MINISTRY OF EDUCATION AND TRAINING

LE QUY DON TECHNICAL UNIVERSITY

TRAN HUNG CUONG

DC ALGORITHMS IN NONCONVEX QUADRATIC PROGRAMMING AND APPLICATIONS IN DATA CLUSTERING

DOCTORAL DISSERTATION MATHEMATICS

HANOI - 2021

MINISTRY OF EDUCATION AND TRAINING

LE QUY DON TECHNICAL UNIVERSITY

TRAN HUNG CUONG

DC ALGORITHMS IN NONCONVEX QUADRATIC PROGRAMMING AND APPLICATIONS IN DATA CLUSTERING

DOCTORAL DISSERTATION

Major: Mathematical Foundations for Informatics

Code: 9 46 01 10

RESEARCH SUPERVISIORS:

1. Prof. Dr.Sc. Nguyen Dong Yen

2. Prof. Dr.Sc. Pham The Long

HANOI - 2021

Confirmation

This dissertation was written on the basis of my research works carried out at the Le Quy Don Technical University, under the guidance of Prof. Nguyen Dong Yen and Prof. Pham The Long. All the results presented in this dissertation have got agreements of my coauthors to be used here.

June 15, 2021

The author

Tran Hung Cuong

Acknowledgments

I would like to express my deep gratitude to my advisor, Professor Nguyen Dong Yen and Professor Pham The Long, for their careful and effective guidance.

I would like to thank the board of directors of Le Quy Don Technical University for providing me with pleasant working conditions.

I am grateful to the leaders of Hanoi University of Industry, the Faculty of Information Technology, and my colleagues, for granting me various financial supports and/or constant help during the three years of my PhD study.

I am sincerely grateful to Prof. Jen-Chih Yao from Department of Applied Mathematics, National Sun Yat-sen University, Taiwan, and Prof. Ching-Feng Wen from Research Center for Nonlinear Analysis and Optimization, Kaohsiung Medical University, Taiwan, for granting several short-termed scholarships for my doctorate studies.

I would like to thank the following experts for their careful readings of this dissertation and for many useful suggestions which have helped me to improve the presentation: Prof. Dang Quang A, Prof. Pham Ky Anh, Prof. Le Dung Muu, Assoc. Prof. Phan Thanh An, Assoc. Prof. Truong Xuan Duc Ha, Assoc. Prof. Luong Chi Mai, Assoc. Prof. Tran Nguyen Ngoc, Assoc. Prof. Nguyen Nang Tam, Assoc. Prof. Nguyen Thi Thu Thuy, Assoc. Prof. Nguyen Van Tuyen, Assoc. Prof. Nguyen Quang Uy, Dr. Duong Thi Viet An, Dr. Bui Van Dinh, Dr. Vu Van Dong, Dr. Hoang Nam Dung, Dr. Phan Thi Hai Hong, Dr. Nguyen Ngoc Luan, Dr. Ngo Huu Phuc, Dr. Le Xuan Thanh, Dr. Le Quang Thuy, Dr. Nguyen Thi Toan, Dr. Ha Chi Trung, Dr. Hoang Ngoc Tuan.

I am so much indebted to my family for their love, support and encouragement, not only in the present time, but also in the whole my life. With love and gratitude, I dedicate this dissertation to them.

Contents

Table of Notations

$\mathbb{N} := \{0, 1, 2, \ldots\}$	the set of natural numbers
$\emptyset$	empty set
$\mathbb{R}$	the set of real numbers
$\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty, -\infty\}$	the set of generalized real numbers
$\mathbb{R}^n$	n-dimensional Euclidean vector space
$\mathbb{R}^{m \times n}$	set of $m \times n$-real matrices
(a, b)	set of $x \in \mathbb{R}$ with $a < x < b$
$[a, b]$	set of $x \in \mathbb{R}$ with $a \leq x \leq b$
$\langle x, y \rangle$	canonical inner product
$\lvert x \rvert$	absolute value of $x \in \mathbb{R}$
$\lVert x \rVert$	the Euclidean norm of a vector x
E	the $n \times n$ unit matrix
A^T	transposition of a matrix A
$\operatorname{pos} \Omega$	convex cone generated by Ω
$T_C(x)$	tangent cone to C at $x \in C$
$N_C(x)$	normal cone to C at $x \in C$
$d(x, \Omega)$	distance from x to Ω
$\{x^k\}$	sequence of vectors
$x^k \to x$	x^k converges to x in norm topology
$\liminf_{k \to \infty} \alpha_k$	lower limit of a sequence $\{\alpha_k\}$ of real numbers
$\limsup_{k \to \infty} \alpha_k$	upper limit of a sequence $\{\alpha_k\}$ of real numbers

