node to k centers. Then a headquarter is a center that minimize the overall distance of the network and also serve as a cluster center.Then headquarter is defined to be mean of x^j for $j = 1,..,k$. i.e $\bar{x} = \frac{1}{k}\sum_{j=1}^{k} x^j$. This constraint limits the search region for headquarter to mean of selected centers or node near mean. Mathematically, the problem is defined as:

$$f(X) = \sum_{i=1}^{m} \min\left\{\|x^1 - a^i\|_1, ..., \|x^k - a^i\|_1, \|\bar{x} - a^i\|_1\right\} + \sum_{j=1}^{k} \|x^j - \bar{x}\|_1,$$

is minimized, where,

$$\bar{x} = \frac{1}{k}\sum_{j=1}^{k} x^j.$$

In addition, to insure the centers are real node the points $\bar{x}, x^1, x^2, ..., x^k$ should satisfy the following condition:

$$\min_{i=1,...,m} \|\bar{x} - a^i\|_1 + \sum_{j=1}^{k} \min_{i=1,...m} \|x^j - a^i\|_1 = 0.$$

Thus the problem is formulated as,

$$\text{Minimize} \quad \left(\sum_{i=1}^{m} \min_{j=1,...,k} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - \bar{x}\|_1\right) \tag{5.1}$$

$$\text{subject to} \quad \sum_{j=1}^{k+1} \min_{i=1,...m} \|x^j - a^i\|_1 = 0. \tag{5.2}$$

where x^{k+1} in the summation is $\bar{x}$. The constraints in (5.2) are used to force the centers to lie on real node and to force headquarter to be on or near mean of the centers based on minimum

distance.

We can write (5.1) as unconstrained problem using penalty parameter $\tau > 0$, as follows

$$\text{minimize} \left(\sum_{i=1}^{m} \min_{j=1,\ldots,k+1} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - \bar{x}\|_1 + \tau \sum_{i=1}^{k+1} \min_{i=1,\ldots,m} \|x^j - a^i\|_1 \right).$$

The problem in this chapter is non-smooth and non-convex, and it can be handled using smoothing techniques and the DCA, much like the clustering problem in chapter 4. A specific instance of this model in two dimensions was examined in (A. Bagirov et al., 2003), whereby the derivative free (A. Bagirov, 1999) method was employed to solve the problem. However, this approach is not appropriate for large-scale settings in high dimensions. These models, in which the squared Euclidean distance defines the similarity measure, were successfully solved by the DCA in (L. An & Minh, 2007). Despite offering a viable solution for bi-level hierarchical clustering in high dimensions, the DCA in (L. An & Minh, 2007) has not been applied to solve the initial model specified by the inappropriate Euclidean distance measurement proposed in (A. Bagirov et al., 2003) which is not suitable for the resulting DCA according to the authors of (L. An & Minh, 2007). Nevertheless, we will show in what follows that the DCA is applicable to this model when combined with Nesterov's smoothing technique.

Writing (5.3) as the sum and Maximum of convex functions using the formula in (1.25) as follows:

$$\begin{aligned} f(X) = \sum_{i=1}^{m}\sum_{j=1}^{k+1} \|x^j - a^i\|_1 - \sum_{i=1}^{m} \max_{t=1,\ldots,k+1} \sum_{j=1, j\neq t}^{k+1} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - \bar{x}\|_1 \\ + \tau \sum_{i=1}^{m}\sum_{j=1}^{k+1} \|x^j - a^i\|_1 - \tau \max_{t=1,\ldots,k+1} \sum_{j=1, j\neq t}^{k+1} \|x^j - a^i\|_1. \end{aligned} \tag{5.3}$$

Expressing (5.3) as DC function, we have

$$\begin{aligned} f(X) &= (1+\tau) \sum_{i=1}^{m}\sum_{j=1}^{k+1} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - \bar{x}\|_1 \\ &\quad - \sum_{i=1}^{m} \max_{t=1,\ldots,k+1} \sum_{j=1, j\neq t}^{k+1} \|x^j - a^i\|_1 - \tau \sum_{j=1}^{k+1} \max_{r=1,\ldots,m} \sum_{i=1, i\neq r}^{m} \|x^j - a^i\|_1, \end{aligned} \tag{5.4}$$

where

$$g(X) = (1+\tau) \sum_{i=1}^{m}\sum_{j=1}^{k+1} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - \bar{x}\|_1, \text{ and}$$

$$h(X) = \sum_{i=1}^{m} \max_{t=1,\ldots,k+1} \sum_{j=1,j\neq t}^{k+1} \|x^j - a^i\|_1 + \tau \sum_{j=1}^{k+1} \max_{r=1,\ldots,m} \sum_{i=1,i\neq r}^{m} \|x^j - a^i\|_1. \tag{5.5}$$

Since f is DC function based on Proposition 1.7.19 and ℓ_1 smoothing studied in (Mau Nam et al., 2020), we obtain a Nesterov's approximation of $\|x-a\|_1$ as:

$$\|x-a\|_1 := \frac{\gamma}{2}\left[\|\frac{x-a}{\gamma}\|^2 - [d(\frac{x-a}{\gamma};F)]^2\right].$$

The main goal is to minimize the partially smoothed objective given by,

$$\begin{aligned} f_\gamma(X) = {} & \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k+1}\|\frac{x^j-a^i}{\gamma}\|^2 + \sum_{j=1}^{k}\|\frac{x^j-\bar{x}}{\gamma}\|^2 \\ & -\frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k+1}[d(\frac{x^j-a^i}{\gamma};F))]^2 - \frac{\gamma}{2}\sum_{j=1}^{k}[d(\frac{x^j-\bar{x}}{\gamma};F))]^2 \\ & -\sum_{i=1}^{m}\max_{t=1,\ldots,k+1}\sum_{j=1,j\neq t}^{k+1}\|x^j-a^i\|_1 \\ & -\tau\sum_{j=1}^{k+1}\max_{r=1,\ldots,m}\sum_{i=1,i\neq r}^{m}\|x^j-a^i\|_1. \end{aligned} \tag{5.6}$$

That is minimize $\left\{f_\gamma(X) = g_\gamma(X) - h_\gamma(X)\right\}, X \in \mathbb{R}^{k\times n}$.
In addition, g_γ is the sum of convex functions defined as:

$$g_\gamma(X) = g_{1\gamma}(X) + g_{2\gamma}(X) \tag{5.7}$$

where

$$g_{1\gamma}(X) = \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k+1}\|\frac{x^j-a^i}{\gamma}\|^2, \;\; g_{2\gamma}(X) = \sum_{j=1}^{k}\|\frac{x^j-\bar{x}}{\gamma}\|^2.$$

And h_γ is also the sum of four convex functions defined as:

$$h_\gamma(X) = h_{1\gamma}(X) + h_{2\gamma}(X) + h_{3\gamma}(X) + h_{4\gamma}(X), \tag{5.8}$$

where

$$h_{1\gamma}(X) = \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k+1}[d(\frac{x^j-a^i}{\gamma};F))]^2,\ \ h_{2\gamma}(X)=\frac{\gamma}{2}\sum_{j=1}^{k}[d(\frac{x^j-\bar{x}}{\gamma};F))]^2,$$

$$h_{3\gamma}(X) = \sum_{i=1}^{m}\max_{t=1,\ldots,k+1}\sum_{j=1,j\neq t}^{k+1}\|x^j-a^i\|_1,\ \ h_{4\gamma}(X)=\tau\sum_{j=1}^{k+1}\max_{r=1,\ldots,m}\sum_{i=1,i\neq r}^{m}\|x^j-a^i\|_1.$$

For the calculation of gradient and sub-gradient consider a data matrix A with a^i , $i=1,...,m$, in the i^{th} row and a variable matrix X with $x^j, j=1,2,...,k+1$ in the j^{th} row.

Since X and A belongs to a linear space of real matrices, we can apply inner product such that

$$\langle X,A\rangle = \text{ trace}(X^TA)=\sum_{i=1}^{n}\sum_{j=1}^{k}x_{ij}a_{ij}.$$

And the Frobenius norm on $\mathbb{R}^{k\times m}$ is given by

$$\|A\|_F=\sqrt{\langle A,A\rangle}=\sqrt{\sum_{j=1}^{k}\langle a^j,a^j\rangle}=\sqrt{\sum_{j=1}^{k}\|a^j\|^2}. \tag{5.9}$$

To calculate the gradient of g_γ in (5.7), let X be of size $(k+1)\times n$ is variable matrix. Then

$$\begin{aligned} g_{1\gamma}(X) &:= \frac{(1+\tau)}{2\gamma}\sum_{i=1}^{m}\sum_{j=1}^{k+1}\|x^j-a^i\|^2, \\ &= \frac{(1+\tau)}{2\gamma}\sum_{i=1}^{m}\sum_{j=1}^{k}\left[\|x^j\|^2-2\langle x^j,a^i\rangle+\|a^i\|^2\right], \\ &= \frac{(1+\tau)}{2\gamma}\left[m\|X\|_F^2-2\langle X,E_{km}A\rangle+k\|A\|_F^2\right], \end{aligned}$$

where $E_{km}\in\mathbb{R}^{k+1\times m}$ is a matrix of all ones. As $g_{1\gamma}$ is smooth, then

$$\nabla_x g_{1\gamma}(X)=\frac{(1+\tau)}{\gamma}\left[mX-B\right],\ \ \text{where}\ \ B=E_{km}A.$$

Again consider $g_{2\gamma}$ which is differentiable function,

$$\begin{aligned} g_{2\gamma}(X) &:= \frac{1}{2\gamma}\sum_{j=1}^{k}\|x^j-\bar{x}\|^2, \\ &= \frac{1}{2\gamma}\sum_{j=1}^{k}\left[\|x^j\|^2-2\langle x^j,\bar{x}\rangle+\langle x^j,\bar{x}\rangle\right], \\ &= \frac{1}{2\gamma}\left[\|X\|_F^2-\frac{2}{k}\langle X,E_{kk}X\rangle+\frac{1}{k}\langle X,E_{kk}X\rangle\right], \end{aligned}$$

where E_{kk} is a $k\times k$ matrix with elements all ones. Then, the gradients of $g_{2\gamma}$ are given by:

$$\begin{aligned} \nabla_x g_{2\gamma}(X) &= \frac{1}{\gamma}\left[X-\frac{1}{k}E_{kk}X\right], \\ &= \frac{1}{\gamma}\left[X-HX\right], \text{where } H=\frac{1}{k}E_{kk}. \end{aligned}$$

Next,we focus on $X\in\partial g^*(Y)$ where g^* is a Fenchel conjugate defined in (1.8) and can be calculated using the fact that $X\in\partial g^*(Y)\Leftrightarrow Y\in\partial g(X)$. Since g_γ is differentiable. Thus,

$$\begin{aligned} \nabla_x g_\gamma(X) &= \nabla_x g_{1\gamma}(X)+\nabla_x g_{2\gamma}(X), \\ Y &= \frac{(1+\tau)}{\gamma}\left[mX-B\right]+\frac{1}{\gamma}\left[X-HX\right], \\ &= \left[\frac{(1+\tau)}{\gamma}m+\frac{1}{\gamma}\left[\mathbb{I}-H\right]\right]X-\left[\frac{(1+\tau)}{\gamma}B\right], \\ &= \frac{1}{\gamma}\left[(1+\tau)m+\mathbb{I}-H\right]X-\left[\frac{(1+\tau)}{\gamma}B\right], \\ &= \frac{1}{\gamma}\left[(1+\tau)m+1\right]\mathbb{I}-H\right]X-\left[\frac{(1+\tau)}{\gamma}B\right], \\ &= \frac{1}{\gamma}\left[a\mathbb{I}-bH\right]X-\left[\frac{(1+\tau)}{\gamma}B\right], \end{aligned}$$

where: $a=(1+\tau)m+1$ and $b=1$.

Let $N=a\mathbb{I}-bH$ then, N is invertible as $N^{-1}=\alpha\mathbb{I}+\beta H$ where

$$\alpha=\frac{1}{a}=\frac{1}{(1+\tau)m+1} \text{ and } \beta=\frac{b}{a[a+bk]}=\frac{1}{(1+\tau)m+1[(1+\tau)m+1+k]},$$

(see Lemma 5.1 of (Geremew et al., 2018)). Therefore,

$$X=\left[\alpha\mathbb{I}-\beta H\right]\left[\gamma Y_x+(1+\tau)B\right]. \tag{5.10}$$

Next we find the sub-gradient in (5.8), this can be done by search $Y\in\partial h_\gamma(X)$. Given a smooth functions $h_{1\gamma}$, and $h_{2\gamma}$ the partial gradient at x^j for $j=1,\ldots,k+1$ are

$$h_{1\gamma} = \frac{(1+\tau)\gamma}{2} \sum_{i=1}^{m} \sum_{j=1}^{k+1} \left[d(\frac{x^j - a^i}{\gamma}; F)) \right]^2$$

$$\frac{\partial h_{1\gamma}}{\partial x^j}(X) = (1+\tau)\gamma \sum_{i=1}^{m} \left[\frac{x^j - a^i}{\gamma} - P(\frac{x^j - a^i}{\gamma}; F) \right]. \tag{5.11}$$

Thus, $\nabla h_{1\gamma}(X))$ is a matrix with dimension $(k+1) \times n$ with j^{th} row is $\frac{\partial h_{1\gamma}}{\partial x^j}(X)$.
The gradients of $h_{2\gamma} = \frac{\gamma}{2} \sum_{j=1}^{k} \left[d(\frac{x^j - \bar{x}}{\gamma}; F)) \right]^2$ at X is given by:

$$\frac{\partial h_{2\gamma}}{\partial x^j}(X) = \frac{x^j - \bar{x}}{\gamma} - P(\frac{x^j - \bar{x}}{\gamma}; F) - \frac{1}{k} \sum_{\ell=1}^{k} \left[\frac{x^\ell - \bar{x}}{\gamma} - P(\frac{x^\ell - \bar{x}}{\gamma}; F) \right]. \tag{5.12}$$

The projection in (5.11) and (5.12) are the Euclidean projection from $v \in \mathbb{R}^n$ onto a unit closed box F is defined as,

$$P(v, F) = \max(-e, \min(v, e)).$$

Where $e \in \mathbb{R}^n$ is a vector with one in each coordinate and zero elsewhere.
Since we use ℓ_1 norm, next we illustrate how to find the sub-gradient $Y \in \partial h_\gamma(X)$ for the case where F is the closed unit box in $\mathbb{R}^n$.
For a given $x \in \mathbb{R}$ we define

$$\text{sign}(x) := \begin{cases} 1 & if \quad x > 0, \\ 0 & if \quad x = 0, \\ -1 & if \quad x < 0. \end{cases}$$

