MINISTRY OF EDUCATION AND TRAINING MINISTRY OF NATIONAL DEFENCE

MILITARY TECHNICAL ACADEMY

TRAN HUNG CUONG

DC ALGORITHMS IN NONCONVEX QUADRATIC PROGRAMMING AND APPLICATIONS IN DATA CLUSTERING

Major: Mathematical Foundations for Informatics

Code: 9 46 01 10

SUMMARY

DOCTORAL DISSERTATION MATHEMATICS

HANOI - 2021

The dissertation was written on the basis of the author's research works carried at Military Technical Academy.

Supervisors: **Prof. Dr.Sc. Nguyen Dong Yen**
Prof. Dr.Sc. Pham The Long

First referee: Prof. Dr.Sc. Pham Ky Anh

Second referee: Assoc. Prof. Nguyen Thi Thu Thuy

Third referee: Assoc. Prof. Nguyen Quang Uy

To be defended at the Jury of Military Technical Academy:

. . . .

. . . .

on . . . . , at . . . . o'clock . . . .

The dissertation is publicly available at:

- The National Library of Vietnam
- The Library of Military Technical Academy

Introduction

In this dissertation, we are concerned with several concrete topics in *DC programming* and *data mining*. Here and in the sequel, the word "DC" stands for **D**ifference of **C**onvex functions. Fundamental properties of DC functions and DC sets can be found in the book *"Convex Analysis and Global Optimization"* (2016) of Professor Hoang Tuy, who made fundamental contributions to global optimization. The whole Chapter 7 of that book gives a deep analysis of DC optimization problems and their applications in design calculation, location, distance geometry, and clustering. We refer to the books *"Global Optimization, Deterministic Approaches"* (1993) of R. Horst and H. Tuy, *"Optimization on Low Rank Nonconvex Structures"* (1997) of H. Konno, P. T. Thach, and H. Tuy, the dissertation *"Some Nonconvex Optimization Problems: Algorithms and Applications"* (2019) of P. T. Hoai, and the references therein for methods of global optimization and numerous applications. We will consider some algorithms for finding *locally optimal solutions* of optimization problems. Thus, techniques of global optimization, like the branch and bound method and the cutting plane method, will not be applied herein. Note that since global optimization algorithms are costly for many large-scale nonconvex optimization problems, local optimization algorithms play an important role in optimization theory and real world applications.

DC programming and *DC algorithms* (*DCA*, for brevity) treat the problem of minimizing a function $f = g - h$, with g, h being lower semicontinuous, proper, convex functions on $\mathbb{R}^n$, on the whole space. Usually, g and h are called *d.c. components* of f. The DCA are constructed on the basis of the DC programming theory and the *duality theory* of J. F. Toland. Details about DCA can be seen in the paper by Pham Dinh Tao and Le Thi Hoai An (Acta Math. Vietnam., 1997), and Hoang Ngoc Tuan's PhD Dissertation *"DC Algorithms and Applications in Nonconvex Quadratic Programing"* (2015).

The main applications of DC programming and DCA include: Nonconvex optimization, Image analysis, Data mining, Machine learning; see H. A. Le

Thi and T. Pham Dinh (Math. Program., 2018).

DCA has a tight connection with the *proximal point algorithm* (PPA, for brevity). One can apply PPA to solve monotone and pseudomonotone variational inequalities. Since the necessary optimality conditions for an optimization problem can be written as a variational inequality, PPA is also a solution method for solving optimization problems. In the paper by L. D. Muu and T. D. Quoc (Optimization, 2010), PPA is applied to mixed variational inequalities by using DC decompositions of the cost function. Linear convergence rate is achieved when the cost function is strongly convex. For nonconvex case, global algorithms are proposed to search a global solution.

Indefinite quadratic programming problems (IQPs for short) under linear constraints form an important class of optimization problems. IQPs have various applications (see, e.g., Bomze (1998)). Since the IQP is NP-hard (see Pardalos and Vavasis (1991), Bomze and Danninger (1994)), finding its *global solutions* remains a challenging question.

New results on the convergence and the convergence rate of DCA applied for the IQP problems are proved in this dissertation. We also study the asymptotic stability of the *Proximal DC decomposition algorithm* (which is called Algorithm B) for the given IQP problem. Numerical results together with an analysis of the influence of the decomposition parameter, as well as a comparison between the *Projection DC decomposition algorithm* (which is called Algorithm A) and Algorithm B are given.

According to Han, Kamber, and Pei (2012), Wu (2012), and Jain and Srivastava (2013), *data mining* is the process of discovering patterns in large data to extract information and transform it into an understandable structure for further use.

Cluster analysis or simply *clustering* is a technique dealing with problems of organizing a collection of patterns into clusters based on similarity. Cluster analysis is applied in different areas; see, e.g., Aggarwal and Reddy (2014), Kumar and Reddy (2017).

Clustering problems are divided into two categories: *constrained* clustering problems (see, e.g., Basu, Davidson, and Wagstaff (2009), Covões, Hruschka, and Ghosh (2013), Davidson and Ravi (2005)) and *unconstrained* clustering problems. We focus on studying some problems of the second category.

In *the **M**inimum **S**um-of-**S**quares **C**lustering* (MSSC for short) problems (see, e.g., (see Bock (1998), Brusco (2006), Costa, Aloise, and Mladenović, (2017), Du Merle, Hansen, Jaumard, and Mladenović (2000), Kumar and Reddy (2017), Le Thi and Pham Dinh (2009), Peng and Xiay (2005), Sherali and Desai (2005), Aragón Artacho, Fleming, and Vuong (2018)), one has to find a centroid system with the minimal *sum of the minimal of the squared Euclidean distances of the data points to the closest centroids*. The MSSC problems with the required numbers of clusters being larger or equal to 2 are NP-hard (see Aloise, Deshpande, Hansen, and Popat (2009)).

We establish a series of basic qualitative properties of the MSSC problem. We also analyze and develop solution methods for the MSSC problem. Among other things, we suggest several modifications for the incremental algorithms of Ordin and Bagirov (see Ordin and Bagirov (2015)) and of Bagirov (see Bagirov (2014)). We focus on Ordin and Bargirov's approaches, because they allow one to find good starting points, and they are efficient for dealing with large data sets. Properties of the new algorithms are obtained and preliminary numerical tests of those on real-world databases are shown. The finite convergence, the convergence, and the rate of convergence of solution methods for the MSSC problem are presented here for the first time.