χ_C	indicator function of a set C
$\varphi : \mathbb{R}^n \to \overline{\mathbb{R}}$	extended-real-valued function
$\operatorname{dom} \varphi$	effective domain of φ
$\partial \varphi(x)$	subdifferential of φ at x
$\varphi^* : \mathbb{R}^n \to \overline{\mathbb{R}}$	Fenchel conjugate function of φ
$\Gamma_0(X)$	the set of all lower semicontinuous, proper, convex functions on $\mathbb{R}^n$
sol(P)	the set of the solutions of problem (P)
loc(P)	the set of the local solutions of problem (P)
DC	Difference-of-Convex functions
DCA	DC algorithm
PPA	proximal point algorithm
IQP	indefinite quadratic programming
KKT	Karush-Kuhn-Tucker
C^*	the KKT point set of IQP
$\mathcal{S}$	the global solution set of IQP
MSSC	the minimum-sum-of-square clustering
KM	k-means algorithm

Introduction

0.1 Literature Overview and Research Problems

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 [94] 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 [37,46], the dissertation [36], 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.

First, let us begin with some facts about *DC programming*.

As noted in [93], "*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. It was Pham Dinh Tao who suggested a general DCA theory, which has been developed intensively by him and Le Thi Hoai An, starting from their fundamental paper [77] published in Acta Mathematica Vietnamica in 1997."

The interested reader is referred to the comprehensive survey paper of Le Thi and Pham Dinh [55] on the thirty years (1985–2015) of the development

of the DC programming and DCA, where as many as 343 research works have been commented and the following remarks have been given: "DC programming and DCA were the subject of several hundred articles in the high ranked scientific journals and the high-level international conferences, as well as various international research projects, and were the methodological basis of more than 50 PhD theses. About 100 invited symposia/sessions dedicated to DC programming and DCA were presented in many international conferences. The ever-growing number of works using DC programming and DCA proves their power and their key role in nonconvex programming/global optimization and many areas of applications."

DCA has been successfully applied to many large-scale DC optimization problems and proved to be more robust and efficient than related standard methods; see [55]. The main applications of DC programming and DCA include:

- *Nonconvex optimization problems*: The trust-region subproblems, indefinite quadratic programming problems,...

- *Image analysis*: Image analysis, signal and image restoration.

- *Data mining and Machine learning*: data clustering, robust support vector machines, learning with sparsity.

DCA has a tight connection with the *proximal point algorithm* (PPA, for brevity). One can apply PPA to solve monotone and pseudomonotone variational inequalities (see, e.g., [85] and [89] and the references therein). 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 [69], 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. In the 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., [16, 29]). In general, the constraint set of an IQP can be unbounded. Therefore, unlike the case of the trust-region subproblem (see, e.g., [58]), the boundedness of the iterative sequence generated by a DCA and a starting point for a given IQP require additional investigations.

For a general IQP, one can apply [82] the *Projection DC decomposition algorithm* (which is called Algorithm A) and the *Proximal DC decomposition algorithm* (which is called Algorithm B). Le Thi, Pham Dinh, and Yen [57] have shown that DCA sequences generated by Algorithm A converge to a locally unique solution if the initial points are taken from a neighborhood of it, and DCA sequences generated by either Algorithm A or Algorithm B are all bounded if a condition guaranteeing the solution existence of the given problem is satisfied. By using error bounds for affine variational inequalities, Tuan [92] has proved that any iterative sequence generated by Algorithm A is R-linearly convergent, provided that the original problem has solutions. His result solves in the affirmative the first part of the conjecture stated in [57, p. 489]. It is of interest to know whether results similar to those of [57] and [92] can be estanlished for Algorithm B, or not.

Now, we turn our attention to *data mining*.

Han, Kamber, and Pei [32, p. xxiii] have observed that "The computerization of our society has substantially enhanced our capabilities for both generating and collecting data from diverse sources. A tremendous amount of data has flooded almost every aspect of our lives. This explosive growth in stored or transient data has generated an urgent need for new techniques and automated tools that can intelligently assist us in transforming the vast amounts of data into useful information and knowledge. This has led to the generation of a promising and flourishing frontier in computer science called data mining, and its various applications. Data mining, also popularly referred to as knowledge discovery from data (KDD), is the automated or convenient extraction of patterns representing knowledge implicitly stored or captured in large databases, data warehouses, the Web, other massive information repositories, or data streams." According to Wu [97, p. 1], the phrase "data mining", which describes the activity that attempts to extract interesting patterns from some data source, appeared in the late eighties of the last century.