Then we define $\text{sign}(x) := (\text{sign}(x_1), \ldots, \text{sign}(x_n))$ for $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Note that the sub-gradient of $f(x) = \|x\|_1$ at $x \in \mathbb{R}^n$ are $s_i = \text{sign}(x)$ if $x_i \neq 0$ and $s_i \in [-1, 1]$ if $x_i = 0$.
The sub-gradients of the non-smooth functions $h_{3\gamma}$ and $h_{4\gamma}$ are calculated as the sub-differential of point-wise maximum functions,

$$h_{3\gamma} := \sum_{i=1}^{m} \max_{r=1,\ldots,k} \sum_{j=1, j\neq r}^{k} \|x^j - a^i\|_1 = \sum_{i=1}^{m} \phi_i(\mathbf{X}),$$

where, for $i = 1, \ldots, m$,

$$\phi_i(\mathbf{X}) := \max \left\{ \phi_{ir}(\mathbf{X}) = \sum_{j=1, j\neq r}^{k} \|x^j - a^i\|_1, \;\; r = 1, \ldots, k \right\}.$$

To do this, for each $i = 1, \ldots, m$, we first find $\mathbf{U}_i \in \partial \phi_i(\mathbf{X})$ according to Lemma 1.7.8. Then

we define $\mathbf{U} := \sum_{i=1}^{m} \mathbf{U}_i$ to get a sub-gradient of the function $h_{3\gamma}$ at $\mathbf{X}$ by the sub-differential sum rule. To accomplish this goal, we first choose an index r^* from the index set $\{1,\ldots,k\}$ such that

$$\phi_i(\mathbf{X}) = \phi_{ir^*}(\mathbf{X}) = \sum_{j=1, j\neq r^*}^{k} \|x^j - a^i\|_1.$$

Using the familiar sub-differential formula of the ℓ_1 norm function, the j^{th} row u_i^j for $j \neq r^*$ of the matrix $\mathbf{U}_i$ is determined as follows

$$u_i^j := \text{sign}(x^j - a^i) = \begin{cases} 1 & \text{if } x^j > a^i, \\ 0 & \text{if } x^j = a^i, \\ -1 & \text{if } x^j < a^i. \end{cases}$$

The r^{*th} row of the matrix $\mathbf{U}_i$ is $u_i^{r^*} := 0$.
Similarly the sub-gradient of $h_{4\gamma}$ is given by

$$h_{4\gamma} \;:=\; \tau \sum_{j=1}^{k} \max_{s=1,\ldots,m} \sum_{i=1, i\neq s}^{m} \|x^j - a^i\|_1 = \tau \sum_{j=1}^{k} \psi_j(\mathbf{X}),$$

where, for $j = 1,\ldots,k$,

$$\psi_j(\mathbf{X}) := \max\left\{ \psi_{js}(\mathbf{X}) = \sum_{i=1, i\neq s}^{m} \|x^j - a^i\|_1,\;\; s = 1,\ldots,m \right\}.$$

To do this, for each $j = 1,\ldots,k$, we first find $\mathbf{W}_j \in \partial\psi_j(\mathbf{X})$. Then we define $\mathbf{W} := \tau \sum_{j=1}^{k} \mathbf{W}_j$ to get a sub-gradient of the function $h_{4\gamma}$ at $\mathbf{X}$ by the sub-differential sum rule. To accomplish this goal, we first choose an index s^* from the index set $\{1,\ldots,m\}$ such that

$$\psi_j(\mathbf{X}) = \psi_{js^*}(\mathbf{X}) = \sum_{j=1, j\neq s^*}^{k} \|x^j - a^i\|_1.$$

The s^{*th} row of the matrix $\mathbf{W}_j$ is $w_{s^*}^j := 0$.
Thus the sub-gradient of $h_{4\gamma}$ is defined as

$$\frac{\partial h_{4\gamma}}{\partial x^j} := \tau \mathbf{W}.$$

From the sub-gradient calculated above we have,

$$Y = \frac{\partial h_{1\gamma}}{\partial x^j}(X) + \frac{\partial h_{2\gamma}}{\partial x^j}(X) + \frac{\partial h_{3\gamma}}{\partial x^j}(X) + \frac{\partial h_{4\gamma}}{\partial x^j}(X) \tag{5.13}$$

Now, we have in position to implement DCA algorithm that will solve the problem as shown

in DCA algorithm 4.

Algorithm 4: Bi-level Hierarchical Clustering Model I

Input : $A, X_0, \tau_0, \gamma_0, N \in \mathbb{N}$,

while *stopping criteria for* $(\gamma, \tau, \varepsilon)$ *not reached* **do**

$\alpha \leftarrow \frac{1}{(1+\tau_1)m+1}$,

$\beta \leftarrow \frac{1}{(1+\tau_1)m+1[(1+\tau_1)m+1+k]}$,

for $k = 1, \ldots, N$ **do**

Find $Y_k \in \partial h(x_{k-1})$, (5.13)

$X_k \leftarrow [\alpha\mathbb{I} + \beta H]\,[\gamma Y_k + (1+\tau)B]$, (5.10)

end

update γ and τ,

end

Output: X_N.

5.2 Simulation Results

The numerical simulation was performed on an HP laptop with an Intel(R) Core(TM) i7-8565U @ 1.80 GHz 1.99 GHz processor, 8.00 GB RAM with MATLAB version R2017b. Various parameters were used during the simulation, among others we used a large increasing penalty parameter τ and a decay smoothing parameter γ . These parameters are updated during iteration as: $\tau_{i+1} = \sigma_1 \tau_i, \sigma_1 > 1$ and $\gamma_{i+1} = \sigma_2 \gamma_i$, $0 < \sigma_2 < 1$ and $\varepsilon = 1e-6$. We chose the initial penalty parameter ($\tau_0 = e^{-4}$) and the initial smoothing parameter $\gamma_0 = 1$. In addition, after varying the parameters, we chose $\sigma_1 \leq 16e^9$ as the growth factor of the penalty parameter, $\sigma_2 = 0.5$ the decrease factor of the smoothing parameter, and the stopping criterion $\frac{\|X_{k+1}-X_k\|_F}{\|X_k\|_F+1} \leq \varepsilon$ for inner for loop. To implement the algorithms, we used randomly selected random cluster centers from the datasets.

The performance of the DCA algorithm 4 was tested with different data sets. We first tested the algorithm on a small dataset taken from (Geremew et al., 2018) and the result shows that it converges to the same cluster centers with a different objective value due to the ℓ_1 norm. Since the ℓ_1 distance is greater than or equal to the Euclidean distance, it depends on the data points. As shown in Table 5.1, the algorithm converges to the optimal point about 85.71%. This means that out of 7 iterations, 6 of them converge to the same objective valve.

Second, we tested the proposed algorithm with EIL76 (The 76 City Problem) data sets taken from (Reinelt, 1991) with four clusters, one of which serves as HQ, which converge to near-optimal cluster centers in a reasonable time compared to work (Geremew et al., 2018; Nam et al., 2018) (see figure 5.2).

It is also observed in the EIL76 (The 76 City Problem) data which converge to the same or close cluster centers with higher objective cost, fewer iterations and almost the same time compared to the work of (Geremew et al., 2018) iterated using MATLAB, (see table 5.2).

Third, we applied our proposed algorithm to a GPS data from 142 cities and towns in Ethiopia with more than 7,000 inhabitants, including 65 in Oromia regional state. We tested the algorithm with 65 nodes, 4 cluster centers, one of which serves as HQ, and 142 nodes with six clusters (see Figure 5.3 and Figure 5.4), which converge 86% to the optimal solution. This means that out of 7 iterations, 6 of them converge to the near-optimal values shown in Table 5.4 and 5.5.

Fourth, we tested the proposed algorithm with PR1002 (The 1002 City Problem) data sets taken from (Reinelt, 1991) with seven clusters, one of which serves as HQ, which converge to near-optimal cluster centers in a reasonable time compared to work (Geremew et al., 2018; Nam et al., 2018) (see Table 5.3 and figure 5.5).

To show how the objective functions improved with iteration, we include a plot of the first few iterations of Figure 5.4 and Figure 5.5, which shows the dynamics of the algorithm (see Figure 5.4a & 5.4b figure 5.5a & 5.5b).

In general, since the algorithms is a modified DCA and DCA is a local search algorithm, there is no guarantee that our algorithms converge to the global optimal solution. However, we compared our result with ℓ_2 norm in (Geremew et al., 2018; Nam et al., 2018) and it shows that our proposed algorithm converges with fewer iteration but relatively the same computational time for data iterated with MATLAB in (Geremew et al., 2018). In addition, we compared our result with Brute-force generated solutions for datasets with fewer nodes (see Figure 5.2a & 5.2b and Figure 5.3a & 5.3b) which converge to a near-optimal value with reasonable time compared to the brute-force iterations.

We expect that our method used in this section to solve the two-level clustering problem with the ℓ_1 norm is less sensitive to outliers compared to the ℓ_2 norm, which minimizes possible clustering errors. In addition, it can be used to solve other non-smooth and non-convex optimization problems in signal processing, such as image pixel clustering for image segmentation and compressed sensing.

After a number of experiment, by trial and error we verify that the values chosen for smoothing and penalty parameter determine the performance of the algorithm for each data set. It can be seen that very small values of the smoothing parameter may prevent the algorithm from clustering, and thus we gradually decrease these values of smoothing and gradually increase the penalty parameters.

For the following tables, we conducted an experiment with fixed iteration numbers for each data set and initial cluster centers were randomly selected from the datasets. The cost is obtained by minimizing equation (6.1).

Table 5.1: Ten iterations for the 15 points test data set.

For $\gamma_0 = 1$, $\tau_0 = 10^{-6}$, $\sigma_1 = 16e+9$ and $\sigma_2 = 0.75$

Dataset	Cost	Time	Iteration	Centers	Data size
Test data	19.142400	1.101548	80	3	15
Test data	19.189316	1.980110	80	3	15
Test data	19.232334	1.839473	80	3	15
Test data	19.232334	1.839473	80	3	15
Test data	19.213091	1.807942	80	3	15
Test data	19.210684	1.816595	80	3	15
Test data	20.904108	2.005234	80	3	15
Test data	19.199307	1.617948	80	3	15
Test data	19.191683	1.880256	80	3	15
Test data	19.190641	1.795777	80	3	15

Table 5.2: Ten iterations for EIL76 data set.

For $\gamma_0 = 1$, $\tau_0 = 10^{-6}$, $\sigma_1 = 16e+9$ and $\sigma_2 = 0.5$

Dataset	Cost	Time	Iteration	Centers	Data size
EIL76	1370.002095	4.966008	300	4	76
EIL76	1360.029069	6.577355	300	4	76
EIL76	1360.034103	5.752709	300	4	76
EIL76	1360.034103	5.789645	300	4	76
EIL76	1360.034103	6.184165	300	4	76
EIL76	1362.015115	11.135013	300	4	76
EIL76	1360.040060	6.680976	300	4	76
EIL76	1360.034103	7.120168	300	4	76
EIL76	1360.034103	5.275355	300	4	76
EIL76	1360.034103	6.191321	300	4	76

Table 5.3: Ten iterations for PR1002 data set.

For $\gamma_0 = 1600$, $\tau_0 = 10^{-6}$, $\sigma_1 = 8000$ and $\sigma_2 = 0.5$

Dataset	Cost	Time	Iteration	Centers	Data size
PR1002	2.037468e+6	55.747814	350	7	1002
PR1002	2.036560e+6	49.831589	350	7	1002
PR1002	2.036560e+6	47.624006	350	7	1002
PR1002	2.034309e+6	56.673023	350	7	1002
PR1002	2.034309e+6	62.799942	350	7	1002
PR1002	2.034309e+6	58.704935	350	7	1002
PR1002	2.034309e+6	61.162610	350	7	1002
PR1002	2.033309e+6	65.173468	350	7	1002
PR1002	2.033309e+6	65.183165	350	7	1002
PR1002	2.035309e+6	56.747571	350	7	1002

Table 5.4: Ten iterations for the 65 points test data set.

For $\gamma_0 = 1$, $\tau_0 = 10^{-6}$, $\sigma_1 = 16e+9$ and $\sigma_2 = 0.75$

Dataset	Cost	Time	Iteration	Centers	Data size
A 65 points data	87.689552	7.639907	250	4	65
A 65 points data	87.889552	7.655031	250	4	65
A 65 points data	89.889552	7.542318	250	4	65
A 65 points data	87.889552	7.520294	250	4	65
A 65 points data	87.389552	8.101386	250	4	65
A 65 points data	87.689552	7.324263	250	4	65
A 65 points data	87.789552	7.289665	250	4	65
A 65 points data	87.789552	7.545756	250	4	65
A 65 points data	87.389552	7.399864	250	4	65
A 65 points data	87.089552	7.460620	250	4	65

Table 5.5: Ten iterations for the 142 points test data set.

For $\gamma_0 = 1$, $\tau_0 = 10^{-6}$, $\sigma_1 = 16e+9$ and $\sigma_2 = 0.5$

Dataset	Cost	Time	Iteration	Centers	Data size
A 142 points data	233.793443	13.211982	300	6	142
A 142 points data	233.793443	13.967717	300	6	142
A 142 points data	233.793443	13.197907	300	6	142
A 142 points data	233.793443	13.857134	300	6	142
A 142 points data	233.893443	15.259091	300	6	142
A 142 points data	233.893443	12.552241	300	6	142
A 142 points data	241.793443	13.426098	300	6	142
A 142 points data	233.783342	12.750388	300	6	142
A 142 points data	233.783241	10.986166	300	6	142
A 142 points data	233.783242	12.123523	300	6	142

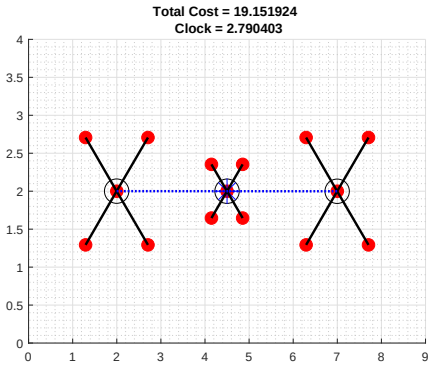
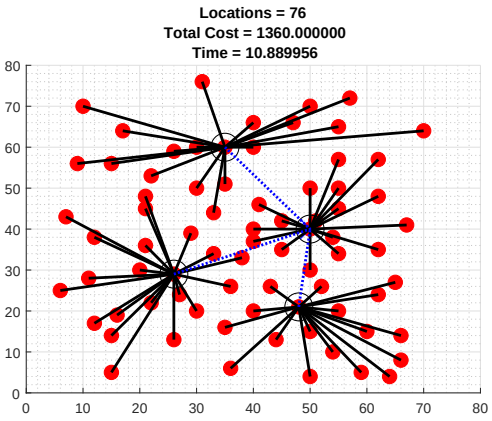
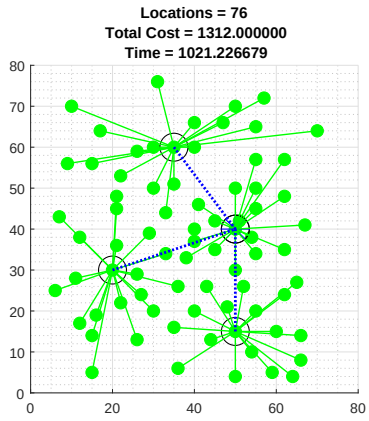


Figure 5.1: A 15 points test data set with 3 clusters and one serves as HQ

$$X = \begin{pmatrix} 7.0000 & 2.0000 \\ 4.5000 & 2.0000 \\ 2.0000 & 2.0000 \end{pmatrix}, \quad HQ = \begin{pmatrix} 4.5000 & 2.0000 \end{pmatrix}$$



(a) A EIL76 data with Algorithm 4

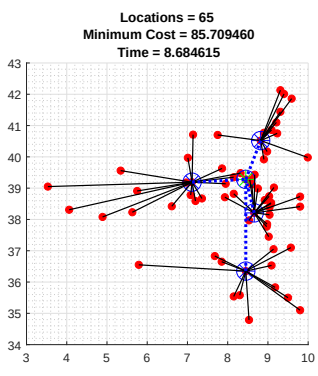


(b) A EIL76 data with brute-force iteration

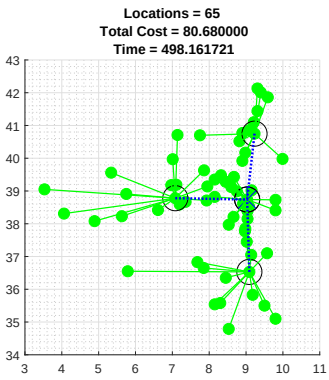
Figure 5.2: EIL76 (The 76 City Problem) data taken from (Reinelt, 1991) with 4 clusters one serves as HQ.