So, this dissertation proves the convergence and the convergence rate of DCA applied to IQPs, establishes a series of basic qualitative properties of the MSSC problem, suggests several modifications for the incremental algorithms in the papers of Ordin and Bagirov (2015), and of Bagirov (2014), studies finite convergence, convergence, and the rate of convergence of the algorithms.

The dissertation has four chapters and a list of references.

Chapter 1 collects some basic notations and concepts from DC programming and DCA.

Chapter 2 considers an application of DCA to indefinite quadratic programming problems under linear constraints.

In Chapter 3, several basic qualitative properties of the MSSC problem are established.

Chapter 4 analyzes and develops some solution methods for the MSSC problem.

Chapter 1

Background Materials

In this chapter, we review some background materials on Difference-of-Convex Functions Algorithms (DCAs for brevity); see Pham Dinh and Le Thi (1997, 1998), Hoang Ngoc Tuan's PhD Dissertation (2015).

1.1 Basic Definitions and Some Properties

By $\mathbb{N}$ we denote the set of natural numbers, i.e., $\mathbb{N} = \{0, 1, 2, \ldots\}$. Consider the n-dimensional Euclidean vector space $X = \mathbb{R}^n$ which is equipped with the canonical inner product $\langle x, u\rangle := \sum_{i=1}^{n} x_i u_i$ for all vectors $x = (x_1, \ldots, x_n)$ and $u = (u_1, \ldots, u_n)$. Here and in the sequel, vectors in $\mathbb{R}^n$ are represented as rows of real numbers in the text, but they are interpreted as columns of real numbers in matrix calculations. The transpose of a matrix $A \in \mathbb{R}^{m\times n}$ is denoted by A^T. So, one has $\langle x, u\rangle = x^T u$. The norm in X is given by $\|x\| = \langle x, x\rangle^{1/2}$. Then, the dual space Y of X can be identified with X.

A function $\theta : X \to \overline{\mathbb{R}}$, where $\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty, -\infty\}$ denotes the set of generalized real numbers, is said to be proper if it does not take the value $-\infty$ and it is not equal identically to $+\infty$, i.e., there is some $x \in X$ with $\theta(x) \in \mathbb{R}$. The effective domain of θ is defined by $\operatorname{dom}\theta := \{x \in X : \theta(x) < +\infty\}$.

Let $\Gamma_0(X)$ be the set of all lower semicontinuous, proper, convex functions on X. The *Fenchel conjugate function* g^* of a function $g \in \Gamma_0(X)$ is defined by

$$g^*(y) = \sup\{\langle x, y\rangle - g(x) \mid x \in X\} \quad \forall\, y \in Y.$$

Denote by g^{**} the conjugate function of g^*, i.e.,

$$g^{**}(x) = \sup\{\langle x, y\rangle - g^*(y) \mid y \in Y\}.$$

Definition 1.1 The *subdifferential* of a convex function $\varphi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ at $u \in \operatorname{dom} \varphi$ is the set

$$\partial\varphi(u) := \{x^* \in \mathbb{R}^n \mid \langle x^*, x - u\rangle \leq \varphi(x) - \varphi(u) \;\; \forall x \in \mathbb{R}^n\}.$$

If $x \notin \operatorname{dom} \varphi$ then one puts $\partial\varphi(x) = \emptyset$.

Proposition 1.1 *The inclusion* $x \in \partial g^*(y)$ *is equivalent to the equality*

$$g(x) + g^*(y) = \langle x, y\rangle.$$

Proposition 1.2 *The inclusions* $y \in \partial g(x)$ *and* $x \in \partial g^*(y)$ *are equivalent.*

In the sequel, we use the convention $(+\infty)-(+\infty)=+\infty$.

Definition 1.2 The optimization problem

$$\inf\{f(x) := g(x) - h(x) \, : \, x \in X\}, \tag{P}$$

where g and h are functions belonging to $\Gamma_0(X)$, is called a *DC program*. The functions g and h are called *d.c. components* of f.

Definition 1.3 For any $g, h \in \Gamma_0(X)$, the DC program

$$\inf\{h^*(y) - g^*(y) \mid y \in Y\}, \tag{D}$$

is called the dual problem of (P).

Proposition 1.3 (Toland's Duality Theorem; see Pham Dinh and Le Thi (1998)) *The DC programs* (P) *and* (D) *have the same optimal value.*

Definition 1.4 One says that $\bar{x} \in \mathbb{R}^n$ is a *local solution* of (P) if the value $f(\bar{x}) = g(\bar{x}) - h(\bar{x})$ is finite (i.e., $\bar{x} \in \operatorname{dom} g \cap \operatorname{dom} h$) and there exists a neighborhood U of $\bar{x}$ such that

$$g(\bar{x}) - h(\bar{x}) \leq g(x) - h(x) \quad \forall x \in U.$$

If we can choose $U = \mathbb{R}^n$, then $\bar{x}$ is called a (global) solution of (P).

Proposition 1.4 (First-order optimality condition; see Pham Dinh and Le Thi (1997)) *If $\bar{x}$ is a local solution of* (P)*, then* $\partial h(\bar{x}) \subset \partial g(\bar{x})$.

Definition 1.5 A point $\bar{x} \in \mathbb{R}^n$ satisfying $\partial h(\bar{x}) \subset \partial g(\bar{x})$ is called a *stationary point* of (P).

Definition 1.6 A vector $\bar{x} \in \mathbb{R}^n$ is said to be a *critical point* of (P) if

$$\partial g(\bar{x}) \cap \partial h(\bar{x}) \neq \emptyset.$$

1.2 DCA Schemes

The main idea of the theory of DCAs in Pham Dinh and Le Thi (1997) is to decompose the given difficult DC program (P) into two sequences of convex programs (P_k) and (D_k) with $k \in \mathbb{N}$ which, respectively, approximate (P) and (D). Namely, every DCA scheme requires to construct two sequences $\{x^k\}$ and $\{y^k\}$ in an appropriate way such that, for each $k \in \mathbb{N}$, x^k is a solution of a convex program (P_k) and y^k is a solution of a convex program (D_k), and next properties are valid:

(i) The sequences $\{(g-h)(x^k)\}$ and $\{(h^*-g^*)(y^k)\}$ are decreasing;

(ii) Any cluster point $\bar{x}$ (resp. $\bar{y}$) of $\{x^k\}$ (resp., of $\{y^k\}$) is a critical point of (P) (resp., of (D)).