Jain and Srivastava [40] have noted that *data mining*, as a scientific theory, is an interdisciplinary subfield of computer science which involves computational processes of patterns discovery from large data sets. The goal of such an advanced analysis process is to extract information from a data set and transform it into an understandable structure for further use. The methods

of data mining are at the juncture of artificial intelligence, machine learning, statistics, database systems, and business intelligence. In other words, data mining is about solving problems by analyzing the data already present in the related databases. As explained in [32, pp.15–22], *data mining functionalities* include

- characterization and discrimination;
- the mining of frequent patterns, associations, and correlations;
- classification and regression;
- clustering analysis;
- outlier analysis.

Cluster analysis or simply *clustering* is a technique dealing with problems of organizing a collection of patterns into clusters based on similarity. So, clustering can be considered a concise model of the data which can be interpreted in the sense of either a summary or a generative model. Cluster analysis is applied in different areas such as image segmentation, information retrieval, pattern recognition, pattern classification, network analysis, vector quantization and data compression, data mining and knowledge discovery business, document clustering and image processing (see, e.g., [1, p. 32] and [48]). For basic concepts and methods of cluster analysis, we refer to [32, Chapter 10].

Clustering problems are divided into two categories: *constrained* clustering problems (see, e.g., [14, 23, 24]) and *unconstrained* clustering problems. We will focus on studying some problems of the second category. Different criteria are used for unconstrained problems. For example, Tuy, Bagirov, and Rubinov [95] used the DC programming approach and the branch and bound method to solve globally the problem of finding a centroid system with the minimal *sum of the Euclidean distances of the data points to the closest centroids*. Recently, Bagirov and Mohebi [8] and Bagirov and Taher [10] solved a similar problem where L_1-*distances* are used instead of the above Euclidean distances. The first paper applies a hyperbolic smoothing technique, while the second one relies on DC programming. Since the just mentioned problems are nonconvex, it is very difficult to find global solutions when the data sets are large.

In *the* ***M****inimum* ***S****um-of-****S****quares* ***C****lustering* (MSSC for short) problems

(see, e.g., [5, 11, 15, 18, 22, 28, 44, 48, 60, 75, 87]), one has to find a centroid system with the minimal *sum of squared Euclidean distances of the data points to the closest centroids*. Since the square of the Euclidean distance from a moving point to a fixed point is a smooth function, the MSSC problems have attracted much more attention than the clustering problems which aim at minimizing the sum of the minimum 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 [3]. This means that solving them globally in a polynomial time is not realistic. Therefore, various methods have been proposed to find local solutions of the MSSC problems: the k-means algorithm and its modifications, the simulated annealing method, variable neighborhood search method, genetic algorithms, branch and bound algorithms, cutting plane algorithms, interior point algorithms, etc.; see [76] and references therein. Of course, among the local solutions, those with smaller objective functions are more preferable.

Algorithms proposed for solving the MSSC problem in the past 5 decades can be divided into the following groups [71]:

- *Clustering algorithms based on deterministic optimization techniques*: The MSSC problem is a nonconvex optimization problem, therefore different global and local optimization algorithms were applied to solve it. The dynamic programming, the interior point method, the cutting plane method are local methods (see, e.g., [28, 71, 75] and the references therein). Global search methods include the branch and bound and the neighborhood search methods [18, 27, 34, 47].

- *Clustering algorithms relied on heuristics*: Since above mentioned algorithms are not efficient to solve MSSC problems with large data sets, various heuristic algorithms have been developed. These heuristics include k-means algorithms [66] and their variations such as h-means, j-means [35, 76]. However, these algorithms are very sensitive to the choice of initial centroid system. Hence, Ordin and Bagirov [71] have proposed a heuristic algorithm based on control parameters to find good initial points, which make the value of objective function at the resulted centroid systems smaller.

- *Heuristics based on the incremental approach*: These algorithms start with the computation of the centroid of the whole data set and attempt to optimally add one new centroid at each stage. This means that one creates a k-th centroid from the $(k-1)$ available centroids. The global k-means,

modified global k-means, and fast global k-means are representatives of the algorithms of this type [6, 11, 12, 33, 44, 49, 61, 98].