The optimal cluster centers and HQ using Algorithm 4 are,

$$X = \begin{pmatrix} 26.0000 & 29.0000 \\ 35.0000 & 60.0000 \\ 50.0000 & 40.0000 \\ 48.0000 & 21.0000 \end{pmatrix} \quad HQ = (50.0000 \quad 40.0000),$$



(a) A 65 data points using Algorithm 4

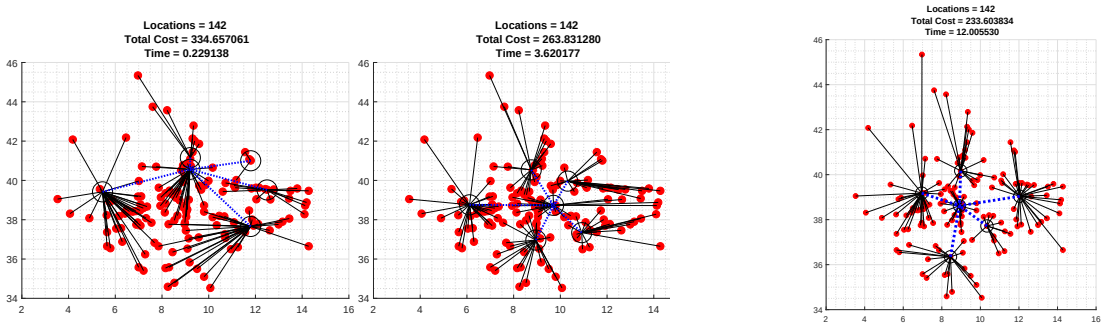


(b) A 65 data points with Brute force iteration

Figure 5.3: A 65 Oromia regional cities and towns with 4 clusters one serves as HQ

The selected centers and HQ using Algorithm 4 are

$$X = \begin{pmatrix} 8.4500 & 36.3500 \\ 8.8990 & 39.9171 \\ 8.6607 & 38.2124 \\ 6.6100 & 38.4200 \end{pmatrix} \text{ and } \quad HQ = (8.6607 \quad 38.2124).$$

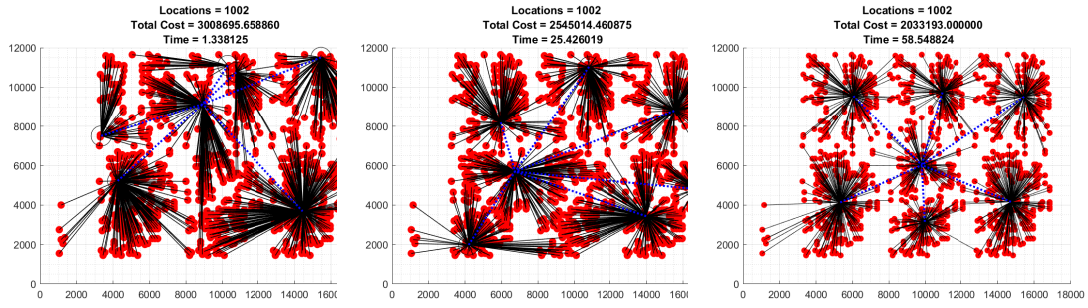


(a) Beginning of iterations (b) Few iterations (c) Optimal iterations

Figure 5.4: A 142 Ethiopian towns and cities with 6 clusters one serves as HQ.

The optimal selected centers and HQ with Algorithm 4 are,

$$X = \begin{pmatrix} 10.3400 & 37.7199 \\ 8.4500 & 36.3500 \\ 6.9595 & 39.1795 \\ 8.9131 & 38.6186 \\ 8.9808 & 40.1709 \\ 12.0400 & 39.0400 \end{pmatrix} \text{ and } \quad HQ = (8.9156 \quad 38.6189).$$



(a) Beginning of iteration (b) Few iteration (c) Optimal iteration

Figure 5.5: PR1002 (The 1002 City Problem) taken from (Reinelt, 1991) with 7 clusters and one serving as HQ

.

For this particular dataset we used $\gamma_0 = 1600$ and $\sigma_1 = 8000$. The optimal cluster centers

and HQ selected using Algorithm 4 are,

$$X = \begin{pmatrix} 5218.0000 & 4090.0000 \\ 5923.0000 & 9557.0000 \\ 1083.9000 & 9857.0000 \\ 1473.5000 & 4145.0000 \\ 9977.0000 & 3008.0000 \\ 1547.1000 & 9522.0000 \\ 9892.0000 & 6023.0000 \end{pmatrix} \text{ and } \quad HQ = (9891.6000 \quad 6023.0000).$$

CHAPTER 6

HIERARCHICAL CLUSTERING MODEL II

6.1 Bi-level Hierarchical Clustering Model IIa

Let $A = \{a^1, ..., a^m\} \subset \mathbb{R}^n$ be the set of nodes, and let $X = \{x^1, ..., x^k\} \subset \mathbb{R}^n$ be the set of free cluster centers (i.e., not necessarily from the existing nodes).Then we define bi-level hierarchical clustering model as

$$f(X) = \sum_{i=1}^{m} \min_{j=1,...,k} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - \bar{x}\|_1$$

is minimized, where,

$$\bar{x} = \frac{1}{k} \sum_{j=1}^{k} x^j$$

In addition, to insure the centers are real node the points $\bar{x}, x^1, x^2, ..., x^k$ should satisfy the following condition:

$$\sum_{j=1}^{k} \min_{i=1,...m} \|x^j - a^i\|_1 = 0.$$

Thus the problem is formulated as:

$$\text{Minimize} \left\{ \sum_{i=1}^{m} \min_{j=1,...,k} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - \bar{x}\|_1 \right\} \tag{6.1}$$

subject to

$$\sum_{j=1}^{k} \min_{i=1,...m} \|x^j - a^i\|_1 = 0, \tag{6.2}$$

where x^k in the summation is $\bar{x}$ and the constraints in (6.2) are used to force the centers to lie on real node and to force headquarter to be on or near mean of the centers based on minimum distance.

We can write (6.1) as unconstrained problem using penalty parameter $\tau > 0$, as follows

$$\text{Minimize} \sum_{i=1}^{m} \min_{j=1,...,k} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - \bar{x}\|_1 + \tau \sum_{i=1}^{k} \min_{i=1,...,m} \|x^j - a^i\|_1. \tag{6.3}$$

Writing (6.3) as the sum and Maximum of convex functions using the formula in (1.25) as follows:

$$f(X) = \sum_{i=1}^{m}\sum_{j=1}^{k}\|x^j - a^i\|_1 - \sum_{i=1}^{m}\max_{t=1,\ldots,k}\sum_{j=1,j\neq t}^{k}\|x^j - a^i\|_1 + \sum_{j=1}^{k}\|x^j - \bar{x}\|_1$$
$$+\tau\sum_{i=1}^{m}\sum_{j=1}^{k}\|x^j - a^i\|_1 - \tau\max_{t=1,\ldots,k}\sum_{j=1,j\neq t}^{k}\|x^j - a^i\|_1. \tag{6.4}$$

Expressing (6.4) as DC function, we have

$$\begin{aligned} f(X) &= (1+\tau)\sum_{i=1}^{m}\sum_{j=1}^{k}\|x^j - a^i\|_1 + \sum_{j=1}^{k}\|x^j - \bar{x}\|_1 \\ &\quad -\sum_{i=1}^{m}\max_{t=1,\ldots,k}\sum_{j=1,j\neq t}^{k}\|x^j - a^i\|_1 - \tau\sum_{j=1}^{k}\max_{r=1,\ldots,m}\sum_{i=1,i\neq r}^{m}\|x^j - a^i\|_1 \end{aligned} \tag{6.5}$$

where

$$g(X) = (1+\tau)\sum_{i=1}^{m}\sum_{j=1}^{k}\|x^j - a^i\|_1 + \sum_{j=1}^{k}\|x^j - \bar{x}\|_1$$

$$h(X) = \sum_{i=1}^{m}\max_{t=1,\ldots,k}\sum_{j=1,j\neq t}^{k}\|x^j - a^i\|_1 + \tau\sum_{j=1}^{k}\max_{r=1,\ldots,m}\sum_{i=1,i\neq r}^{m}\|x^j - a^i\|_1.$$

Since f is DC function based on Proposition 1.7.19 and ℓ_1 smoothing studied in (Nesterov, 2005; M. N. Nguyen et al., 2019), we obtain a Nesterov's approximation of $||x-a||_1$ as:

$$||x-a||_1 = \frac{\gamma}{2}\left[||\frac{x-a}{\gamma}||^2 - [d(\frac{x-a}{\gamma};F)]^2\right].$$

$$\begin{aligned} f_\gamma(X) &= \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}\|\frac{x^j - a^i}{\gamma}\|^2 + \sum_{j=1}^{k}\|\frac{x^j - \bar{x}}{\gamma}\|^2 \\ &\quad -\frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}[d(\frac{x^j - a^i}{\gamma};F)]^2 - \frac{\gamma}{2}\sum_{j=1}^{k}[d(\frac{x^j - \bar{x}}{\gamma};F)]^2 \\ &\quad -\sum_{i=1}^{m}\max_{t=1,\ldots,k}\sum_{j=1,j\neq t}^{k}||x^j - a^i||_1 \\ &\quad -\tau\sum_{j=1}^{k}\max_{r=1,\ldots,m}\sum_{i=1,i\neq r}^{m}\|x^j - a^i\|_1. \end{aligned} \tag{6.6}$$

$$\text{Minimize } \left\{ f_\gamma(X) = g_\gamma(X) - h_\gamma(X) \right\}, X \in \mathbb{R}^{k\times n}.$$

In addition, g_γ, h_γ are the sum of convex functions defined as:

$$g_\gamma(X) = g_{1\gamma}(X) + g_{2\gamma}(X) \tag{6.7}$$

where

$$g_{1\gamma}(X) = \frac{(1+\tau)\gamma}{2} \sum_{i=1}^{m} \sum_{j=1}^{k} \| \frac{x^j - a^i}{\gamma} \|^2, \;\; g_{2\gamma}(X) = \sum_{j=1}^{k} \| \frac{x^j - \bar{x}}{\gamma} \|^2.$$

And

$$h_\gamma(X) = h_{1\gamma}(X) + h_{2\gamma}(X) + h_{3\gamma}(X) + h_{4\gamma}(X) \tag{6.8}$$

where

$$h_{1\gamma}(X) = \frac{(1+\tau)\gamma}{2} \sum_{i=1}^{m} \sum_{j=1}^{k} [d(\frac{x^j - a^i}{\gamma}; F)]^2 \;\; h_{2\gamma}(X) = \frac{\gamma}{2} \sum_{j=1}^{k} [d(\frac{x^j - \bar{x}}{\gamma}; F)]^2,$$

$$h_{3\gamma}(X) = \sum_{i=1}^{m} \max_{t=1,\ldots,k} \sum_{j=1, j\neq t}^{k} ||x^j - a^i||_1, \;\; h_{4\gamma}(X) = \tau \sum_{j=1}^{k} \max_{r=1,\ldots,m} \sum_{i=1, i\neq r}^{m} \|x^j - a^i\|_1.$$

For the calculation of gradient and sub-gradient consider a data matrix A with a^i , $i = 1, \ldots, m$, in the i^{th} row and a variable matrix X with x^j, $j = 1, 2, \ldots, k$ in the j^{th} row.