Based on above propositions, definitions and observations, we get a simplified version of DCA (which is called DCA Scheme 1). It can be found in Hoang Ngoc Tuan's PhD Dissertation (2015), and Pham Dinh and Le Thi (1997). The following DCA Scheme 2 includes a termination procedure.

Scheme 1.2

Input: *$f(x) = g(x) - h(x)$.*
Output: *Finite or infinite sequences $\{x^k\}$ and $\{y^k\}$.*
Step 1. Choose $x^0 \in \operatorname{dom} g$. Take $\varepsilon > 0$. Put $k = 0$.
Step 2.
Calculate y^k by solving the convex program (D_k)

$$\min\{h^*(y) - \langle x^k, y\rangle \mid y \in Y\}.$$

Calculate x^{k+1} by solving the convex program (P_k)

$$\min\{g(x) - \langle x, y^k\rangle \mid x \in X\}.$$

Step 3. If $||x^{k+1} - x^k|| \leq \varepsilon$ then stop, else go to Step 4. Step 4. $k := k+1$ and return to Step 2.

1.3 General Convergence Theorem

The general convergence theorem for DCA from the paper T. Pham Dinh, H. A. Le Thi, *"Convex analysis approach to d.c. programming: theory, algorithms and applications"* (Acta Math. Vietnam., 1997) is recalled in this section.

Four illustrative examples for the fundamental properties of DC programming and DCA are given in Chapter 1.

Chapter 2

Analysis of a Solution Algorithm in Indefinite Quadratic Programming

This chapter addresses the convergence and the asymptotical stability of iterative sequences generated by the Proximal DC decomposition algorithm (Algorithm B). We also analyze the influence of the decomposition parameter on the rates of convergence of DCA sequences and compare the performances of the Projection DC decomposition algorithm (Algorithms A) and Algorithm B upon randomly generated data sets.

2.1 Indefinite Quadratic Programs and DCAs

Consider *the indefinite quadratic programming problem under linear constraints* (called the IQP for brevity):

$$\min\Big\{f(x) := \frac{1}{2}x^TQx + q^Tx \mid Ax \geq b\Big\}, \tag{2.1}$$

where $Q \in \mathbb{R}^{n\times n}$ and $A \in \mathbb{R}^{m\times n}$ are given matrices, Q is symmetric, $q \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ are arbitrarily given vectors. The constraint set of the problem is $C := \big\{x \in \mathbb{R}^n \mid Ax \geq b\big\}$. Since x^TQx is an indefinite quadratic form, the objective function $f(x)$ may be nonconvex; hence (2.1) is a nonconvex optimization problem.

Following Pham Dinh, Le Thi, and Akoa (2008), to solve the IQP via a sequence of strongly convex quadratic programs, one decomposes $f(x)$ into the difference of two convex linear-quadratic functions

$$f(x) = \varphi(x) - \psi(x)$$

with $\varphi(x) = \frac{1}{2}x^T Q_1 x + q^T x$ and $\psi(x) = \frac{1}{2}x^T Q_2 x$, where $Q = Q_1 - Q_2$, Q_1 is a symmetric positive definite matrix and Q_2 is a symmetric positive semidefinite matrix. Then (2.1) is equivalent to the *DC program*

$$\min \Big\{ \frac{1}{2}x^T Q_1 x + q^T x - x^T Q_2 x^k \mid x \in C \Big\}.$$

Definition 2.1 For $x \in \mathbb{R}^n$, if there exists a multiplier $\lambda \in \mathbb{R}^m$ such that

$$\begin{cases} Qx + q - A^T\lambda = 0, \\ Ax \geq b, \quad \lambda \geq 0, \quad \lambda^T(Ax - b) = 0, \end{cases}$$

then x is said to be a *Karush-Kuhn-Tucker point* (a *KKT point*) of the IQP.

The smallest eigenvalue (resp., the largest eigenvalue) of Q is denoted by $\lambda_1(Q)$ (resp., by $\lambda_n(Q)$). The number ρ is called *the decomposition parameter*. We have the following iterative algorithms.

Algorithm A (*Projection DC decomposition algorithm*) can be found in the paper by Pham Dinh, Le Thi, and Akoa (2008).

Algorithm B. (*Proximal DC decomposition algorithm*) Fix a positive number $\rho > -\lambda_1(Q)$ and choose an initial point $x^0 \in \mathbb{R}^n$. For any $k \geq 0$, compute the unique solution, denoted by the point x^{k+1}, of the strongly convex quadratic minimization problem

$$\min \Big\{ \psi(x) := \frac{1}{2}x^T Q x + q^T x + \frac{\rho}{2}\|x - x^k\|^2 \mid Ax \geq b \Big\}. \tag{2.2}$$

The objective function of (2.2) can be written as $\frac{1}{2}x^T Q_1 x + q^T x - x^T Q_2 x^k$, where $Q_1 = Q + \rho E$ and $Q_2 = \rho E$.

One illustrative example for Algorithms A and B is given in this section.

2.2 Convergence and Convergence Rate of the Algorithm

Denote by C^* the KKT point set of (2.1). The convergence and the rate of convergence of Algorithm B, the Proximal DC decomposition algorithm, can be formulated as follows.

Theorem 2.1 *If* (2.1) *has a solution, then for each $x^0 \in \mathbb{R}^n$, the DCA sequence $\{x^k\}$ constructed by Algorithm B converges R-linearly to a KKT point of* (2.1)*, that is, there exists $\bar{x} \in C^*$ such that*

$$\limsup_{k \to \infty} \|x^k - \bar{x}\|^{1/k} < 1.$$

2.3 Asymptotical Stability of the Algorithm

The concept of asymptotical stability of a KKT point can be found in in Leong and Goh (2013). The main result of this section can be formulated as follows.

Theorem 2.2 *Consider Algorithm B and require additionally that $\rho > \|Q\|$. Suppose $\bar{x}$ is a locally unique solution of problem* (2.1)*. In that case, for any $\varepsilon > 0$ there exists $\delta > 0$ such that if $x^0 \in C \cap B(\bar{x}, \delta)$ and if $\{x^k\}$ is the DCA sequence generated by Algorithm B and the initial point x^0, then*

(a) $x^k \in C \cap B(\bar{x}, \varepsilon)$ *for any* $k \geq 0$*;*

(b) $x^k \to \bar{x}$ *as* $k \to \infty$*.*

In other words, $\bar{x}$ is asymptotically stable w.r.t. Algorithm B.