- *Clustering algorithms based on DC programming*: Such an algorithm starts with representing the objective function of the MSSC problem as a difference of two convex functions (see e.g. [7, 11, 42, 44, 51, 52]). Le Thi, Belghiti, and Pham Dinh [51] suggested an algorithm based on DC programming for the problem. They also showed how to find a good starting point for the algorithm by combining the k-means algorithm and a procedure related to DC programming. Based on a suitable penalty function, another version of the above algorithm was given in [52]. Bagirov [7] suggested a method which combines an heuristic algorithm, and an incremental algorithm with DC algorithms to solve the MSSC problem. The purpose of this combination is to find good starting points, work effectively with large data sets, and improve the speed of computation.

It is well known that a deep understanding on qualitative properties of an optimization problem is very helpful for its numerical solution. To our knowledge, apart from the fundamental necessary optimality condition given recently by Ordin and Bagirov [71], qualitative properties of the MSSC problem have not been addressed in the literature until now. Thus, it is of interest to study the solution existence of the MSSC problem, chracterizations of the global and local solutions of the problem, as well as its stability properties when the data set is subject to change. In addition, it is worthy to analyze the heuristic incremental algorithm of Ordin and Bagirov and the DC incremental algorithm of Bagirov, and propose some modifications. Numerical tests of the algorithms on real-world databases are also important.

0.2 The Subjects of Research

- Indefinite quadratic programming problems under linear constraints;
- The Minimum Sum-of-Squares Clustering problems with data sets consisting of finitely many data points in Euclidean spaces.
- Solution algorithms for Minimum Sum-of-Squares Clustering problems, where the number of clusters is given in advance.

0.3 The Range of Research

- Qualitative properties of the related nonconvex optimization problems;
- Algorithms for finding local solutions;

• Numerical tests of the algorithms on radomly generated indefinite quadratic programming problems and Minimum Sum-of-Squares Clustering problems with several real-world databases.

0.4 The Main Results

We will prove that, for a general IQP, any iterative sequence generated by Algorithm B converges R-linearly to a Karush-Kuhn-Tucker point, provided that the problem has a solution. Our another major result says that DCA sequences generated by the algorithm converge to a locally unique solution of the problem if the initial points are taken from a suitably-chosen neighborhood of it. To deal with the implicitly defined iterative sequences, a local error bound for affine variational inequalities and novel techniques are used. Numerical results together with an analysis of the influence of the decomposition parameter, as well as a comparison between Algorithm A and Algorithm B will be given. Our results complement a recent and important paper of Le Thi, Huynh, and Pham Dinh [53].

A series of basic qualitative properties of the MSSC problem will be established herein. We will 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 [71] and of Bagirov [7]. 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.

Thus, briefly speaking, we will prove the convergence and the $R-$linear convergence rate of DCA applied to IQPs, establish a series of basic qualitative properties of the MSSC problem, suggest several modifications for the incremental algorithms in [7, 71], and study the finite convergence, the convergence, and the $Q-$linear convergence rate of the algorithms.

0.5 Scientific and Practical Meanings of the Results

• Solve the open question from [57, p. 488] on IQPs.

• Clarify the influence of the decomposition parameter for Algorithm A and Algorithm B to solve IQPs.

• Clarify the solution existence, structures of the local solution set and the

global solution set of the MSSC problem, as well as the problem's stability under data perturbations.

• Present for the first time finite convergence, convergence, and the $Q-$linear convergence rate of solution methods for the MSSC problem.

• Deepen one's knowledge on DC algorithms for solving IQPs, as well as properties of and solution algorithms for the MSSC problem.

0.6 Tools of Research

• Convex analysis;

• Set-valued analysis;

• Optimization theory.

0.7 The Structure of Dissertation

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. Here we prove convergence and convergence rate of DCA sequences generated by the Proximal DC decomposition algorithm. We also show that if the initial points are taken from a suitably-chosen neighborhood of it, DCA sequences generated by the algorithm converge to a locally unique solution of the IQP problem. In addition, we analyze the influence of the decomposition parameter on the speed of computation of the Proximal DC decomposition algorithm and the Projection DC decomposition algorithm, as well as a comparison between two these algorithms.

In Chapter 3, several basic qualitative properties of the MSSC problem are established. Among other things, we clarify the solution existence, properties of the global solutions, characteristic properties of the local solutions, locally Lipschitz property of the optimal value function, locally upper Lipschitz property of the global solution map, and the Aubin property of the local solution map.