Since X and A belongs to a linear space of real matrices, we can apply inner product such that

$$\langle X, A \rangle = \text{ trace}(X^T A) = \sum_{i=1}^{n} \sum_{j=1}^{k} x_{ij} a_{ij}.$$

And the Frobenius norm on $\mathbb{R}^{k\times m}$ is given by

$$\|A\|_F = \sqrt{\langle A, A \rangle} = \sqrt{\sum_{j=1}^{k} \langle a^j, a^j \rangle} = \sqrt{\sum_{j=1}^{k} ||a^j||^2}. \tag{6.9}$$

To calculate the gradient of g_γ in (6.7), let X be of size $k \times n$ is variable matrix. Then

$$\begin{aligned} g_{1\gamma}(X) &:= \frac{(1+\tau)}{2\gamma} \sum_{i=1}^{m} \sum_{j=1}^{k+1} ||x^j - a^i||^2 \\ &= \frac{(1+\tau)}{2\gamma} \sum_{i=1}^{m} \sum_{j=1}^{k} \left[\|x^j\|^2 - 2\langle x^j, a^i \rangle + \|a^i\|^2 \right] \\ &= \frac{(1+\tau)}{2\gamma} \left[m\|X\|_F^2 - 2\langle X, E_{km}A \rangle + k\|A\|_F^2 \right], \end{aligned}$$

where $E_{km} \in \mathbb{R}^{k \times m}$ is a matrix of all ones. As $g_{1\gamma}$ is smooth, then

$$\nabla_x g_{1\gamma}(X) = \frac{(1+\tau)}{\gamma} \left[mX - B \right] \text{where } \; B = E_{km}A.$$

Again consider $g_{2\gamma}$ which is differentiable function,

$$\begin{aligned} g_{2\gamma}(X) \;\; &:= \;\; \frac{1}{2\gamma} \sum_{j=1}^{k} ||x^j - \bar{x}||^2 \\ &= \;\; \frac{1}{2\gamma} \sum_{j=1}^{k} \left[\|x^j\|^2 - 2\langle x^j, \bar{x}j \rangle + \langle x^j, \bar{x} \rangle \right] \\ &= \;\; \frac{1}{2\gamma} \left[\|X\|_F^2 - \frac{2}{k} \langle X, E_{kk}X \rangle + \frac{1}{k} \langle X, E_{kk}X \rangle \right] \end{aligned}$$

where E_{kk} is a $k \times k$ matrix with elements all ones. Then, the gradients of $g_{2\gamma}$ are given by:

$$\begin{aligned} \nabla_x g_{2\gamma}(X) &= \frac{1}{\gamma} \left[X - \frac{1}{k} E_{kk}X \right] \\ &= \frac{1}{\gamma} \left[X - HX \right], \text{where } \; H = \frac{1}{k} E_{kk}. \end{aligned}$$

Next,we focus on $X \in \partial g^*(Y)$ where g^* is a Fenchel conjugate defined in (1.8) and can be calculated using the fact that $X \in \partial g^*(Y) \Leftrightarrow Y \in \partial g(X)$. Since g_γ is differentiable.
Thus,

$$\nabla_x g_\gamma(X) = \nabla_x g_{1\gamma}(X) + \nabla_x g_{2\gamma}(X).$$

$$
\begin{aligned}
Y &= \frac{(1+\tau)}{\gamma}[mX - B] + \frac{1}{\gamma}[X - HX] \\
&= \left[\frac{(1+\tau)}{\gamma}m + \frac{1}{\gamma}[\mathbb{I} - H]\right]X - \left[\frac{(1+\tau)}{\gamma}B\right] \\
&= \frac{1}{\gamma}[(1+\tau)m + \mathbb{I} - H]X - \left[\frac{(1+\tau)}{\gamma}B\right] \\
&= \frac{1}{\gamma}[(1+\tau)m + 1]\mathbb{I} - H]X - \left[\frac{(1+\tau)}{\gamma}B\right] \\
&= \frac{1}{\gamma}[a\mathbb{I} - bH]X - \left[\frac{(1+\tau)}{\gamma}B\right],
\end{aligned}
$$

where: $a = (1+\tau)m+1$ and $b = 1$. Let $N = a\mathbb{I} - bH$ then, N is invertible as $N^{-1} = \tau\mathbb{I} + \beta H$ where

$$
\tau = \frac{1}{a} = \frac{1}{(1+\tau)m+1} \text{ and } \beta = \frac{b}{a[a+bk]} = \frac{1}{(1+\tau)m+1[(1+\tau)m+1+k]},
$$

(see Lemma 5.1 of (Geremew et al., 2018)). Therefore,

$$
X = [\tau\mathbb{I} - \beta H]\,[\gamma Y_x + (1+\tau)B]\,.
$$

On the other hand, in the h parts in equation (6.8) of the DC decomposition, there are three types of convex functions involved. Two of the kinds are smooth, so computing their sub-gradients is equivalent to computing their gradients, and the third kind is non-smooth, but it is either the sum of the maximum of a finite collection of convex functions or just the maximum of a finite collection of convex functions. So, our next step is to demonstrate how one can compute a sub-gradient for such functions.

We will start with the first kinds – functions $h_{1\gamma}$, and $h_{2\gamma}$. For instance

$$
h_{1\gamma} := \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}\left[d\left(\frac{x^j - a^i}{\gamma}; F\right)\right]^2,
$$

and from its representation, one can see that it is differentiable, and hence its sub-gradient coincides with its gradient, that can be computed by the partial derivatives with respect to $x^1, \cdots, x^k$. Hence,

$$
\frac{\partial h_{1\gamma}}{\partial x^j} = \frac{\partial}{\partial x^j}\left[\frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}\left[d\left(\frac{x^j - a^i}{\gamma}; F\right)\right]^2\right] = (1+\tau)\sum_{i=1}^{m}\left[\frac{x^j - a^i}{\gamma} - P\left(\frac{x^j - a^i}{\gamma}; F\right)\right].
$$

Thus, for $j = 1, \ldots, k$, $\partial h_{1\gamma}(\mathbf{X})$ is a singleton and it is the $k \times n$ matrix whose j^{th} row is

$$
(1+\tau)\sum_{i=1}^{m}\left[\frac{x^j - a^i}{\gamma} - P\left(\frac{x^j - a^i}{\gamma}; F\right)\right].
$$

Similarly, it can be shown that

$$\frac{\partial h_{2\gamma}}{\partial x^j} = \left[\frac{\mathbf{x}^j - \bar{x}}{\gamma} - P\left(\frac{x^j - \bar{x}}{\gamma}; F\right)\right] - \frac{1}{k}\sum_{j=1}^{k}\left[\frac{x^j - \bar{x}}{\gamma} - P\left(\frac{x^j - \bar{x}}{\gamma}; F\right)\right].$$

Note that in the above cases, the Euclidean projection $P(z;F)$ from $z \in \mathbb{R}^n$ to F a unit box is given by

$$P(z;F) := \max(-e, \min(z, e)), \text{component-wise}, \tag{6.10}$$

where $e \in \mathbb{R}^n$ is the vector consisting of one in each component and zero elsewhere.

On the other hand, the third kinds $h_{3\gamma}$, and $h_{4\gamma}$, are non-smooth since we use ℓ_1 norm. Then, by applying Lemma 1.7.8 we can compute a sub-gradient for $h_{3\gamma}$ and $h_{4\gamma}$, as demonstrated below. For instance, we consider

$$h_{3\gamma} := \sum_{i=1}^{m} \max_{r=1,\ldots,k} \sum_{j=1, j\neq r}^{k} \|x^j - a^i\|_1 = \sum_{i=1}^{m} \phi_i(\mathbf{X}),$$

where, for $i = 1, \ldots, m$,

$$\phi_i(\mathbf{X}) := \max\left\{\phi_{ir}(\mathbf{X}) = \sum_{j=1, j\neq r}^{k} \|x^j - a^i\|_1, \;\; r = 1, \ldots, k\right\}.$$

To do this, for each $i = 1, \ldots, m$, we first find $\mathbf{U}_i \in \partial\phi_i(\mathbf{X})$ according to Lemma 1.7.8. Then we define $\mathbf{U} := \sum_{i=1}^{m} \mathbf{U}_i$ to get a sub-gradient of the function $h_{3\gamma}$ at $\mathbf{X}$ by the sub-differential sum rule. To accomplish this goal, we first choose an index r^* from the index set $\{1, \ldots, k\}$ such that

$$\phi_i(\mathbf{X}) = \phi_{ir^*}(\mathbf{X}) = \sum_{j=1, j\neq r^*}^{k} \|x^j - a^i\|_1.$$

Using the familiar sub-differential formula of the ℓ_1 norm function, the j^{th} row u_i^j for $j \neq r^*$ of the matrix $\mathbf{U}_i$ is determined as follows

$$u_i^j := \text{sign}(x^j - a^i) = \begin{cases} 1 & \text{if } x^j > a^i, \\ 0 & \text{if } x^j = a^i, \\ -1 & \text{if } x^j < a^i. \end{cases}$$

The r^{*th} row of the matrix $\mathbf{U}_i$ is $u_i^{r^*} := 0$.
Thus the sub-gradient of $h_{3\gamma}$ is defined as

$$\frac{\partial h_{3\gamma}}{\partial x^j} := \mathbf{U}.$$

Similarly the sub-gradient of $h_{4\gamma}$ is given by

$$h_{4\gamma} := \tau \sum_{j=1}^{k} \max_{s=1,\ldots,m} \sum_{i=1, i\neq s}^{m} \|x^j - a^i\|_1 = \tau \sum_{j=1}^{k} \psi_j(\mathbf{X}),$$

where, for $j = 1,\ldots,k$,

$$\psi_j(\mathbf{X}) := \max \left\{ \psi_{js}(\mathbf{X}) = \sum_{i=1, i\neq s}^{m} \|x^j - a^i\|_1, \;\; s = 1,\ldots,m \right\}.$$

To do this, for each $j = 1,\ldots,k$, we first find $\mathbf{W}_j \in \partial \psi_j(\mathbf{X})$. Then we define $\mathbf{W} := \tau \sum_{j=1}^{k} \mathbf{W}_j$ to get a sub-gradient of the function $h_{4\gamma}$ at $\mathbf{X}$ by the sub-differential sum rule. To accomplish this goal, we first choose an index s^* from the index set $\{1,\ldots,m\}$ such that

$$\psi_j(\mathbf{X}) = \psi_{js^*}(\mathbf{X}) = \sum_{j=1, j\neq s^*}^{k} \|x^j - a^i\|_1.$$

The s^{*th} row of the matrix $\mathbf{W}_j$ is $w^j_{s^*} := 0$.
Thus,

$$\frac{\partial h_{4\gamma}}{\partial x^j} := \tau \mathbf{W}.$$

Finally, by putting all these gradient and sub-gradient calculations together, we can propose a new improved algorithm for bi-level hierarchical clustering using the ℓ_1 norm. The description of the proposed algorithm is as follows: From the sub-gradient calculated above we have,

$$Y = \frac{\partial h_{1\gamma}}{\partial x^j}(X) + \frac{\partial h_{2\gamma}}{\partial x^j}(X) + \frac{\partial h_{3\gamma}}{\partial x^j}(X) + \frac{\partial h_{4\gamma}}{\partial x^j}(X)$$

Now, we have in position to implement DCA algorithm that will solve the problem as shown in Algorithm 5.

Algorithm 5: Bi-level Hierarchical Clustering Model II(a)

Input : $A, X_0, \tau_0, \gamma_0, N \in \mathbb{N}$,
while *stopping criteria for* $(\gamma, \tau, \varepsilon)$ *not reached* **do**
  $\tau = \frac{1}{(1+\tau)m+1}$
  $\beta = \frac{1}{(1+\tau)m+1[(1+\tau)m+1+k]}$
  for $k = 1,...,N$ **do**
    Search $Y_k \in \partial h_\gamma(X_{k-1})$
    Search $X_k = [\tau\mathbb{I} + \beta H]\,[\gamma Y_k + (1+\tau)B]$
  end
  update γ and τ,
end
Output: X_N.

6.1.1 Numerical Experiments

We wrote a MATLAB code to implement our algorithm and run it on hp with 1.99 GHz Intel Core i7 Processor, and 8 GB Memory. For the simulation part we consider different parameters , among those, we selected starting penalty parameter begin with any small number and increase through iteration. The initial smoothing parameter,begin with $\gamma = 1$ and decreasing through iteration. We tested our code on a dataset in $\mathbb{R}^2$ with 58 locations as shown in Figure 6.1. We run k-means on all the $30,856$ possible selections of three rows from the dataset, then we used the output returned from the k-means as a starting center $\mathbf{X}_0$ for our algorithm. The result of this experiment is a 90% convergence rate to the optimal solution with an average run time in seconds. In addition we used a GPS data of 142 towns, and cities in Ethiopia with population greater than 7000, out of which 65 are in Oromia state. we also tested the iteration time by taking 1000 randomly generated artificial dataset (see Fig.6.4).
Since the algorithms are modified DCA, there is no guarantee that our algorithms converge to a global optimal solution. However, we compared our result with the solutions generated by Brute-force for small data points. The performance of DCA Algorithm for model with a 142 towns and cities in Ethiopia with 4 clusters and a headquarter (see Fig.6.2) and 65 nodes, 4 cluster centers and one node serves as headquarter (see Fig.6.3) converges to optimal solution 86% of the time, it means out of 7 iteration 6 of them converges to near optimal value.

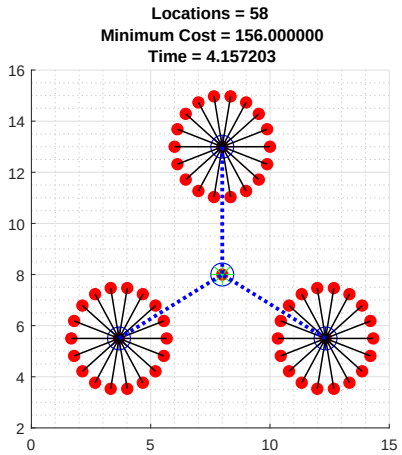


Figure 6.1: 58 Data Points, 3 Cluster Centers and a headquarter (HQ)

$$X = \begin{pmatrix} 12.3297 & 5.5002 \\ 8.0000 & 12.9994 \\ 3.6703 & 5.5002 \end{pmatrix} \text{ and } \quad HQ = (8.0000, 8.0000)$$

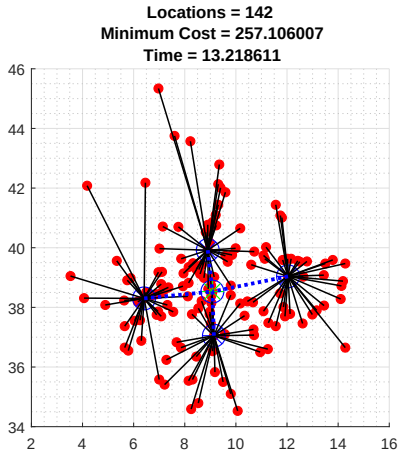


Figure 6.2: A 142 Ethiopian towns and cities four clusters and a headquarter (HQ).

$$X = \begin{pmatrix} 8.8994 & 39.9167 \\ 9.1398 & 37.0499 \\ 12.0400 & 39.0400 \\ 6.4107 & 38.3002 \end{pmatrix} \text{ and } \quad HQ = (9.0800,\ 38.5300)$$

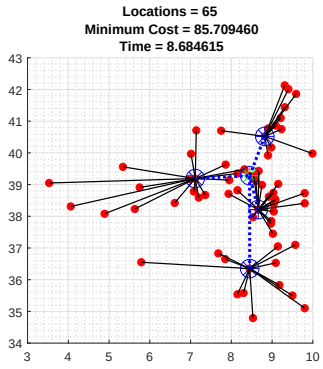


Figure 6.3: A 65 Oromia state towns and cities with four clusters and a headquarter (HQ).

$$X = \begin{pmatrix} 8.4500 & 36.3500 \\ 8.8199 & 40.5198 \\ 7.1142 & 39.1903 \\ 8.6601 & 38.2102 \end{pmatrix} \text{ and } \quad HQ = (8.4500,\ 39.2800)$$

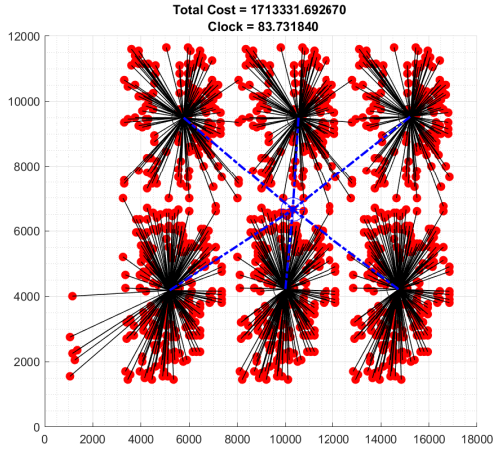


Figure 6.4: PR1002 (The 1002 City Problem) taken from (Reinelt, 1991) with 6 clusters and a HQ