An illustrative example for asymptotical stability of Algorithm B is given in this section.

2.4 Influence of the Decomposition Parameter

Algorithm A and B were implemented in the Visual C++ 2010 environment. We have carried many numerical experiments which demonstrate that:

- The decomposition parameter greatly influences the convergence rate of DCA sequences. When decomposition parameter increases, the execution time is also increased.

- Algorithm B is more efficient and more stable than Algorithm A upon randomly generated data sets.

Chapter 3

Qualitative Properties of the Minimum Sum-of-Squares Clustering Problem

A series of basic qualitative properties of the minimum sum-of-squares clustering are established in this chapter.

3.1 Clustering Problems

*The **M**inimum **S**um-of-**S**quares **C**lustering* (MSSC for short) problem requires to partition a finite data set into a given number of clusters in order to minimize the sum of the squared Euclidean distances from each data point to the centroid of its cluster as small as possible.

Let $A = \{a^1, ..., a^m\}$ be a finite set of points (representing the data points to be grouped) in the n-dimensional Euclidean space $\mathbb{R}^n$. Given a positive integer k with $k \leq m$, one wants to partition A into disjoint subsets $A^1, \ldots, A^k$, called *clusters*, such that a *clustering criterion* is optimized.

If one associates to each cluster A^j a *center* (or *centroid*), denoted by $x^j \in \mathbb{R}^n$, then the following well-known *variance* or *SSQ (Sum-of-Squares) clustering criterion* (see, e.g., Bock (1998)). Thus, the above partitioning problem can be formulated as the constrained optimization problem

$$\min\Big\{\psi(x,\alpha) := \frac{1}{m}\sum_{i=1}^{m}\Big(\sum_{j=1}^{k}\alpha_{ij}\|a^i - x^j\|^2\Big) \mid x \in \mathbb{R}^{n\times k}, \\ \alpha = (\alpha_{ij}) \in \mathbb{R}^{m\times k},\ \alpha_{ij} \in \{0,1\}, \\ \sum_{j=1}^{k}\alpha_{ij} = 1,\ i = 1,\dots,m,\ j = 1,\dots,k\Big\}, \tag{3.1}$$

where the centroid system $x = (x^1,\dots,x^k)$ and the incident matrix $\alpha = (\alpha_{ij})$ are to be found. Since this model is a difficult *mixed integer programming problem*, one considers the following problem (see, e.g., Ordin and Bagirov (2015)):

$$\min\Big\{f(x) := \frac{1}{m}\sum_{i=1}^{m}\Big(\min_{j=1,\dots,k}\|a^i - x^j\|^2\Big) \mid x = (x^1,\dots,x^k) \in \mathbb{R}^{n\times k}\Big\}. \tag{3.2}$$

Put $I = \{1,\dots,m\}$ and $J = \{1,\dots,k\}$.

3.2 Basic Properties of the MSSC Problem

Given a vector $\bar{x} = (\bar{x}^1,\dots,\bar{x}^k) \in \mathbb{R}^{n\times k}$, we inductively construct k subsets $A^1,\dots,A^k$ of A in the following way. Put $A^0 = \emptyset$ and

$$A^j = \Big\{a^i \in A \setminus \Big(\bigcup_{p=0}^{j-1} A^p\Big) \mid \|a^i - \bar{x}^j\| = \min_{q\in J}\|a^i - \bar{x}^q\|\Big\} \tag{3.3}$$

for $j \in J$.

Definition 3.1 Let $\bar{x} = (\bar{x}^1,\dots,\bar{x}^k) \in \mathbb{R}^{n\times k}$. We say that the component $\bar{x}^j$ of $\bar{x}$ is *attractive* with respect to the data set A if the set

$$A[\bar{x}^j] := \Big\{a^i \in A \mid \|a^i - \bar{x}^j\| = \min_{q\in J}\|a^i - \bar{x}^q\|\Big\}$$

is nonempty. The latter is called the *attraction set* of $\bar{x}^j$.

Proposition 3.1 *If $(\bar{x},\bar{\alpha})$ is a solution of (3.1), then $\bar{x}$ is a solution of (3.2). Conversely, if $\bar{x}$ is a solution of (3.2), then the natural clustering defined by (3.3) yields an incident matrix $\bar{\alpha}$ such that $(\bar{x},\bar{\alpha})$ is a solution of (3.1).*

Proposition 3.2 *If $a^1,...,a^m$ are pairwise distinct points and $\{A^1,\dots,A^k\}$ is the natural clustering associated with a global solution $\bar{x}$ of (3.2), then A^j is nonempty for every $j \in J$.*

Theorem 3.1 *Both problems (3.1), (3.2) have solutions. If $a^1, ..., a^m$ are pairwise distinct points, then the solution sets are finite. Moreover, in that case, if $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k) \in \mathbb{R}^{n\times k}$ is a global solution of (3.2), then the attraction set $A[\bar{x}^j]$ is nonempty for every $j \in J$ and one has*

$$\bar{x}^j = \frac{1}{|I(j)|} \sum_{i\in I(j)} a^i, \tag{3.4}$$

where $I(j) := \{i \in I \mid a^i \in A[\bar{x}^j]\}$ with $|\Omega|$ denoting the number of elements of Ω.

Proposition 3.3 *If $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k) \in \mathbb{R}^{n\times k}$ is a global solution of (3.2), then the components of $\bar{x}$ are pairwise distinct, i.e., $\bar{x}^{j_1} \neq \bar{x}^{j_2}$ whenever $j_2 \neq j_1$.*

Theorem 3.2 *If $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k) \in \mathbb{R}^{n\times k}$ is a local solution of (3.2), then (3.4) is valid for all $j \in J$ whose index set $I(j)$ is nonempty, i.e., the component $\bar{x}^j$ of $\bar{x}$ is attractive w.r.t. the data set A.*

3.3 The k-means Algorithm

The k-means clustering algorithm (see, e.g., Aggarwal and Reddy (2014), Jain (2010), Kantardzic (2011), and MacQueen (1967)) is one of the most popular solution methods for (3.2).

An illustrative example is given in this section to show how the k-means algorithm is performed in practice.