Chapter 4 analyzes and develops some solution methods for the MSSC problem. We suggest some improvements of the incremental algorithms of

Ordin and Bagirov, and of Bagirov based on the DCA in DC programming and qualitative properties of the MSSC problem. In addition, we obtain several properties of the new algorithms and preliminary numerical tests of those on real-world databases. Finite convergence, convergence, and convergence rate of solution methods for the MSSC problem are presented here for the first time.

The dissertation is written on the basis of the following four articles in the List of author's related papers (see p. 112): paper No. 1 (submitted), paper No. 2 published in *Optimization*, paper No. 3 and paper No. 4 published in *Journal of Nonlinear and Convex Analysis*.

The results of this dissertation were 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);

- Seminar of Department of Numerical Analysis and Scientific Computing, Institute of Mathematics, Vietnam Academy of Science and Technology (Hanoi, Vietnam);

- The Scientific Seminar of Department of Computer Science, Faculty of Information Technology, Le Quy Don University (February 21, 2020, Hanoi, Vietnam);

- The Expanded Scientific Seminar of Department of Computer Science, Faculty of Information Technology, Le Quy Don University (June 16, 2020, Hanoi, Vietnam).

Chapter 1

Background Materials

In this chapter, we will review some background materials on Difference-of-Convex Functions Algorithms (DCAs for brevity), which were developed by Pham Dinh Tao and Le Thi Hoai An. Besides, two kinds of linear convergence rate of vector sequences will be defined.

It is well known that DCAs have a key role in nonconvex programming and many areas of applications [55]. For more details, we refer to [77, 79] and the references therein.

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

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

Note that $g^* : Y \to \bar{\mathbb{R}}$ is also a lower semicontinuous, proper, convex function [38, Propostion 3, p. 174]. From the definition it follows that

$$g(x) + g^*(y) \geq \langle x, y\rangle \quad (\forall x \in X,\ \forall y \in Y).$$

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

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

Since $g \in \Gamma_0(X)$, one has $g^{**}(x) = g(x)$ for all $x \in X$ by the Fenchel-Moreau theorem ([38, Theorem 1, p. 175]). This fact is the basis for various duality theorems in convex programming and DC programming.

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

$$\partial\varphi(u) := \{x^* \in \mathbb{R}^n \mid \langle x^*, x - u\rangle \leq \varphi(x) - \varphi(u) \ \ \forall x \in \mathbb{R}^n\}. \tag{1.1}$$

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

Clearly, the subdifferential $\partial\varphi(u)$ in (1.1) is a closed, convex set. The Fermat Rule for convex optimization problems asserts that $\bar{x} \in \mathbb{R}^n$ is a solution of the minimization problem

$$\min\{\varphi(x) \mid x \in \mathbb{R}^n\}$$

if and only if $0 \in \partial\varphi(\bar{x})$.

We now recall some useful properties of the Fenchel conjugate functions. The proofs of the next two propositions can be found in [77].

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 [79]) *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).

The set of the solutions (resp., the local solutions) of (P) is denoted by sol(P) (resp., by loc(P)).

Proposition 1.4 (First-order optimality condition; see [77]) *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).

The forthcoming example, which is similar to Example 1.1 in [93], shows that a stationary point needs not to be a local solution.

Example 1.1 Consider the DC program (P) with $f(x) = g(x) - h(x)$, where $g(x) = |x - 1|$ and $h(x) = (x - 1)^2$ for all $x \in \mathbb{R}$. For $\bar{x} := \dfrac{1}{2}$, one has $\partial g(\bar{x}) = \partial h(\bar{x}) = \{-1\}$. Since $\partial h(\bar{x}) \subset \partial g(\bar{x})$, $\bar{x}$ is a stationary point of (P). But $\bar{x}$ is not a local solution of (P), because $f(x) = x - x^2$ for all $x \leq 1$.

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

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

If $\partial h(\bar{x}) \neq \emptyset$ and $\bar{x}$ is a stationary point of (P), then $\bar{x}$ is a critical point of (P). The reverse implication does not hold in general. The following example is similar to Example 1.2 in [93].

Example 1.2 Consider the DC program (P) with $f(x) = g(x) - h(x)$ with $g(x) = (x - \frac{1}{2})^2$ and $h(x) = |x - 1|$ for all $x \in \mathbb{R}$. For $\bar{x} := 1$, we have $\partial g(\bar{x}) = \{1\}$ and $\partial h(\bar{x}) = [-1, 1]$. Hence $\partial g(\bar{x}) \cap \partial h(\bar{x}) \neq \emptyset$. So $\bar{x}$ is a critical point of (P). But, $\bar{x}$ is not a stationary point of (P), because $\partial h(\bar{x})$ is not a subset of $\partial g(\bar{x})$.