.

$$X = \begin{pmatrix} 15281 & 9519 \\ 10581 & 9496 \\ 5819 & 9490 \\ 5186 & 4162 \\ 14727 & 4194 \\ 9986 & 4201 \end{pmatrix} \text{ and } \quad HQ = (10300 \quad 6650)$$

6.2 Bi-level Hierarchical Clustering Model IIb

Let the data of the problem be an $m \times n$ matrix A whose m rows $a^1, a^2, \ldots, a^m$ are potentially representing m locations in $\mathbb{R}^2$. The $k \times n$ matrix X is formed by the $1 \times n$ row vectors x^1, x^2, $\ldots$, x^k, represents the k cluster centers of the m locations. We are to choose k members in A, one as headquarter and others as centers of disjoint clusters. Headquarter is defined to lie in convex hull of x^j, i.e $x^* = \sum_{j=1}^{k} \lambda_j x^j$ where $\lambda_j \geq 0, \sum_{j=1}^{k} \lambda_j = 1$. This constraint limits the search for headquarter to the convex hull of the union of x^j for $j = 1, 2, ..., k$.
Thus the problem is formulated as:

$$\text{Minimize} \left\{ \sum_{i=1}^{m} \min_{j=1,\ldots,k} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - x^*\|_1 \right\}.$$

To ensure that the artificial centers to be real centers we add the following constraints.

$$\left\{ \begin{array}{l} \sum_{j=1}^{k} \min_{i=1,\ldots,m} \|x^j - a^i\|_1 = 0, x^1, ..., x^k \in \mathbb{R}^n, j = 1, 2, ..., k \\ \min_{i=1,\ldots,m} \|x^* - a^i\|_1 = 0, \lambda_j \in \mathbb{R}^k, \lambda_j \geq 0, \sum_{j=1}^{k} \lambda_j = 1 \end{array} \right\}.$$

Now we convert the constrained problem to an unconstrained optimization problem using penalty parameter τ and
$F = \left\{\Lambda = (\lambda_1, \lambda_2, ..., \lambda_k) \in \mathbb{R}^k : \lambda_j \geq 0, \sum_{j=1}^k \lambda_j = 1\right\}$.

$$\text{Minimize} \sum_{i=1}^{m} \min_{j=1,...,k} \|x^j - a^i\|_1 + \sum_{j=1}^{k} \|x^j - x^*\|_1 + \tau \sum_{i=1}^{k} \min_{j=1,...,m} \|x^j - a^i\|_1 + \tau \min_{i=1,...,m} \|x^* - a^i\|_1 + \frac{\tau}{2} [d(\Lambda; F)]^2, x^1, ..., x^k \in \mathbb{R}^n. \tag{6.11}$$

6.2.1 Developing The Proposed Algorithm

After the introduction of the penalty parameter $\tau > 0$ and the smoothing parameter $\gamma > 0$, the reformulated problem in equation (6.11) is still non-smooth and non-convex. However, for $X \in \mathbb{R}^{k \times n}$ and $\Lambda \in \mathbb{R}^{k \times 1}$ is a vector whose column is λ_j, it can be written as unconstrained DC optimization problem as follows:

$$\text{minimize} \quad f(X, \Lambda, \tau, \gamma) = [g_1 + g_2 + g_3 + g_4] - [h_1 + h_2 + h_3 + h_4 + h_5 + h_6 + h_7]. \tag{6.12}$$

where,

$$g_1 = \frac{1+\tau}{2\gamma} \sum_{i=1}^{m} \sum_{j=1}^{k} \|x^j - a^i\|^2, \qquad g_2 = \frac{1}{2\gamma} \sum_{j=1}^{k} \|x^j - \Lambda^T X\|^2,$$
$$g_3 = \frac{\tau}{2\gamma} \sum_{i=1}^{m} \|\Lambda^T X - a^i\|^2, \qquad g_4 = \frac{\tau}{2} \sum_{j=1}^{k} \lambda_j^2,$$
$$h_1 = \frac{(1+\tau)\gamma}{2} \sum_{i=1}^{m} \sum_{j=1}^{k} \left(d\left(\frac{x^j - a^i}{\gamma}; \mathbb{B}\right)\right)^2, \qquad h_2 = \frac{\gamma}{2} \sum_{j=1}^{k} \left(d\left(\frac{x^j - \Lambda^T X}{\gamma}; \mathbb{B}\right)\right)^2,$$
$$h_3 = \frac{\tau\gamma}{2} \sum_{i=1}^{m} \left(d\left(\frac{\Lambda^T X - a^i}{\gamma}; \mathbb{B}\right)\right)^2, \qquad h_4 = \sum_{i=1}^{m} \max_{r=1,...,k} \sum_{j=1, j\neq r}^{k} \|x^j - a^i\|_1,$$
$$h_5 = \tau \sum_{j=1}^{k} \max_{s=1,...,m} \sum_{i=1, i\neq s}^{m} \|x^j - a^i\|_1, \qquad h_6 = \tau \max_{s=1,...,m} \sum_{i=1, i\neq s}^{m} \|\Lambda^T X - a^i\|_1,$$

$h_7 = \frac{\tau}{2} \varphi_{\Delta_k}(\Lambda)$, where $\Delta_k = \{(\lambda_1, \lambda_2, \ldots, \lambda_k) \in \mathbb{R}^k_+ \mid \sum_{j=1}^k \lambda_j = 1\}$, and $\varphi_{\Delta_k}(\Lambda)$ is as defined in Proposition 1.7.20.
Our objective in this section is to develop a DCA based algorithm and to report on how we cope with selecting the penalty parameter $\tau > 0$, and the Nesterov's smoothing parameter $\gamma > 0$, such that (6.12) can be successfully solved by the DCA, consequently providing a solution to the original problem.

6.2.2 Gradient and Sub-gradient Calculations for the DCA

For the development of the proposed DCA based algorithm, we need to compute the gradients and sub-gradients of the functions involved in (6.12). For instance, for $j = 1, 2, \ldots, k$ we

have

$$\begin{aligned}
\frac{\partial g_1}{\partial x^j} &= \frac{(1+\tau)m}{\gamma}\left[x^j - \frac{1}{m}\sum_{i=1}^{m} a^i\right], \\
\frac{\partial g_2}{\partial x^j} &= \frac{1}{\gamma}\left[x^j - \Lambda^T X + k\lambda_j\left(\Lambda^T X - \frac{1}{k}\sum_{j=1}^{k} x^j\right)\right], \\
\frac{\partial g_3}{\partial x^j} &= \frac{m\tau\lambda_j}{\gamma}\left[\Lambda^T X - \frac{1}{m}\sum_{i=1}^{m} a^i\right], \\
\frac{\partial g_4}{\partial x^j} &= 0.
\end{aligned}$$

With some algebraic manipulation, the sum of the four partial derivatives above can be simplified as follows:

$$\begin{aligned}
\frac{\partial g_1}{\partial x^j} + \frac{\partial g_2}{\partial x^j} + \frac{\partial g_3}{\partial x^j} + \frac{\partial g_4}{\partial x^j} &= \left[\frac{(1+\tau)m+1}{\gamma} e_j + \frac{(k+\tau m)\lambda_j - 1}{\gamma}\Lambda^T - \frac{\lambda_j}{\gamma} e\right] X \\
&\quad - \left[\frac{(1+\tau+\tau\lambda_j)m+1}{\gamma}\right]\bar{A}
\end{aligned}$$

where, $\bar{A} = \frac{1}{m}\sum_{i=1}^{m} a^i$, $e = (1,1,\ldots,1) \in \mathbb{R}^k$, and $e_j = (0,\ldots,1,\ldots,0) \in \mathbb{R}^k$, with 1 at the j^{th} position, and for convenience of writing we set $(e_j)^T := e^j$.
Similarly, the partial derivatives, $\frac{\partial g_1}{\partial \lambda_j}$, $\frac{\partial g_2}{\partial \lambda_j}$, $\frac{\partial g_3}{\partial \lambda_j}$, and $\frac{\partial g_4}{\partial \lambda_j}$ can be computed and their sum can be simplified. For instance, for $j = 1,2,\ldots,k$ we have

$$\begin{aligned}
\frac{\partial g_1}{\partial \lambda_j} &:= \frac{\partial}{\partial \lambda^j}\left[\frac{1+\tau}{2\gamma}\sum_{i=1}^{m}\sum_{j=1}^{k}\|x^j - a^i\|^2\right] = 0, \\
\frac{\partial g_2}{\partial \lambda_j} &:= \frac{\partial}{\partial \lambda_j}\left[\frac{1}{2\gamma}\sum_{j=1}^{k}\|x^j - \Lambda^T X\|^2\right] = \frac{k}{\gamma}\left[\Lambda^T X - \frac{1}{k}\sum_{j=1}^{k} x^j\right] X^T e^j, \\
\frac{\partial g_3}{\partial \lambda_j} &:= \frac{\partial}{\partial \lambda_j}\left[\frac{\tau}{2\gamma}\sum_{i=1}^{m}\|\Lambda^T X - a^i\|^2\right] = \frac{\tau m}{\gamma}\left[\Lambda^T X - \frac{1}{m}\sum_{i=1}^{m} a^i\right] X^T e^j, \\
\frac{\partial g_4}{\partial \lambda_j} &:= \frac{\partial}{\partial \lambda_j}\left[\frac{\tau}{2}\sum_{j=1}^{k}\lambda_j^2\right] = \tau\lambda_j.
\end{aligned}$$

Again, with some algebraic manipulation, and by letting $\bar{X} = \frac{1}{k}\sum_{j=1}^{k} x^j$, the sum of the above four partial derivatives with respect to λ_j can be simplified as follows:

$$\frac{\partial g_1}{\partial \lambda_j} + \frac{\partial g_2}{\partial \lambda_j} + \frac{\partial g_3}{\partial \lambda_j} + \frac{\partial g_4}{\partial \lambda_j} = \left[\frac{k+\tau m}{\gamma} e_j X X^T + \tau e_j\right]\Lambda - \frac{k+\tau m}{\gamma}\left[\bar{X} + \bar{A}\right] X^T e^j.$$

Then, setting $\gamma = k + \tau m$, $\beta = m(1+\tau)+1$, and by applying Theorem 1.7.4, one can obtain

$[X_t, \Lambda_t] \in \partial g^*(Y_t, \Gamma_t)$ by solving the following systems of nonlinear equations,

$$\begin{bmatrix} \beta\mathbb{I} + \gamma\Lambda\Lambda^T - \Lambda e - e^T\Lambda^T & 0 \\ 0 & \frac{\tau}{\gamma}\mathbb{I} + \gamma XX^T \end{bmatrix} \begin{bmatrix} X & 0 \\ 0 & \Lambda \end{bmatrix} - \begin{bmatrix} B(\Lambda) & 0 \\ 0 & C(X) \end{bmatrix} = \gamma[\,Y,\,\Gamma] \quad (6.13)$$

In equation (6.13) 0 is a block matrix whose entries are zero, $B(\Lambda)$ is a $k \times n$ matrix, depending on Λ, whose j^{th} row is given by

$$m(1 + \tau + \tau\lambda_j)\bar{A},$$

and $C(X)$ is a $k \times 1$ matrix, depending on X, whose j^{th} row is given by

$$\gamma\left[\bar{X} + \bar{A}\right] X^T e^j.$$

The solution of the system is given by,

$$\begin{bmatrix} X & 0 \\ 0 & \Lambda \end{bmatrix} = \begin{bmatrix} \beta\mathbb{I} + \gamma\Lambda\Lambda^T - \Lambda e - e^T\Lambda^T & 0 \\ 0 & \frac{\tau}{\gamma}\mathbb{I} + \gamma XX^T \end{bmatrix}^{-1} \left(\gamma[\,Y,\,\Gamma] + \begin{bmatrix} B(\Lambda) & 0 \\ 0 & C(X) \end{bmatrix}\right).$$

$$= \begin{bmatrix} \beta\mathbb{I} + P & 0 \\ 0 & \frac{\tau}{\gamma}\mathbb{I} + \gamma Q \end{bmatrix}^{-1} \left(\gamma[\,Y,\,\Gamma] + \begin{bmatrix} B(\Lambda) & 0 \\ 0 & C(X) \end{bmatrix}\right).$$

where the matrices P and Q are respectively given as $P = \gamma\Lambda\Lambda^T - \Lambda e - e^T\Lambda^T$ and $Q = XX^T$. Thus, the gradients are obtained as follows.

$$\begin{bmatrix} X & 0 \\ 0 & \Lambda \end{bmatrix} = \begin{bmatrix} (\beta\mathbb{I} + P)^{-1} & 0 \\ 0 & \left(\frac{\tau}{\gamma}\mathbb{I} + \gamma Q\right)^{-1} \end{bmatrix} \left(\gamma[\,Y,\,\Gamma] + \begin{bmatrix} B(\Lambda) & 0 \\ 0 & C(X) \end{bmatrix}\right)$$

$$= \begin{bmatrix} \frac{1}{\beta}\mathbb{I} - \frac{1}{\beta P(P+\beta)}P & 0 \\ 0 & \frac{\gamma}{\tau}\mathbb{I} - \frac{\gamma\gamma^2 Q^2}{\tau(\tau XX^T + \gamma\gamma Q^2)} \end{bmatrix} \left(\gamma[\,Y,\,\Gamma] + \begin{bmatrix} B(\Lambda) & 0 \\ 0 & C(X) \end{bmatrix}\right).$$

On the other hand, in the h parts in equation (6.12) of the DC decomposition, there are three types of convex functions involved. Two of the kinds are smooth, so computing their sub-gradients is equivalent to computing their gradients, and the third kind is non-smooth, but it is either the sum of the maximum of a finite collection of convex functions or just the maximum of a finite collection of convex functions. So, our next step is to demonstrate how one can compute a sub-gradient for such functions.