3.4 Characterizations of the Local Solutions

Proposition 3.4 *One has $J_i(x) = \{j \in J \mid a^i \in A[x^j]\}$.*

Consider the following condition on the local solution x:

(C1) *The components of x are pairwise distinct, i.e., $x^{j_1} \neq x^{j_2}$ whenever $j_2 \neq j_1$.*

Definition 3.2 A local solution $x = (x^1, ..., x^k)$ of (3.2) that satisfies **(C1)** is called a *nontrivial local solution*.

Theorem 3.3 (Necessary conditions for nontrivial local optimality) *Suppose that $x = (x^1, ..., x^k)$ is a nontrivial local solution of (3.2). Then, for any $i \in I$, $|J_i(x)| = 1$. Moreover, for every $j \in J$ such that the attraction set $A[x^j]$ of x^j is nonempty, one has*

$$x^j = \frac{1}{|I(j)|} \sum_{i \in I(j)} a^i, \tag{3.5}$$

where $I(j) = \{i \in I \mid a^i \in A[x^j]\}$. For any $j \in J$ with $A[x^j] = \emptyset$, one has

$$x^j \notin \mathcal{A}[x], \tag{3.6}$$

where $\mathcal{A}[x]$ is the union of the balls $\bar{B}(a^p, \|a^p - x^q\|)$ with $p \in I$, $q \in J$ satisfying $p \in I(q)$.

Theorem 3.4 (Sufficient conditions for nontrivial local optimality) *Suppose that a vector $x = (x^1, ..., x^k) \in \mathbb{R}^{n \times k}$ satisfies condition* **(C1)** *and $|J_i(x)| = 1$ for every $i \in I$. If (3.5) is valid for any $j \in J$ with $A[x^j] \neq \emptyset$ and (3.6) is fulfilled for any $j \in J$ with $A[x^j] = \emptyset$, then x is a nontrivial local solution of (3.2).*

Three examples are given in this section. The first shows that a local solution of the MSSC problem need not be a global solution; the second one presents a complete description of the set of nontrivial local solutions; and the last one analyzes the convergence of the k-means algorithm.

3.5 Stability Properties

Now, let the data set $A = \{a^1, ..., a^m\}$ of the problem (3.2) be subject to change. Put $a = (a^1, ..., a^m)$ and observe that $a \in \mathbb{R}^{n \times m}$. Denoting by $v(a)$ the *optimal value* of (3.2), one has

$$v(a) = \min\{f(x) \mid x = (x^1, \ldots, x^k) \in \mathbb{R}^{n \times k}\}.$$

The *global solution set* of (3.2), denoted by $F(a)$, is given by

$$F(a) = \left\{x = (x^1, \ldots, x^k) \in \mathbb{R}^{n \times k} \mid f(x) = v(a)\right\}.$$

Definition 3.3 A family $\{I(j) \mid j \in J\}$ of pairwise distinct, nonempty subsets of I is said to be a *partition* of I if $\bigcup_{j \in J} I(j) = I$.

From now on, let $\bar{a} = (\bar{a}^1, ..., \bar{a}^m) \in \mathbb{R}^{n\times m}$ *be a fixed vector with the property that* $\bar{a}^1, ..., \bar{a}^m$ *are pairwise distinct.*

Theorem 3.5 (Local Lipschitz property of the optimal value function) *The optimal value function* $v : \mathbb{R}^{n\times m} \to \mathbb{R}$ *is locally Lipschitz at* $\bar{a}$*, i.e., there exist* $L_0 > 0$ *and* $\delta_0 > 0$ *such that*

$$|v(a) - v(a')| \leq L_0\|a - a'\|$$

for all a *and* a' *satisfying* $\|a - \bar{a}\| < \delta_0$ *and* $\|a' - \bar{a}\| < \delta_0$*.*

Theorem 3.6 (Local upper Lipschitz property of the global solution map) *The global solution map* $F : \mathbb{R}^{n\times m} \rightrightarrows \mathbb{R}^{n\times k}$ *is locally upper Lipschitz at* $\bar{a}$*, i.e., there exist* $L > 0$ *and* $\delta > 0$ *such that*

$$F(a) \subset F(\bar{a}) + L\|a - \bar{a}\|\bar{B}_{\mathbb{R}^{n\times k}}$$

for all a *satisfying* $\|a - \bar{a}\| < \delta$*. Here*

$$\bar{B}_{\mathbb{R}^{n\times k}} := \Big\{x = (x^1, \ldots, x^k) \in \mathbb{R}^{n\times k} \ \mid \ \sum_{j\in J}\|x^j\| \leq 1\Big\}$$

denotes the closed unit ball of the product space $\mathbb{R}^{n\times k}$*, which is equipped with the sum norm* $\|x\| = \sum_{j\in J}\|x^j\|$*.*

Theorem 3.7 (Aubin property of the local solution map) *Let* $\bar{x} = (\bar{x}^1, ..., \bar{x}^k)$ *be an element of* $F_1(\bar{a})$ *satisfying condition* **(C1)***, that is,* $\bar{x}^{j_1} \neq \bar{x}^{j_2}$ *whenever* $j_2 \neq j_1$*. Then, the local solution map* $F_1 : \mathbb{R}^{n\times m} \rightrightarrows \mathbb{R}^{n\times k}$ *has the Aubin property at* $(\bar{a}, \bar{x})$*, i.e., there exist* $L_1 > 0$*,* $\varepsilon > 0$*, and* $\delta_1 > 0$ *such that*

$$F_1(a) \cap B(\bar{x}, \varepsilon) \subset F_1(\widetilde{a}) + L_1\|a - \widetilde{a}\|\bar{B}_{\mathbb{R}^{n\times k}}$$

for all a *and* $\widetilde{a}$ *satisfying* $\|a - \bar{a}\| < \delta_1$ *and* $\|\widetilde{a} - \bar{a}\| < \delta_1$*.*

Chapter 4

Some Incremental Algorithms for the Clustering Problem

Solution methods for the minimum sum-of-squares clustering (MSSC) problem are analyzed and developed in this chapter.

4.1 Incremental Clustering Algorithms

One calls a clustering algorithm *incremental* if the number of the clusters increases step by step. As noted in (Ordin and Bagirov (2015)), the available numerical results demonstrate that *incremental clustering algorithms* (see, e.g., Bagirov (2008), Ordin and Bagirov (2015)) are efficient for dealing with large data sets.