Consider problem (P). If the set $\partial h(\bar{x})$ is a singleton, then h is Gâteaux differentiable at $\bar{x}$ and $\partial h(\bar{x}) = \{\nabla_G h(\bar{x})\}$, where $\nabla_G h(\bar{x})$ denotes the Gâteaux derivative of h at $\bar{x}$. The converse is also true, i.e., if h is Gâteaux differentiable at $\bar{x}$, then $\partial h(\bar{x})$ is a singleton and $\partial h(\bar{x}) = \{\nabla_G h(\bar{x})\}$. In that case, the relation $\partial g(\bar{x}) \cap \partial h(\bar{x}) \neq \emptyset$ is equivalent to the inclusion $\partial h(\bar{x}) \subset \partial g(\bar{x})$. So, *if h is Gâteaux differentiable at $\bar{x}$, then $\bar{x}$ is a critical point if and only if it is a stationary point.*

1.2 DCA Schemes

The main idea of the theory of DCAs in [77] 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 the 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)).

Following Tuan [93], we can formulate and analyze the general DC algorithm of [77] as follows.

Scheme 1.1

Input: $f(x) = g(x) - h(x)$.
Output: $\{x^k\}$ *and* $\{y^k\}$.
Step 1. Choose $x^0 \in \operatorname{dom} g$. *Set* $k = 0$.
Step 2. Calculate

$$y^k \in \partial h(x^k); \tag{1.2}$$

$$x^{k+1} \in \partial g^*(y^k). \tag{1.3}$$

Step 3. Set $k = k + 1$ *and return to Step 2.*

For each $k \geq 0$, we have constructed a pair (x^k, y^k) satisfying (1.2) and (1.3).

Thanks to Proposition 1.2, we can transform the inclusion (1.2) equivalently as

$$\begin{aligned}
&y^k \in \partial h(x^k)\\
&\Leftrightarrow x^k \in \partial h^*(y^k)\\
&\Leftrightarrow h^*(y) - h^*(y^k) \geq \langle x^k, y - y^k\rangle \quad \forall\, y \in Y\\
&\Leftrightarrow h^*(y) - \langle x^k, y\rangle \geq h^*(y^k) - \langle x^k, y^k\rangle \quad \forall\, y \in Y.
\end{aligned}$$

Consequently, the condition (1.2) is equivalent to the requirement that y^k is a solution of the problem

$$\min\{h^*(y) - [g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle] \mid y \in Y\}, \tag{D_k}$$

where $y^{k-1} \in \operatorname{dom} g^*$ is the vector defined at the previous step $k-1$.

The inclusion $x^k \in \partial g^*(y^{k-1})$ means that

$$g^*(y) - g^*(y^{k-1}) \geq \langle x^k, y - y^{k-1}\rangle \quad \forall\, y \in Y.$$

Hence

$$g^*(y) \geq g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle \quad \forall\, y \in Y.$$

Thus, the affine function $g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle$ is a lower approximation of $g^*(y)$. If at step k we replace the term $g^*(y)$ in the object function of (D) by that lower approximation, we get the auxiliary problem (D_k).

Since (D_k) is a convex program, solving (D_k) is much easier than solving the DC program (D). Recall that y^k is a solution of (D_k).

Similarly, at each step $k+1$, the DC program (P) is replaced by the problem

$$\min\Big\{g(x) - [h(x^k) + \langle x - x^k, y^k\rangle] \mid x \in X\Big\}, \tag{P_k}$$

where $x^k \in \operatorname{dom} h^*$ has been defined at step k.

Since (P_k) is a convex program, solving (P_k) is much easier than solving the original DC program (P). As x^{k+1} satisfies (1.3), it is a solution of (P_k).

The objective function of (D_k) is a convex upper approximation of the objective function of (D). Moreover, the values of these functions at y^{k-1} coincide. Deleting some real constants from the expression of the objective function of (D_k), we get the following equivalent problem

$$\min\{h^*(y) - \langle x^k, y\rangle \mid y \in Y\}. \tag{1.4}$$

The objective function of (P_k) is a convex upper approximation of the objective function of (P). Moreover, the values of these functions at x^k coincide. Deleting some real constants from the expression of the objective function of (P_k), we get the following equivalent problem

$$\min\{g(x) - \langle x, y^k\rangle \mid x \in X\}. \tag{1.5}$$

If x^k is a critical point of (P), i.e., $\partial g(x^k) \cap \partial h(x^k) \neq \emptyset$, then DCA may produce a sequence $\{(x^\ell, y^\ell)\}$ with

$$(x^\ell, y^\ell) = (x^k, y^k) \quad \forall \ell \geq k.$$

Indeed, since there exists a point $\bar{x} \in \partial g(x^k) \cap \partial h(x^k)$, to satisfy (1.2) we can choose $y^k = \bar{x}$. Next, by Proposition 1.2, the inclusion (1.3) is equivalent to $y^k \in \partial g(x^{k+1})$. So, if we choose $x^{k+1} = x^k$ then (1.3) is fulfilled, because $y^k = \bar{x} \in \partial g(x^k)$.