We will start with the first kinds of functions h_1, h_2, and h_3. For instance

$$h_1 := \frac{(1+\tau)\gamma}{2} \sum_{i=1}^{m} \sum_{j=1}^{k} \left[d\left(\frac{x^j - a^i}{\gamma}; \mathbb{B}\right)\right]^2,$$

and from its representation, one can see that it is differentiable, and hence its sub-gradient coincides with its gradient, that can be computed by the partial derivatives with respect to $x^1, \cdots, x^k$. Hence,

$$\frac{\partial h_1}{\partial x^j} = \frac{\partial}{\partial x^j}\left[\frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}\left[d\left(\frac{x^j-a^i}{\gamma};\mathbb{B}\right)\right]^2\right] = (1+\tau)\sum_{i=1}^{m}\left[\frac{x^j-a^i}{\gamma} - P\left(\frac{x^j-a^i}{\gamma};\mathbb{B}\right)\right].$$

Thus, for $j = 1,\ldots,k$, $\partial h_1(\mathbf{X})$ is a singleton and it is the $k \times n$ matrix $\mathbf{Y}_1$ whose j^{th} row is

$$(1+\tau)\sum_{i=1}^{m}\left[\frac{x^j-a^i}{\gamma} - P\left(\frac{x^j-a^i}{\gamma};\mathbb{B}\right)\right].$$

Similarly, it can be shown that

$$\frac{\partial h_2}{\partial x^j} = \left[\frac{\mathbf{x}^j-\Lambda^T X}{\gamma} - P\left(\frac{x^j-\Lambda^T X}{\gamma};\mathbb{B}\right)\right] - \lambda_j\sum_{j=1}^{k}\left[\frac{x^j-\Lambda^T X}{\gamma} - P\left(\frac{x^j-\Lambda^T X}{\gamma};\mathbb{B}\right)\right].$$

and

$$\frac{\partial h_3}{\partial x^j} = \tau\lambda_j\sum_{i=1}^{m}\left[\frac{\Lambda^T X-a^i}{\gamma} - P\left(\frac{\Lambda^T X-a^i}{\gamma};\mathbb{B}\right)\right].$$

Again the partial of h_1, h_2, and h_3 with respect to λ_j for $j = 1,2,...,k$ is given as follows:

$$\frac{\partial h_1}{\partial \lambda_j}(\mathbf{X},\Lambda) = 0 \text{ as } h_1 \text{ free of } \lambda_j$$

$$\frac{\partial h_2}{\partial \lambda_j} = -\sum_{j=1}^{k}\left[\frac{x^j-\Lambda^T X}{\gamma} - P\left(\frac{x^j-\Lambda^T X}{\gamma};\mathbb{B}\right)\right]X^T e^j.$$

and

$$\frac{\partial h_3}{\partial \lambda_j} = \tau\sum_{i=1}^{m}\left[\frac{\Lambda^T X-a^i}{\gamma} - P\left(\frac{\Lambda^T X-a^i}{\gamma};\mathbb{B}\right)\right]X^T e^j.$$

Note that in the above cases, the Euclidean projection $P_{\mathbb{B}}(z)$ from $z \in \mathbb{R}^n$ to $\mathbb{B}$ a unit box is given by

$$P(z;\mathbb{B}) := \max(-e, \min(z,e)), \text{component-wise}, \tag{6.14}$$

where $e \in \mathbb{R}^n$ is the vector consisting of one in each component and zero elsewhere.
The second kind is the function $h_7 = \varphi_{\Delta_k}(\Lambda)$ as defined in Proposition 1.7.20. As stated in the Proposition, this function is smooth and its gradient with respect to Λ is given by

$$\nabla\varphi_{\Delta_k}(\Lambda) = 2P_{\Delta_k}(\Lambda), \tag{6.15}$$

the projection in (6.15) is a projections onto the unit simplex Δ_k. But the gradient of h_7 with respect to X is zero as it has no X terms.
On the other hand, the third kinds h_4, h_5, and h_6, are non-smooth since we use ℓ_1 norm. Then, by applying Lemma 1.7.8 we can compute a sub-gradient for h_4, h_5, and h_6, as demonstrated below. For instance, we consider

$$h_4 := \sum_{i=1}^{m} \max_{r=1,\dots,k} \sum_{j=1, j\neq r}^{k} \|x^j - a^i\|_1 = \sum_{i=1}^{m} \phi_i(\mathbf{X}),$$

where, for $i = 1,\dots,m$,

$$\phi_i(\mathbf{X}) := \max\left\{ \phi_{ir}(\mathbf{X}) = \sum_{j=1, j\neq r}^{k} \|x^j - a^i\|_1, \;\; r = 1,\dots,k \right\}.$$

To do this, for each $i = 1,\dots,m$, we first find $\mathbf{U}_i \in \partial\phi_i(\mathbf{X})$ according to Lemma 1.7.8. Then we define $\mathbf{U} := \sum_{i=1}^{m} \mathbf{U}_i$ to get a sub-gradient of the function h_4 at $\mathbf{X}$ by the sub-differential sum rule. To accomplish this goal, we first choose an index r^* from the index set $\{1,\dots,k\}$ such that

$$\phi_i(\mathbf{X}) = \phi_{ir^*}(\mathbf{X}) = \sum_{j=1, j\neq r^*}^{k} \|x^j - a^i\|.$$

Using the familiar sub-differential formula of the ℓ_1 norm function, the j^{th} row u_i^j for $j \neq r^*$ of the matrix $\mathbf{U}_i$ is determined as follows

$$u_i^j := \operatorname{sign}(x^j - a^i) = \begin{cases} 1 & \text{if } x^j > a^i, \\ 0 & \text{if } x^j = a^i, \\ -1 & \text{if } x^j < a^i. \end{cases}$$

The r^{*th} row of the matrix $\mathbf{U}_i$ is $u_i^{r^*} := 0$.
Similarly the sub-gradient of h_5 is given by

$$h_5 := \tau \sum_{j=1}^{k} \max_{s=1,\dots,m} \sum_{i=1, i\neq s}^{m} \|x^j - a^i\|_1 = \tau \sum_{j=1}^{k} \psi_j(\mathbf{X}),$$

where, for $j = 1,\dots,k$,

$$\psi_j(\mathbf{X}) := \max\left\{ \psi_{js}(\mathbf{X}) = \sum_{i=1, i\neq s}^{m} \|x^j - a^i\|, \;\; s = 1,\dots,m \right\}.$$

To do this, for each $j = 1,\dots,k$, we first find $\mathbf{W}_j \in \partial\psi_j(\mathbf{X})$. Then we define $\mathbf{W} := \tau\sum_{j=1}^{k} \mathbf{W}_j$ to get a sub-gradient of the function h_5 at $\mathbf{X}$ by the sub-differential sum rule. To accomplish

this goal, we first choose an index s^* from the index set $\{1,\ldots,m\}$ such that

$$\psi_j(\mathbf{X}) = \psi_{js^*}(\mathbf{X}) = \sum_{j=1, j\neq s^*}^{k} \|x^j - a^i\|.$$

The s^{*th} row of the matrix $\mathbf{W}_j$ is $w^j_{s^*} := 0$.
thus the sub-gradient of h_5 is defined as

$$\frac{\partial h_5}{\partial x^j} := \tau \mathbf{W}.$$

The sub-gradient of h_4 and h_5 with respect to λ_j are zero as they have no λ term. A sub-gradient of h_6 is obtained as,

$$h_6 := \tau \max_{s=1,\ldots,m} \left(\sum_{j=1, j\neq s}^{k} \|\Lambda^T X - a^i\|_1 \right) = \tau \xi(X,\Lambda)$$

$$\xi(X,\Lambda) = \max \left\{ \xi_s(X,\Lambda) = \sum_{j=1, j\neq s}^{k} \|\Lambda^T X - a^i\|_1, \;\; s = 1,\ldots,m \right\}.$$

Using the definition of sub-gradient of the max function, let $\mathbf{T}_x \in \partial\xi(X,\Lambda)$ at X. Then the sub-gradient of h_6 at X is given by

$$\frac{\partial h_6}{\partial x^j} = \tau \mathbf{T}_x$$

where $\mathbf{T}_x$ is the $k \times n$ matrix whose j^{th} row is $\lambda_j t_s$, for $s = 1,\ldots,m$.
Similarly, the sub-gradient of h_6 at Λ is given by

$$\frac{\partial h_6}{\partial \lambda_j} := \tau \mathbf{M}.$$

where $\mathbf{M}$ is the $1 \times n$ matrix whose j^{th} row is $m_s X^T e^j$, for $s = 1,\ldots,m$.
Finally, by putting all these gradient and sub-gradient calculations together, we can propose a new improved algorithm for bi-level hierarchical clustering using the ℓ_1 norm. The description of the proposed algorithm is as follows:

Algorithm 6: Bi-level Hierarchical Clustering Model II(b)

Input : $A, X_0, \tau_0, \gamma_0, N \in \mathbb{N}$.

while *stopping is not reached* **do**

  for $t = 1, \ldots, N$ **do**

    $[Y_t, \Gamma_t] \leftarrow \partial h(X_{t-1}, \Lambda_{t-1})$

    $[X_t, \Lambda_{t-1}] \leftarrow \partial g^*(Y_t, \Gamma_t)$

  end

end

Output: X_N

6.2.3 Simulation Results

We wrote a MATLAB code to implement our algorithm and run it on hp with 1.99 GHz Intel Core i7 Processor, and 8 GB Memory. We tested our code on a dataset in $\mathbb{R}^2$ with 58 locations as shown in Figure 6.5. However, for the artificial test dataset we created to test the performance of Algorithm 6 with 58 nodes, 3 clearly identifiable cluster centers, and a HQ (see Fig. 6.5), the algorithm converges 90% of the time to a global optimal solution.Out of 11 iterations 10 of them converge to the same cluster centers. In all the cases we fixed Λ to a constant value Λ_0. we used $\Lambda_0 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$ for artificial data in Figure 6.5. For the real 142 data points collected from some cities and towns in Ethiopia and 65 towns and cities in Oromia regional state respectively we used $\Lambda_0 = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})^T$ (see figure 6.6 and figure 6.7). For the randomly generated 1000 data points in Figure 6.8, $\Lambda_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})^T$ is used. The result of this experiment is a 90% convergence rate to the optimal solution with an average run time in seconds.

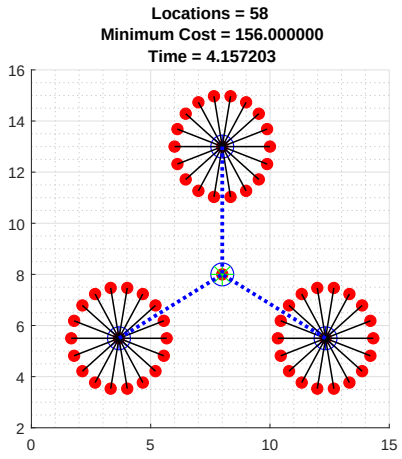


Figure 6.5: 58 Data Points, 3 Cluster Centers and a headquarter (HQ)

$$X = \begin{pmatrix} 12.3297 & 5.5002 \\ 8.0000 & 12.9994 \\ 3.6703 & 5.5002 \end{pmatrix} \text{ and } \quad HQ = (8.0000, 8.0000)$$

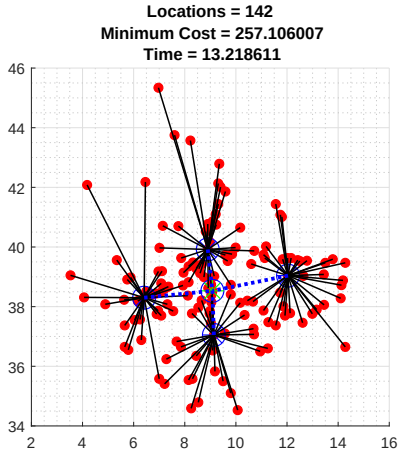


Figure 6.6: A 142 Ethiopian towns and cities four clusters and a HQ.

$$X = \begin{pmatrix} 8.8994 & 39.9167 \\ 9.1398 & 37.0499 \\ 12.0400 & 39.0400 \\ 6.4107 & 38.3002 \end{pmatrix} \text{ and } \quad HQ = (9.0800,\ 38.5300)$$

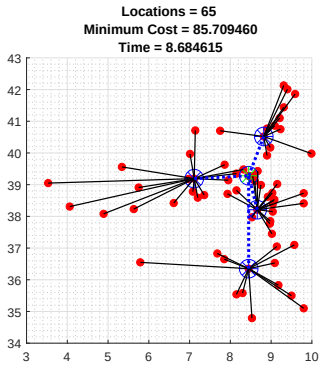


Figure 6.7: A 65 Oromia state towns and cities with four clusters and a HQ.

$$X = \begin{pmatrix} 8.4500 & 36.3500 \\ 8.8199 & 40.5198 \\ 7.1142 & 39.1903 \\ 8.6601 & 38.2102 \end{pmatrix} \text{ and } \quad HQ = (8.4500,\ 39.2800)$$

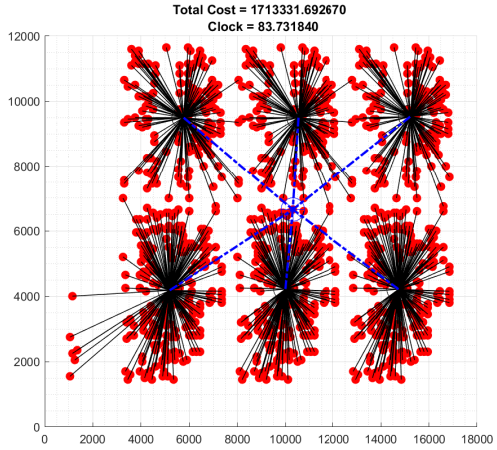


Figure 6.8: PR1002 (The 1002 City Problem) taken from (Reinelt, 1991) with 6 clusters and a HQ

.

$$X = \begin{pmatrix} 15281 & 9519 \\ 10581 & 9496 \\ 5819 & 9490 \\ 5186 & 4162 \\ 14727 & 4194 \\ 9986 & 4201 \end{pmatrix} \text{ and } \quad HQ = (10300 \quad 6650)$$

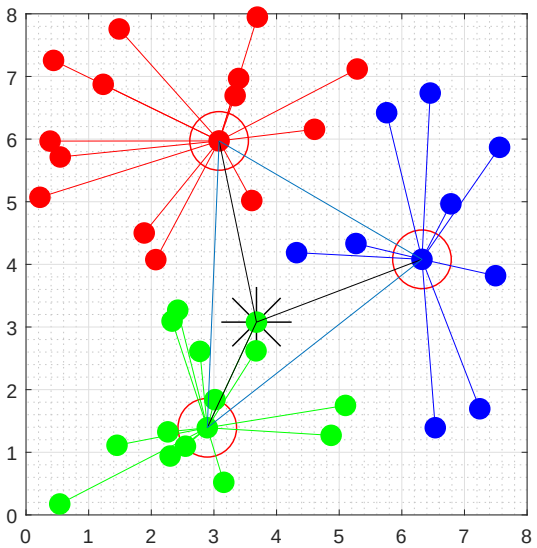


Figure 6.9: Test data

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusions

This dissertation presented the fundamental tools of convex analysis and optimization, establishing the mathematical groundwork for subsequent chapters. It also offered numerical techniques and algorithms, specifically Nesterov's smoothing technique and DCA.
Additionally, it covered calculus rules for generalized differentiation and optimization methods aimed at addressing non-smooth and non-convex problems, particularly in the context of clustering.
A variational geometric approach is employed to establish calculus rules for sub-gradients and Fenchel conjugates of convex functions that may not be differentiable in locally convex topological and Banach spaces. These rules are beneficial for various non-smooth optimization applications, both in theoretical and numerical contexts. We then examine optimization methods for addressing non-smooth optimization problems where the objective functions are not necessarily convex. Our primary focus is on the class of functions that can be expressed as differences of convex functions. This class is sufficiently broad to encompass many issues related to two-level hierarchical clustering, where tools from convex analysis for generalized differentiation can be applied. We develop algorithms to tackle a range of clustering and two-level hierarchical clustering challenges and implement these algorithms computationally using MATLAB.
A numerical algorithm called DCA, which minimizes the differences of convex functions, is employed in this work along with Nesterov's smoothing technique to deal with non-smoothness and non-convexity. The degree of similarity (dissimilarity) between two data points (nodes) is determined by ℓ_1 distances. Consequently, we presented three DCA-based algorithms for solving clustering and two distinct network topologies of bi-level hierarchical clustering problems.
We anticipate that our approach, which minimizes potential clustering mistakes and reduces iteration, will be less susceptible to outliers than the ℓ_2 norm for solving the two-level clustering issue in this dissertation. The selection of the best parameters for our algorithms posed a significant challenge for us during our numerical experiments. Regarding the accuracy and convergence speed of our suggested algorithms, we find that parameter selection plays a critical role. Thus the performance of the proposed algorithms strongly depends on the initial values set for the penalty and smoothing parameters τ_0 and γ_0 and their respective growth/-

damping factors σ_1 and σ_2.
We used MATLAB to implement the algorithms and tested them in two dimensions on various datasets of varying sizes. We anticipate that our approach, which minimizes potential clustering errors, is less susceptible to outliers than the ℓ_2 norm when solving the bi-level clustering problems and the clustering problem.
In addition, we compared our result with Brute-force generated solutions for datasets with fewer nodes which converge to a near-optimal value with reasonable time compared to the brute-force iterations.
This research not only advances the field of non-smooth optimization but also offers a scalable solution to clustering problems in large geographic networks. As the DC Algorithms are flexible, It can also be applied to other non-convex and non-smooth optimization issues in signal processing, like compressed sensing and image pixel clustering for image segmentation.