4.2 Ordin-Bagirov's Clustering Algorithm

This section is devoted to the incremental heuristic algorithm of Ordin and Bagirov (2015) and some properties of the algorithm.

4.2.1 Basic constructions

Let ℓ be an index with $1 \leq \ell \leq k-1$ and let $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ be an approximate solution of (3.2), where k is replaced by ℓ. So, $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$

solves approximately the problem

$$\min\left\{f_\ell(x) := \frac{1}{m}\sum_{i=1}^{m}\Big(\min_{j=1,\dots,\ell}\|a^i - x^j\|^2\Big) \mid x = (x^1,\dots,x^\ell) \in \mathbb{R}^{n\times\ell}\right\}. \tag{4.1}$$

For every $i \in I$, put

$$d_\ell(a^i) = \min\left\{\|\bar{x}^1 - a^i\|^2, ..., \|\bar{x}^\ell - a^i\|^2\right\}.$$

Let $g(y) = f_{\ell+1}(\bar{x}^1, ..., \bar{x}^\ell, y)$. Then, the problem

$$\min\{g(y) \mid y \in \mathbb{R}^n\} \tag{4.2}$$

is called *the auxiliary clustering problem.* The objective function of (4.2) can be represented as $g(y) = g^1(y) - g^2(y)$, where

$$g^1(y) = \frac{1}{m}\sum_{i=1}^{m} d_\ell(a^i) + \frac{1}{m}\sum_{i=1}^{m}\|y - a^i\|^2 \tag{4.3}$$

is a smooth convex function and

$$g^2(y) = \frac{1}{m}\sum_{i=1}^{m}\max\{d_\ell(a^i), \|y - a^i\|^2\} \tag{4.4}$$

is a nonsmooth convex function. Consider the open set

$$Y_1 := \bigcup_{i\in I} B\big(a^i, d_\ell(a^i)\big) = \{y \in \mathbb{R}^n \mid \exists i \in I \text{ with } \|y - a^i\|^2 < d_\ell(a^i)\}.$$

By (4.3) and (4.4), we have

$$g(y) < \frac{1}{m}\sum_{i=1}^{m} d_\ell(a^i) \quad \forall y \in Y_1.$$

Therefore, any *iteration process* for solving (4.2) should start with a point $y^0 \in Y_1$.

To find an approximate solution of (3.2) where k is replaced by $\ell + 1$, i.e., the problem

$$\min\left\{f_{\ell+1}(x) := \frac{1}{m}\sum_{i=1}^{m}\Big(\min_{j=1,\dots,\ell+1}\|a^i - x^j\|^2\Big) \mid x = (x^1,\dots,x^{\ell+1}) \in \mathbb{R}^{n\times(\ell+1)}\right\}, \tag{4.5}$$

we use a procedure in Ordin and Bagirov (2015). The selection of 'good' starting points to solve (4.5) is controlled by two parameters: $\gamma_1 \in [0, 1]$ and $\gamma_2 \in [0, 1]$.

4.2.2 Version 1 of Ordin-Bagirov's algorithm

In this version, Ordin and Bagirov (2015) use the k-means first to find starting points for the auxiliary clustering problem (4.5), and then to find an approximate solution of the problem (3.2). The computation of a set of starting points to solve problem (4.5) is controlled by a parameter $\gamma_3 \in [1, \infty)$.

The input of Version 1 of Ordin-Bagirov's algorithm (called Alogrithm 4.1) is the data set $A = \{a^1, \ldots, a^m\}$, and the output is a centroid system $\{\bar{x}^1, \ldots, \bar{x}^k\}$, which is the approximate solution of (3.2).

Alogrithm 4.1 is based on Procedure 4.1, whose aim is to find starting points for (4.5). The input of the procedure is an approximate solution $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ of problem (4.1), $\ell \geq 1$, and the output is a set $\bar{A}_5$ of starting points to solve problem (4.5).

One illustrative example is given in this subsection.

4.2.3 Version 2 of Ordin-Bagirov's algorithm

We propose a new version of Ordin-Bagirov's algorithm, where we use the k-algorithm only one time.

The input of Version 2 (called Algorithm 4.2) consists of the parameters n, m, k, and the data set $A = \{a^1, \ldots, a^m\}$. The output are a centroid system $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k)$ of (3.2) and the corresponding clusters $A^1, ..., A^k$.

Algorithm 4.2 is based on Procedure 4.2 that finds starting points for (4.5). With an approximate solution $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ of problem (4.1) being the input, the procedure returns an approximate solution $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ of problem (4.5).

Some properties of Procedure 4.2 and Algorithm 4.2 are described in Theorems 4.1 and 4.2 below. We will need the following assumption:

(C2) *The data points $a^1, ..., a^m$ in the given data set A are pairwise distinct.*

Theorem 4.1 *Let ℓ be an index with $1 \leq \ell \leq k-1$ and let $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ be an approximate solution of problem (3.2) where k is replaced by ℓ. If* **(C2)** *is fulfilled and the centroids $\bar{x}^1, ..., \bar{x}^\ell$ are pairwise distinct, then the centroids $\hat{x}^1, \ldots, \hat{x}^{\ell+1}$ of the approximate solution $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ of (4.5), which is* *obtained by Procedure 4.2, are also pairwise distinct.*

Theorem 4.2 *If* **(C2)** *is fulfilled, then the centroids $\bar{x}^1, \ldots, \bar{x}^k$ of the centroid system $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k)$, which is obtained by Algorithm 4.2, are pairwise distinct.*

One illustrative example is given in this subsection.

4.2.4 The ε-neighborhoods technique

The ε-neighborhoods technique (see Ordin and Bagirov (2015)) allows one to reduce the computation volume of Algorithm 4.1 (as well as that of Algorithm 4.2, or another incremental clustering algorithm), when it is applied to large data sets.

4.3 Incremental DC Clustering Algorithms

Some incremental clustering algorithms based on Ordin-Bagirov's clustering algorithm and the DC algorithms of Pham Dinh and Le Thi are discussed and compared in this section.