In other words, DCA leads us to critical points, but it does not provide any tool for us to escape these critical points. Having a critical point, which is not a local minimizer, we need to use some advanced techniques from variational analysis to find a descent direction.

The following observations can be found in Tuan [93]:

• The DCA is a decomposition procedure which decomposes the solution of the pair of optimization problems (P) and (D) into the parallel solution of the sequence of convex minimization problems (P_k) and (D_k), $k \in \mathbb{N}$;

• The DCA does not include any specific technique for solving the convex problems (P_k) and (D_k). Such techniques should be imported from convex programming;

• The performance of DCA depends greatly on a concrete decomposition of the objective function into DC components;

• Although the DCA is classified as a deterministic optimization, each choice of the initial point x^0 may yield a variety of DCA sequences $\{x^k\}$ and $\{y^k\}$, because of the heuristic selection of $y^k \in \text{sol}(\text{D}_k)$ and $x^k \in \text{sol}(\text{P}_k)$ at every step k, if (D_k) (resp., (P_k)) has more than one solution.

The above analysis allows us to formulate a simplified version of DCA, which includes a termination procedure, as follows.

Scheme 1.2

Input: *$f(x) = g(x) - h(x)$.*
Output: *Finite or infinite sequences $\{x^k\}$ and $\{y^k\}$.*
Step 1. Choose $x^0 \in \text{dom}\, g$. Take $\varepsilon > 0$. Put $k = 0$.
Step 2.
Calculate y^k by solving the convex program (1.4).
Calculate x^{k+1} by solving the convex program (1.5).
Step 3. If $\|x^{k+1} - x^k\| \leq \varepsilon$ then stop, else go to Step 4.
Step 4. Set $k = k + 1$ and return to Step 2.

To understand the performance of the above DCA schemes, let us consider the following example.

Example 1.3 Consider the function $f(x) = g(x) - h(x)$ with $g(x) = (x-1)^2$ and $h(x) = |x - 1|$ for all $x \in \mathbb{R}$. Here $Y = X = \mathbb{R}$ and we have

$$g^*(y) = \sup\{xy - g(x) \mid x \in \mathbb{R}\} = \sup\{xy - (x-1)^2 \mid x \in \mathbb{R}\} = \frac{1}{4}y^2 + y.$$

Hence, $\partial g^*(y) = \{\frac{1}{2}y + 1\}$ for every $y \in Y$. Clearly, $\partial h(x) = \{-1\}$ for $x < 1$, $\partial h(x) = \{1\}$ for $x > 1$, and $\partial h(x) = [-1, 1]$ for $x = 1$. Using DCA Scheme 1.1, we will construct two DCA sequences $\{x^k\}$ and $\{y^k\}$ satisfying the conditions $y^k \in \partial h(x^k)$ and $x^{k+1} \in \partial g^*(y^k)$ for $k \in \mathbb{N}$. First, take any $x^0 > 1$. From the condition $y^0 \in \partial h(x^0) = \{1\}$, we get $y^0 = 1$. As $x^1 \in \partial g^*(y^0) = \{\frac{3}{2}\}$, one has $x^1 = \frac{3}{2}$. Thus, the condition $y^1 \in \partial h(x^1)$ implies that $y^1 = 1$. It is easy to show that $x^k = \frac{3}{2}$ and $y^k = 1$ for all $k \geq 2$. Therefore, the DCA sequences $\{x^k\}$ and $\{y^k\}$ converge respectively to $\bar{x} = \frac{3}{2}$ and $\bar{y} = 1$. Similarly,

starting from any $x^0 < 1$, one obtains the DCA sequences $\{x^k\}$ and $\{y^k\}$ with $x^k = \frac{1}{2}$ and $y^k = -1$ for all $k \geq 1$. These DCA sequences $\{x^k\}$ and $\{y^k\}$ converge respectively to $\bar{x} = \frac{1}{2}$ and $\bar{y} = -1$. Since

$$f(x) = \begin{cases} x^2 - x & \text{for } x \leq 1 \\ x^2 - 3x + 2 & \text{for } x \geq 1, \end{cases}$$

one finds that $\bar{x} = \frac{3}{2}$ and $\hat{x} = \frac{1}{2}$ are global minimizers of (P), and $\widetilde{x} := 1$ is the unique critical point of the problem.