7.2 Future work

We developed two-level hierarchical clustering models and clustering models in this dissertation; however, it should be noted that most of the algorithms were implemented in two dimensions. Hence in the future, it will need to be extended into higher dimensions.
Furthermore, we used an algorithm that didn't have any non-negative or positive weights. Thus, another avenue for future research is the use of arbitrary weights.
From the mathematical point of view, since the cost function in all the three models are convex, they have a global minimum solution on a bounded domain. However, the smoothed approximation is not convex and there is no guarantee that we can find the global optimal value for the cost function. Therefore, rather than using a smoothing approximation, our future research will examine alternative methods that could yield a more accurate approximation of the global solution.

REFERENCES

Abbaszadehpeivasti, H., Klerk, E. de, et al. (2024). On the rate of convergence of the difference-of-convex algorithm (dca). *Journal of Optimization Theory and Applications*, *202*(1), 475–496.

Aggarwal, C. C., Hinneburg, et al. (2001). On the surprising behavior of distance metrics in high dimensional space. In *Database theory-icdt 2001: 8th international conference london, uk, january 4–6, 2001 proceedings 8* (pp. 420–434).

An, L., Belghiti, M., et al. (2007). A new efficient algorithm based on dc programming and dca for clustering. *J. Glob. Optim.*, *27*, 503-608.

An, L., & Minh, L. (2007). Optimization based dc programming and dca for hierarchical clustering. *Eur. J. Oper. Res.*, *183*, 1067-10851.

An, L., Minh, L., et al. (2014). New and efficient dca based algorithms for minimum sum-of-squares clustering. pattern recognit. *Ann. New York Acad. Sci.*, *47*, 388-401.

An, L., Ngai, H., et al. (2018). Convergence analysis of difference-of-convex algorithm with subanalytic data. *J. Optim. Theory Appl.*, *179(1)*, 103-126.

An, L. T. H., & Tao, P. D. (2005). The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems. *Annals of operations research*, *133*, 23–46.

An, N., & Nam, N. (2017). Convergence analysis of a proximal point algorithm for minimizing differences of functions. *Optim.*, *66(1)*, 129-147.

An, T. P., L.T.H. (1997). Convex analysis approach to d.c. programming: theory, algorithms and applications. *Acta Math. Vietnam*, *22*, 289-355.

Bagirov, A. (1999). Derivative-free methods for unconstrained nonsmooth optimization and its numerical analysis. *Investig. Oper.*, *19*, 75–93.

Bagirov, A., Jia, L., et al. (2003). Optimization based clustering algorithms in multicast group hierarchies. in: Proceedings of the australian telecommunications, networks and applications conference (atnac). *Melbourne Australia (published on CD, ISNB 0-646-42229-4*.

Bagirov, A., & Taheri, S. (2017). A dc optimization algorithm for clustering problems with ℓ_1-norm. *Iranian Journal of Operations Research*, *8*(2), 2–24.

Bagirov, A., Taheri, S., et al. (2016). Nonsmooth dc programming approach to the minimum sum-of-squares clustering problems. *Pattern Recognit*, *53*, 12-24.

Bagirov, A. M., Karasözen, B., et al. (2008). Discrete gradient method: derivative-free method for nonsmooth optimization. *Journal of Optimization Theory and Applications*, *137*, 317–334.

Bagirov, A. M., & Mohebi, E. (2015). Nonsmooth optimization based algorithms in cluster analysis. *Partitional clustering algorithms*, 99–146.

Bagirov, A. M., & Mohebi, E. (2016). An algorithm for clustering using l1-norm based on hyperbolic smoothing technique. *Computational Intelligence*, *32*(3), 439–457.

Bajaj, A., Mordukhovich, B. S., Nam, N. M., et al. (2022). Solving a continuous multifacility location problem by dc algorithms. *Optimization Methods and Software*, *37*(1), 338–360.

Barbosa, G. V., & Villas-Boas, S. B. o. (2013). Solving the two-level clustering problem by hyperbolic smoothing approach, and design of multicast networks. *In: Selected Proceedings, WCTR RIO*.

Burke, J. V., Lewis, A. S., et al. (2002). Approximating subdifferentials by random sampling of gradients. *Mathematics of Operations Research*, *27*(3), 567–584.

Carmichael, J. W., & Sneath, P. H. (1969). Taxometric maps. *Systematic Zoology*, *18*(4), 402–415.

Geremew, W., Nam, N. M., et al. (2018). A dc programming approach for solving multicast network design problems via the nesterov smoothing technique. *J Glob Optim*.

Hartman, P. (1959). On functions representable as a difference of convex functions.

Hiriart-Urruty, Jean-Baptiste, et al. (2004). *Fundamentals of convex analysis*. Springer Science & Business Media.

Hiriart-Urruty, J.-B. (1985). Generalized differentiability/duality and optimization for problems dealing with differences of convex functions. In *Convexity and duality in optimization: Proceedings of the symposium on convexity and duality in optimization held at the university of groningen, the netherlands june 22, 1984* (pp. 37–70).

Hiriart-Urruty, J. B., & LemarÃl'chal, C. (2001). Fundamentals of convex analysisl. *Springer*.

Horst, R. (2000). *Introduction to global optimization*. Springer Science & Business Media.

Horst, R., & Thoai, N. V. (1999). Dc programming: overview. *Journal of Optimization Theory and Applications*, *103*, 1–43.

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

Jajuga, K. (1987). A clustering method based on the l1-norm. *Computational Statistics & Data Analysis*, *5*(4), 357–371.

Jörnsten, R. (2004). Clustering and classification based on the l1 data depth. *Journal of Multivariate Analysis*, *90*(1), 67–89.

Kashima, H., Hu, J., et al. (2008). K-means clustering of proportional data using l1 distance. In *2008 19th international conference on pattern recognition* (pp. 1–4).

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

Kiwiel, K. C. (1990). Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical programming*, *46*(1), 105–122.

Lemarechal, C., & Zowe, J. (1994). A condensed introduction to bundle methods in nonsmooth optimization. In *Algorithms for continuous optimization: the state of the art* (pp. 357–382). Springer.

Mau Nam, N., Hoai An, L. T., et al. (2020). Smoothing techniques and difference of convex functions algorithms for image reconstructions. *Optimization*, *69*(7-8), 1601–1633.

Melzer, D. (1986). On the expressibility of piecewise-linear continuous functions as the difference of two piecewise-linear convex functions. *Quasidifferential calculus*, 118–134.

Mordukhovich, B., & Nam, N. (2014). *An easy path to convex analysis and applications*. Morgan & Claypool Publishers.

Mordukhovich, B. S. (2006). Variational analysis and generalized differentiation, i: Basic theory, ii: Applications. *Springer*.

Mordukhovich, B. S., & Nam, N. M. (2022). *Convex analysis and beyond*. Springer.

Moreau, J. J. (1963). *Etude locale d'une fonctionnelle convexe*. Faculté des sciences.

Nam, N., Rector, R., et al. (2017). Minimizing differences of convex functions with applications to facility location and clustering. *J. Optim. Theory Appl.*, *173*, 255-278.

Nam, N., W, G., et al. (2018). The nesterov smoothing technique and minimizing differences of convex functions for hierarchical clustering. *Optim. Lett.*, *12*, 455-473.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Math. Program.Math.*, *103*, 127-152.

Nesterov, Y. (2018). *Lectures on convex optimization* (2nd ed.). Cham, Switzerland: Springer.

Nguyen, L. T. H., P.A. (2020.). Dca approaches for simultaneous wireless information power transfer in miso secrecy channel. *Optim Eng*.

Nguyen, M. N., Thi, H. A. L., et al. (2019). Smoothing techniques and difference of convex functions algorithms for image reconstructions. *Optimization*, 1-33.

Niu, Y.-S. (2022). On the convergence analysis of dca. *arXiv preprint arXiv:2211.10942*.

Ordin, B., & Bagirov, A. M. (2015). A heuristic algorithm for solving the minimum sum-of-squares clustering problems. *Journal of Global Optimization*, *61*(2), 341–361.

Pelleg, D., Moore, A., et al. (2000). X-means: Extending k-means with e cient estimation of the number of clusters. In *Icml'00* (pp. 727–734).

Polak, E., & Royset, J. (2003). Algorithms for finite and semi-infinite min–max–min problems using adaptive smoothing techniques. *Journal of optimization theory and applications*, *119*, 421–457.

Reinelt, G. (1991). Tsplib: A traveling salesman problem library. *ORSA journal on computing*, *3*(4), 376–384.

Rockafellar, R. (1970). Convex analysis. *Princeton University Press, Princeton, NJ*.

Sabo, K. (2014). Center-based l–clustering method. *International Journal of Applied Mathematics and Computer Science*, *24*(1), 151–163.

Späth, H. (1976). Algorithm 30 l 1 cluster analysis. *Computing*, *16*(4), 379–387.

Sun, D., Toh, K.-C., et al. (2021). Convex clustering: Model, theoretical guarantee and efficient algorithm. *Journal of Machine Learning Research*, *22*(9), 1–32.

Tao, P., & An, L. (1998). A d.c. optimization algorithm for solving the trust-region subproblem. *SIAM J.Optim.*, *8*, 476-505.

Teboulle, M. (2007). A unified continuous optimization framework for center-based clustering methods. *Journal of Machine Learning Research*, *8*(1).

THI, H. L., & DINH, T. P. (2018). Dc programming and dca: Thirty years of. *Math. Program.*, *169*, 5-68.

Tran, T. D. T. (2020). *Convex and nonconvex optimization techniques for multifacility location and clustering*. Unpublished doctoral dissertation, Portland State University.

Tuy, H., Hoang, T., et al. (1998). *Convex analysis and global optimization*. Springer.

Tuy, H., & Tuy, H. (2016). Nonconvex quadratic programming. *Convex Analysis and Global Optimization*, 337–390.

Xavier, A. E., & Xavier, V. L. (2011). Solving the minimum sum-of-squares clustering problem by hyperbolic smoothing and partition into boundary and gravitational regions. *Pattern Recognition*, *44*(1), 70–77.

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, *16*(3), 645–678.

Zalinescu, C. (2002). Convex analysis in general vector spaces. *World Scientific*.

Zhang, J., Peng, L., et al. (2012). Robust data clustering by learning multi-metric lq-norm distances. *Expert Systems with Applications*, *39*(1), 335–349.

Contribution of the Dissertation

The dissertation report's original results that contribute knowledge to the scientific advancement in Mathematics. Two of the contributions were published in peer-reviewed journals indexed in Web of Science (ESCI), the DOAJ,EBSCO and Scopus while one is submitted for publication.

P1. Adugna Fita, Wondi Geremew, & Legesse Lemecha. A DC Optimization Approach for Constrained Clustering with ℓ_1-Norm. *Palestine Journal of Mathematics*, 11(3), 2022.

P2. Adugna Fita, and Legesse Lemecha (2024) A DC programming to two-level hierarchical clustering with ℓ_1 norm. *Front. Appl. Math. Stat. 10:1445390.* https://doi:10.3389/fams.2024.144590

APPENDIX A

MATLAB CODES

A.1 Matlab codes

A.1.1 The Matlab m-files for the simulations of clustering model

```
clear; clc; clf;

 load('oro_city_data.mat')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

MaxIter =5;
k =5;

m = size(A, 1 );
Data_Size = m;
lambda_0 = 1e-6; % Starting penality parameter.
mu_0 = 0.5; % Starting smoothing parameter.

sigma_1 = 1e+12; % Penality parameter's growth factor

sigma_2 = 0.75; % Smoothing parameter's decay factor

ifl = 100; % Limit for the inner for loop (DCA!)

ofl =18; % Limits for the outer for loop (Smoothing!)

epsilon = 1e-9;

S = ones(k,m)*A;

for t = 1 : MaxIter