4.3.1 Bagirov's DC Clustering Algorithm and Its Modification

In Step 5 of Procedure 4.1 and Step 4 of Algorithm 4.1, one applies KM. Bagirov (2014) suggested an improvement of Algorithm 4.1 by using DCA (see Le Thi, Belghiti, and Pham Dinh (2007), Pham Dinh and Le Thi (1997, 2009)) twice at each clustering level $\ell \in \{1, \ldots, k\}$. Consider a DC program of the form

$$\min \{\varphi(x) := g(x) - h(x) \mid x \in \mathbb{R}^n\}, \tag{4.6}$$

where g, h are continuous convex functions on $\mathbb{R}^n$. If $\bar{x} \in \mathbb{R}^n$ is a *local solution* of (4.6), then by the necessary optimality condition in DC programming one has $\partial h(\bar{x}) \subset \partial g(\bar{x})$. The DCA scheme for solving (4.6) is shown in Procedure 4.3. The input of the procedure is a starting point $x^1 \in \mathbb{R}^n$, and the ouput is an approximate solution x^p of (4.6).

If $\partial h(x^p)$ is a singleton, then the condition $y^p = \nabla g(x^p)$ is an *exact requirement* for x^p to be a stationary point. From our experience of implementing Procedure 4.3, we know that the stopping criterion $y^p = \nabla g(x^p)$ greatly delays the computation. So, it is reasonable to employ another stopping criterion.

Combining Procedure 4.3 with the above analysis, one obtains Procedure 4.4, which is a modified version of Procedure 4.3 with the new stopping criterion $\|x^{p+1} - x^p\| \leq \varepsilon$, where ε is a small positive constant. The criterion guarantees that Procedure 4.4 always stops after a finite number of steps.

Now we turn our attention back to problem (4.2) whose objective function has the DC decomposition $g(y) = g^1(y) - g^2(y)$, where $g^1(y)$ and $g^2(y)$ are given respectively by (4.3) and (4.4). Specializing Procedure 4.3 for the auxiliary clustering problem (4.2), one gets Procedure 4.5, which is a DCA scheme for solving (4.2). The input and output of Procedure 4.5 are the same as those of Procedure 4.3. Procedure 4.6 is a modified version of Procedure 4.5 with the same stopping criterion as the one in Procedure 4.4.

Theorem 4.3 *The following assertions hold true:*

(i) *The computation by* Procedure 4.5 *may not terminate after finitely many steps.*

(ii) *The computation by* Procedure 4.6 *with* $\varepsilon = 0$ *may not terminate after finitely many steps.*

(iii) *The computation by* Procedure 4.6 *with* $\varepsilon > 0$ *always terminates after finitely many steps.*

(iv) *If the sequence* $\{x^p\}$ *generated by* Procedure 4.6 *with* $\varepsilon = 0$ *is finite) then one has* $x^{p+1} \in \mathcal{B}$*, where* $\mathcal{B} = \{b_\Omega \mid \emptyset \neq \Omega \subset A\}$ *and* b_Ω *is the barycenter of a nonempty subset* $\Omega \subset A$*, i.e.,* $b_\Omega = \frac{1}{|\Omega|} \sum_{a^i \in \Omega} a^i$.

(v) *If the sequence* $\{x^p\}$ *generated by* Procedure 4.6 *with* $\varepsilon = 0$ *is infinite, then it converges to a point* $\bar{x} \in \mathcal{B}$

Theorem 4.4 *If the sequence* $\{x^p\}$ *generated by* Procedure 4.6 *with* $\varepsilon = 0$ *is infinite, then it converges* $Q-$*linearly to a point* $\bar{x} \in \mathcal{B}$*. More precisely, one has*

$$\|x^{p+1} - \bar{x}\| \leq \frac{m-1}{m} \|x^p - \bar{x}\|$$

for all p *sufficiently large.*

Now, we can describe a DCA to solve problem (3.2), whose objective function has the DC decomposition $f(x) = f^1(x) - f^2(x)$, where $f^1(x)$ and $f^2(x)$ are defined by

$$f^1(x) := \frac{1}{m} \sum_{i \in I} \Big(\sum_{j \in J} \|a^i - x^j\|^2 \Big)$$

and

$$f^2(x) := \frac{1}{m} \sum_{i \in I} \Big(\max_{j \in J} \sum_{q \in J \setminus \{j\}} \|a^i - x^q\|^2 \Big).$$

Procedure 4.7 is a DCA scheme for solving (4.5) (see Bagirov (2014)). The input of the procedure is an approximate solution $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$, and the output of it is a set $\widehat{A}_5 \subset \mathbb{R}^{n \times (\ell+1)}$ consisting of some approximate solutions $x^{p+1} = (x^{p+1,1}, \ldots, x^{p+1,\ell+1})$ of (4.5).

Combining Procedure 4.5 with Procedure 4.7, one obtains the DC incremental clustering algorithm of Bagirov (called Algorithm 4.3) to solve (3.2). The algorithm gets parameters n, m, k, and the data set $A = \{a^1, \ldots, a^m\}$ as input. Its output is a centroid system $\{\bar{x}^1, \ldots, \bar{x}^k\}$ and the corresponding clusters $\{A^1, \ldots, A^k\}$.

In Procedure 4.7, the condition $x^{p+1,j} = x^{p,j}$ for $j \in \{1, \ldots, \ell + 1\}$ at Step 4 is an exact requirement which slows down the speed of computation by Algorithm 4.3. So, we prefer to use the stopping criterion $\|x^{p+1,j} - x^{p,j}\| \leq \varepsilon$, where ε is a small positive constant. The condition is used in Procedure 4.8, which is a modified version of Procedure 4.7 with the input and output of it being the same as those of Procedure 4.7.

Based on Procedures 4.6 and 4.8, we can propose Algorithm 4.4, which is an improvement for Algorithm 4.3. Algorithm 4.4 produces a centroid system $\{\bar{x}^1, \ldots, \bar{x}^k\}$ and the corresponding clusters $\{A^1, \ldots, A^k\}$ with input being the parameters n, m, k, and the data set $A = \{a^1, \ldots, a^m\}$. Unlike Algorithms 1 and 2, both Algorithms 4.3 and 4.4 do not depend on the parameter γ_3.

One illustrative example for Algorithm 4.4 is given in this subsection. Another example of this subsection shows the efficiency of Algorithms 4.4 compared with that of Algorithms 1 and 2.

Theorem 4.5 *The following assertions hold true:*

(i) *The computation by* Algorithm 4.3 *may not terminate after finitely many steps.*

(ii) *The computation by* Algorithm 4.4 *with* $\varepsilon = 0$ *may not terminate after finitely many steps.*

(iii) *The computation by* Algorithm 4.4 *with* $\varepsilon > 0$ *always terminates after finitely many steps.*

(iv) *If the computation by* Procedure 4.8 *with* $\varepsilon = 0$ *terminates after finitely many steps then, for every* $j \in \{1, \dots, \ell + 1\}$, *one has* $x^{p+1,j} \in \mathcal{B}$.

(v) *If the computation by* Procedure 4.8 *with* $\varepsilon = 0$ *does not terminate after finitely many steps then, for every* $j \in \{1, \dots, \ell + 1\}$, *the sequence* $\{x^{p,j}\}$ *converges to a point* $\bar{x}^j \in \mathcal{B}$.

4.3.2 The Third DC Clustering Algorithm

To accelerate the computation speed of Algorithm 4.4, one can apply the DCA in the inner loop and apply the k-means algorithm in the outer loop. First, using the DCA scheme in Procedure 4.6 instead of the k-means algorithm, we can modify Procedure 1 and get Procedure 4.9 including inner loop with DCA. An approximate solution $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ of problem (4.1) is the input of Procedure 4.9. Its output is a set $\bar{A}_5$ of starting points to solve problem (4.5).

Based on Procedure 4.9, one has Algorithm 4.5, which includes DCA in the inner loop and k-means algorithm in the outer loop. The input of Algorithm 4.5 are the parameters n, m, k, and the data set $A = \{a^1, \dots, a^m\}$, and the output of it are a centroid system $\{\bar{x}^1, \dots, \bar{x}^k\}$ and the corresponding clusters $\{A^1, \dots, A^k\}$.

An illustrative example is given in this subsection.

4.3.3 The Fourth DC Clustering Algorithm

In Algorithm 4.2, which is Version 2 of Ordin-Bagirov's Algorithm, one applies the k-means algorithm to find an approximate solution of (4.5). Applying the DCA instead, we obtain Algorithm 4.6, which is a DC algorithm.

The input of the algorithm are the parameters n, m, k, and the data set $A = \{a^1, \ldots, a^m\}$, and its output are the set of k cluster centers $\{\bar{x}^1, \ldots, \bar{x}^k\}$ and the corresponding clusters $A^1, ..., A^k$.

Algorithm 4.6 is based on Procedure 4.10, which uses an approximate solution $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ of problem (4.1) as an input and produces an approximate solution $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ of problem (4.5) as the output.

An illustrative example is given in this subsection.

4.4 Numerical Tests

Using several well-known real-world data sets, we have tested the efficiencies of the five Algorithms 4.1, 4.2, 4.4, 4.5, and 4.6 above, and compared them with that of the k-means Algorithm. Namely, 8 real-world data sets, including 2 small data sets (with $m \leq 200$) and 6 medium size data sets (with $200 < m \leq 6000$), have been used in our numerical experiments.

To sum up, in term of the best value of the cluster function, Algorithm 4.2 is preferable to Algorithm 4.1, Algorithm 4.5 is preferable to Algorithm 4.6, Algorithm 4.2 is preferable to KM, and Algorithm 4.5 is also preferable to KM.

General Conclusions

In this dissertation, we have applied DC programming and DCAs to analyze a solution algorithm for the indefinite quadratic programming problem (IQP problem). We have also used different tools from set-valued analysis and optimization theory to study qualitative properties (solution existence, finiteness, and stability) of the minimum sum-of-squares clustering problem (MSSC problem) and develop some solution methods for this problem.

Our main results include:

1) The R-linear convergence of the Proximal DC decomposition algorithm (Algorithm B) and the asymptotic stability of that algorithm for the given IQP problem, as well as the analysis of the influence of the decomposition parameter on the rate of convergence of DCA sequences;

2) The solution existence theorem for the MSSC problem together with the necessary and sufficient conditions for a local solution of the problem, and three fundamental stability theorems for the MSSC problem when the data set is subject to change;

3) The analysis and development of the heuristic incremental algorithm of Ordin and Bagirov together with three modified versions of the DC incremental algorithms of Bagirov, including some theorems on the finite convergence and the $Q-$linear convergence, as well as numerical tests of the algorithms on several real-world databases.

In connection with the above results, we think that the following research topics deserve further investigations:

- Qualitative properties of the clustering problems with L_1-distance and Euclidean distance;

- Incremental algorithms for solving the clustering problems with L_1-distance and Euclidean distance;

- Booted DC algorithms to increase the computation speed;

- Qualitative properties and solution methods for constrained clustering problems for the definition of constrained clustering problems and two basic solution methods.

List of Author's Related Papers

1. T. H. Cuong, Y. Lim, N. D. Yen, *Convergence of a solution algorithm in indefinite quadratic programming,* Preprint (arXiv:1810.02044), submitted.
2. T. H. Cuong, J.-C. Yao, N. D. Yen, *Qualitative properties of the minimum sum-of-squares clustering problem,* Optimization **69** (2020), No. 9, 2131–2154. (SCI-E; IF 1.206, Q1-Q2, H-index 37; MCQ of 2019: 0.75)
3. T. H. Cuong, J.-C. Yao, N. D. Yen, *On some incremental algorithms for the minimum sum-of-squares clustering problem. Part 1: Ordin and Bagirov's incremental algorithm,* Journal of Nonlinear and Convex Analysis **20** (2019), No. 8, 1591–1608. (SCI-E; 0.710, Q2-Q3, H-index 18; MCQ of 2019: 0.56)
4. T. H. Cuong, J.-C. Yao, N. D. Yen, *On some incremental algorithms for the minimum sum-of-squares clustering problem. Part 2: Incremental DC algorithms,* Journal of Nonlinear and Convex Analysis **21** (2020), No. 5, 1109–1136. (SCI-E; 0.710, Q2-Q3, H-index 18; MCQ of 2019: 0.56)

The results of this dissertation have been presented at

- International Workshop "Some Selected Problems in Probability Theory, Graph Theory, and Scientific Computing" (February 16–18, 2017, Hanoi Pedagogical University 2, Vinh Phuc, Vietnam);

- The 7th International Conference on High Performance Scientific Computing (March 19–23, 2018, Hanoi, Vietnam).

- 2019 Winter Workshop on Optimization (December 12–13, 2019, National Center for Theoretical Sciences, Taipei, Taiwan).