```

```
29      tic
30
31  X = datasample( A , k , 1 , 'Replace' , false );
32
33      for w = 1 : ofl
34
35          lambda = lambda_0*(sigma_1^(w-1));
36
37          mu = mu_0*(sigma_2^(w-1));
38
39          c = 1 + lambda;
40
41          for G = 1 : ifl
42
43              clf;
44
45              XO = X;
46
47              U = H11( X , A , lambda, mu );
48
49              W = H12( X , A );
50
51              Z = H13( X , A , lambda);
52
53              X =  mu*( U + W + Z  )  / (c*m) + S/m; % k
                    Cluster Centers.
54              if norm(X - XO, 'fro')/(norm(XO, 'fro') + 1) <
                    epsilon
55
56                  break
57
58              end
59
60          end
61
62      end
63
64                  TA = A;
65              ci = zeros( 1 , m );
```

```
66
67              D = DistMatXA( X , TA );
68               axis( [ 3   11   34   43] ),%oromia
69              grid on ;grid minor;
70              hold on
71
72              plot( A( : , 1), A( : , 2), 'r.', 'markersize',
                   35 );
73              hold on
74
75              plot( X( : , 1), X( : , 2), 'ko', 'markersize',
                   15);
76              hold on
77               plot(  X( : , 1), X( : , 2), 'k*',  'markersize'
                    , 15);
78              hold on
79
80
81              for i = 1 : m
82
83                  ci( i ) = find( D( : , i) == min( D( : , i) )
                        , 1 );
84
85                  x1 = [ TA( i , 1), X( ci( i ) , 1) ];
86
87                  y1 = [ TA( i , 2), X( ci( i ) , 2) ];
88
89                  plot(x1, y1, 'k-', 'LineWidth',2); xlabel('x-
                       axis (Latitude)'),ylabel('y-axis (
                       Longitude)');
90                  hold on
91
92              end
93
94
95              title(sprintf('Outer For Loop = %d  \n Inner For
                   Loop = %d', w, G))
96              pause( 0.0000000000001 )
97
```

```
98      Cost = sum( min( D ));
99
100     Clock = toc;
101
102     title(sprintf('Locations = %d\n Total Cost = %f\n Time =
          %f',Data_Size,...
103     Cost,Clock))
104
105     pause(1)
106
107 end
108
109 X
110 function     DistMat = DistMatXA( X , A )
111
112  % DistMatXA( X , A ) returns a ( k x m ) matrix whose ( l ,
       i )^th entry is the
113  % distance, i.e., norm( Xl - Ai ), of the l^th Cluster
       Center Xl, l = 1 : k, from the
114  % i^th Node Ai, i = 1 : m.
115
116   m = size( A , 1 );     k = size( X , 1 );     DistMat = zeros
        ( k , m );
117
118   for l = 1 : k
119
120       for i = 1 : m
121
122           DistMat( l , i ) = norm( X( l , : ) - A( i , : ),1
                );
123
124       end
125   end
126 end
127
128 function U = H11( X , A , lambda , mu )
129
130 [m,n] = size(A);     k = size(X,1);     U = zeros(k,n);
131
```

```
132 for l = 1:k
133
134     for i = 1:m
135
136         vli = X(l,:) - A(i,:);
137
138         normvli = norm(vli,1);
139
140         if normvli > mu
141
142           U(l,:) = U(l,:) + vli/mu - vli/normvli;
143
144         end
145
146     end
147
148 end
149
150 U = (1 + lambda)*U;
151
152 end
153
154
155 function W = H12( X , A )
156
157 [m,n] = size( A );
158
159 k = size( X , 1 );
160 u = randn(1,n);
161
162 u = u / ( 1 + norm(u) );
163
164 W = zeros( k , n );
165
166 DistMat = DistMatXA( X , A );
167
168 for i = 1 : m
169
170     Wi = zeros( k , n );
```

```
171
172     min_Disti = min(DistMat( : , i ) );
173
174     i_max = find( DistMat( : , i ) == min_Disti );
175
176     G = length( i_max );
177
178     for lMax = 1 : k
179
180         if ~ismember( lMax , i_max )
181
182             continue;
183
184         end
185
186          WlMax = zeros( k , n );
187
188         for j = 1 : k
189
190             if j ~= lMax
191
192                 norm_ji = norm( X( j , : ) - A( i , : ),1 );
193
194                 if norm_ji > 0
195
196                     WlMax( j , : ) = ( X( j , : ) - A( i , :
                            ) ) / norm_ji;
197
198                 else
199
200                     WlMax( j , : ) = u;
201
202                 end
203
204             end
205
206         end
207
208         Wi = Wi + WlMax/G;
```

```
209
210         end
211
212         W =( W + Wi);
213
214 end
215
216 end
217 function  Z = H13( X , A , lambda )
218
219 [m,n] = size( A );
220
221 k = size( X , 1 );
222
223 u = randn( 1 , n );
224
225 u = u / ( 1 + norm(u) );
226
227 Z = zeros( k , n );
228
229 DistMat = DistMatXA( X , A );
230
231 for l = 1 : k
232
233     min_Distl = min( DistMat( l , : ) );
234
235     l_max = find( DistMat( l , : ) == min_Distl );
236
237     G = length( l_max );
238
239     for iMax = 1 : m
240
241         if ~ismember( iMax , l_max )
242
243             continue;
244
245         end
246
247          ZiMax = zeros( 1 , n );
```

```
248
249             for j = 1 : m
250
251                 if j ~= iMax
252
253                     norm_lj = norm( X( 1 , : ) - A( j , : ),1 );
254
255                     if norm_lj > 0
256
257                         ZiMax = ZiMax + ( X( 1 , : ) - A( j , : )
                                ) / norm_lj;
258
259                     else
260
261                         ZiMax = ZiMax + u;
262
263                     end
264
265                 end
266
267             end
268
269              Z( 1 , : ) = Z( 1 , : )  +  ZiMax/G;
270
271         end
272
273 end
274
275 Z = lambda*Z;
276
277 end
```

A.2 Data of Location of Towns and Cities

Taken from https://www.tageo.com/index-e-et-cities-ET.htm

A.2.1 Oromia Cities and Towns

No	City	Population (in 2000)	Latitude (DD)	Longitude (DD)	
1	Moyale	14100	3.53	39.05	
2	Mega	7000	4.06	38.31	
3	Yabelo	13800	4.89	38.08	
4	Negele	40400	5.34	39.56	
5	Bule Hora	17100	5.63	38.23	
6	Shakiso	21100	5.75	38.91	
7	Bako	14000	5.79	36.55	
8	Wendo	15200	6.61	38.42	
9	Dodola	18600	6.97	39.18	
10	Goba	45000	7.01	39.97	
11	Kofele	9800	7.08	38.78	
12	Asasa	14600	7.11	39.19	
13	Ginir	16200	7.14	40.71	
14	Shashemenne	88500	7.2	38.59	
15	Negele	37600	7.36	38.67	
16	Jimma	117600	7.68	36.83	
17	Sheh hussen	10600	7.75	40.7	
18	Agaro	33100	7.85	36.65	
19	Robe	35400	7.86	39.63	
20	Batu	26900	7.93	38.71	
21	Assela	61300	7.95	39.14	
22	Mek'i	33100	8.15	38.82	
23	Huruta	12700	8.15	39.35	
24	Gore	9500	8.15	35.54	
25	Metu	31100	8.3	35.58	
26	Sire	10300	8.32	39.48	
27	Bedele	16000	8.45	36.35	
28	Wonji	12800	8.45	39.28	
29	Giyon	37000	8.53	37.97	
30	Dembi dolo	26700	8.53	34.79	
31	Adama	176800	8.55	39.27	
32	Mojo	32700	8.6	39.12	
33	Tulu bolo	10700	8.66	38.21	
34	Welench'iti	15700	8.67	39.43	
35	Debre zeyit	98700	8.75	38.99	
36	Gelemso	14600	8.82	40.52	
37	Metehara	16000	8.9	39.92	

No	City	Population (in 2000)	Latitude (DD)	Longitude (DD)	
38	Bedesa	14500	8.9	40.77	
39	Sebeta	19400	8.92	38.62	
40	Guder	12800	8.97	37.77	
41	Hagere hiywet	41400	8.98	37.85	
42	Awash	11600	8.98	40.17	
43	Gedo	7700	9.02	37.45	
44	Addis abeba	2763500	9.03	38.74	
45	Ginch'i	14200	9.04	38.15	
46	Addis 'alem	10100	9.04	38.39	
47	Holeta genet	23400	9.07	38.5	
48	Burayu	13400	9.08	38.53	
49	Nek'emte	64600	9.09	36.53	
50	Asbe teferi	28500	9.09	40.86	
51	Bako	10600	9.14	37.05	
52	Sendafa	7500	9.15	39.02	
53	Gimbi	30200	9.18	35.83	
54	Hirna	12500	9.21	41.1	
55	Mi'eso	7800	9.23	40.75	
56	Harer	99600	9.31	42.13	
57	Deder	9100	9.31	41.44	
58	'alem maya	11500	9.4	42.01	
59	Nedjo	14900	9.5	35.5	
60	Shambu	15200	9.57	37.1	
61	Dire dawa	254500	9.59	41.86	
62	Fiche	26900	9.8	38.73	
63	Gebre guracha	14900	9.8	38.41	
64	Mendi	13500	9.8	35.1	
65	Abomsa	14400	9.99	39.98	

A.2.2 Ethiopia Cities and Towns

No	City	Population (in 2000)	Latitude (DD)	Longitude (DD)	
1	Moyale	14100	3.53	39.05	
2	Mega	7000	4.06	38.31	
3	Dollo odo	9600	4.18	42.08	
4	Yabelo	13800	4.89	38.08	
5	Negele	40400	5.34	39.56	
6	Hagere maryam	17100	5.63	38.23	
7	Jinka	16600	5.65	36.65	
8	Gidole	11000	5.65	37.37	
9	Shakiso	21100	5.75	38.91	
10	Bako	14000	5.79	36.55	

No	City	Population (in 2000)	Latitude (DD)	Longitude (DD)	
11	Kibre mengist	27700	5.88	38.98	
12	Arba minch	73200	6.04	37.55	
13	Yirga ch'efe	15500	6.16	38.19	
14	Ch'ench'a	7800	6.25	37.57	
15	Sawla	21100	6.3	36.88	
16	Dilla	45100	6.41	38.31	
17	Imi	10400	6.46	42.18	
18	Hagere selam	5300	6.49	38.51	
19	Wendo	15200	6.61	38.42	
20	Yirga alem	35000	6.75	38.41	
21	Leku	11600	6.88	38.44	
22	Soddo	53800	6.9	37.75	
23	Boditi	18000	6.96	37.86	
24	Dodola	18600	6.97	39.18	
25	Werder	11200	6.97	45.34	
26	Mizan teferi	14300	7	35.58	
27	Goba	45000	7.01	39.97	
28	Awassa	99100	7.06	38.47	
29	Areka	16500	7.08	37.7	
30	Kofele	9800	7.08	38.78	
31	Asasa	14600	7.11	39.19	
32	Ginir	16200	7.14	40.71	
33	Shashemenne	88500	7.2	38.59	
34	Teppi	14200	7.21	35.41	
35	Bonga	14600	7.28	36.24	
36	Alaba kulito	20300	7.32	38.08	
37	Negele	37600	7.36	38.67	
38	Hossa'ina	54300	7.55	37.85	
39	Bircot	9200	7.6	43.75	
40	Jimma	117600	7.68	36.83	
41	Sheh hussen	10600	7.75	40.7	
42	Agaro	33100	7.85	36.65	
43	Robe	35400	7.86	39.63	
44	Batu	26900	7.93	38.71	
45	Assela	61300	7.95	39.14	
46	Butajira	29500	8.12	38.37	
47	Mek'i	33100	8.15	38.82	
48	Huruta	12700	8.15	39.35	
49	Gore	9500	8.15	35.54	
50	Degeh bur	13100	8.22	43.57	

No	City	Population (in 2000)	Latitude (DD)	Longitude (DD)	
51	Gambela	24500	8.25	34.59	
52	Welk'it'e	20600	8.28	37.77	
53	Metu	31100	8.3	35.58	
54	Sire	10300	8.32	39.48	
55	Bedele	16000	8.45	36.35	
56	Wonji	12800	8.45	39.28	
57	Giyon	37000	8.53	37.97	
58	Dembi dolo	26700	8.53	34.79	
59	Adama	176800	8.55	39.27	
60	Mojo	32700	8.6	39.12	
61	Tulu bolo	10700	8.66	38.21	
62	Welench'iti	15700	8.67	39.43	
63	Debre zeyit	98700	8.75	38.99	
64	Gelemso	14600	8.82	40.52	
65	Metehara	16000	8.9	39.92	
66	Bedesa	14500	8.9	40.77	
67	Sebeta	19400	8.92	38.62	
68	Guder	12800	8.97	37.77	
69	Hagere hiywet	41400	8.98	37.85	
70	Awash	11600	8.98	40.17	
71	Gedo	7700	9.02	37.45	
72	Addis abeba	2763500	9.03	38.74	
73	Ginch'i	14200	9.04	38.15	
74	Addis 'alem	10100	9.04	38.39	
75	Holeta genet	23400	9.07	38.5	
76	Burayu	13400	9.08	38.53	
77	Nek'emte	64600	9.09	36.53	
78	Asbe teferi	28500	9.09	40.86	
79	Bako	10600	9.14	37.05	
80	Sendafa	7500	9.15	39.02	
81	Gimbi	30200	9.18	35.83	
82	Hirna	12500	9.21	41.1	
83	Mi'eso	7800	9.23	40.75	
84	Harer	99600	9.31	42.13	
85	Deder	9100	9.31	41.44	
86	Jijiga	40200	9.35	42.79	
87	'alem maya	11500	9.4	42.01	
88	Nedjo	14900	9.5	35.5	
89	Shambu	15200	9.57	37.1	
90	Dire dawa	254500	9.59	41.86	
91	Ankober	5200	9.59	39.73	
92	Debre birhan	52500	9.68	39.53	
93	Fiche	26900	9.8	38.73	
94	Gebre guracha	14900	9.8	38.41	
95	Mendi	13500	9.8	35.1	
96	Debre sina	9600	9.85	39.76	

No	City	Population (in 2000)	Latitude (DD)	Longitude (DD)	
97	Abomsa	14400	9.99	39.98	
98	Asosa	15800	10.07	34.53	
99	Dejen	12000	10.17	38.13	
100	Gewane	11500	10.17	40.65	
101	Debre mark'os	63600	10.34	37.72	
102	Bichena	16700	10.45	38.2	
103	Were ilu	7800	10.6	39.43	
104	Debre werk	10800	10.67	38.17	
105	Finote selam	18600	10.69	37.26	
106	Bure	18000	10.71	37.07	
107	Kemise	14500	10.72	39.87	
108	Chagni	23800	10.95	36.5	
109	Kembolcha	70900	11.08	39.74	
110	Mott'a	25000	11.08	37.87	
111	Dese	126300	11.13	39.63	
112	Bati	19200	11.18	40.02	
113	Dangla	21800	11.25	36.6	
114	Adet	16300	11.27	37.48	
115	Bahir dar	131800	11.56	37.37	
116	Asayita	20800	11.56	41.44	
117	Nefas mewcha	14500	11.73	38.47	
118	Dubti	24800	11.74	41.08	
119	semera		11.8	41.01	
120	Weldiya	36300	11.83	39.59	
121	Debre tabor	31900	11.86	38.01	
122	Werota	20400	11.92	37.7	
123	Shewa robit	19200	12.02	39.63	
124	Lalibela	11400	12.04	39.04	
125	Addis zemen	19200	12.12	37.78	
126	K'obo	30400	12.16	39.63	
127	Alamat'a	42700	12.42	39.56	
128	Korem	22700	12.5	39.53	
129	Gondar	147900	12.61	37.46	
130	Sek'ot'a	10600	12.64	39.03	
131	Maych'ew	27100	12.78	39.54	
132	Dabat	11800	12.99	37.76	
133	Debark	19400	13.16	37.9	
134	Abiy adi	10600	13.44	39.08	
135	Addi ark'ay	5000	13.45	38.06	
136	Mek'ele	133500	13.5	39.47	
137	Wik'ro	21100	13.79	39.59	
138	Endasilasie	42200	14.1	38.28	
139	Aksum	39600	14.13	38.72	
140	Adwa	39000	14.19	38.88	
141	Addigrat	66000	14.28	39.47	
142	Himora	19700	14.28	36.65	