With the initial point $x^0 = \widetilde{x} = 1$, since $y^0 \in \partial h(x^0) = [-1, 1]$, we can choose $y^0 = 0$. So, $x^1 \in \partial g^*(y^0) = \partial g^*(0) = \{1\}$. Hence $x^1 = 1$. Since $y^1 \in \partial h(x^1) = [-1, 1]$, we can choose $y^1 = 0$. Continuing the calculation, we obtain DCA sequences $\{x^k\}$ and $\{y^k\}$, which converge respectively to $\widetilde{x} = 1$ and $\bar{y} = 0$. Note that the limit point $\widetilde{x}$ of the sequence $\{x^k\}$ is the unique critical point of (P), which is neither a local minimizer nor a stationary point of (P).

To ease the presentation of some related programs, we consider the following scheme.

Scheme 1.3

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 using (1.2) and find

$$x^{k+1} \in \operatorname{argmin}\{g(x) - \langle x, y^k \rangle \mid x \in X\}. \tag{1.6}$$

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

1.3 General Convergence Theorem

We will recall the fundamental theorem on DCAs of Pham Dinh Tao and Le Thi Hoai An [77, Theorem 3], which is a firm theoretical basis for intensive uses of these algorithms in practice. Before doing so, we have to recall the

concepts of ρ-convex functions, modulus of convexity of convex functions, and strongly convex functions.

Definition 1.7 Let $\rho \geq 0$ and C be a convex set in the space X. A function $\theta : C \to \mathbb{R} \cup \{+\infty\}$ is called *ρ-convex* if

$$\theta\big(\lambda x + (1-\lambda)x'\big) \leq \lambda\theta(x) + (1-\lambda)\theta(x') - \frac{\lambda(1-\lambda)}{2}\rho \,||\, x - x' \,||^2$$

for all numbers $\lambda \in (0,1)$ and vectors $x, x' \in C$. This amounts to saying that the function $\theta(\cdot) - (\rho/2)\|\cdot\|^2$ is convex on C.

Definition 1.8 The *modulus of convexity* of θ on C is given by

$$\rho(\theta, C) = \sup\Big\{\rho \geq 0 \mid \theta - (\rho/2)\|\cdot\|^2 \text{ is convex on } C\Big\}.$$

If $C = X$ then we write $\rho(\theta)$ instead of $\rho(\theta, C)$. Function θ is called *strongly convex* on C if $\rho(\theta, C) > 0$.

Consider the problem (P). If $\rho(g) > 0$ (resp., $\rho(g^*) > 0$), let ρ_1 (resp., ρ_1^*) be a real number such that $0 \leq \rho_1 < \rho(g)$ (resp., $0 \leq \rho_1^* < \rho(g^*)$). If $\rho(g) = 0$ (resp., $\rho(g^*) = 0$), let $\rho_1 = 0$ (resp., $\rho_1^* = 0$). If $\rho(h) > 0$ (resp., $\rho(h^*) > 0$), let ρ_2 (resp., ρ_2^*) be a real number such that $0 \leq \rho_2 < \rho(h)$ (resp., $0 \leq \rho_2^* < \rho(h^*)$). If $\rho(h) = 0$ (resp., $\rho(h^*) = 0$), let $\rho_2 = 0$ (resp., $\rho_2^* = 0$).

The convenient abbreviations $dx^k := x^{k+1} - x^k$ and $dy^k := y^{k+1} - y^k$ were adopted in [77].

Theorem 1.1 ([77, Theorem 3]) *Let $\alpha := \inf\{f(x) = g(x) - h(x) \mid x \in \mathbb{R}^n\}$. Assume that the iteration sequences $\{x^k\}$ and $\{y^k\}$ are generated by DCA Scheme 1. Then, the following properties are valid:*

(i) *The inequalities*

$$\begin{aligned}(g-h)(x^{k+1}) \leq\ & (h^*-g^*)(y^k) - \max\Big\{\tfrac{\rho_2}{2}\|dx^k\|^2, \tfrac{\rho_2^*}{2}\|dy^k\|^2\Big\}\\ \leq\ & (g-h)(x^k) - \max\Big\{\tfrac{\rho_1+\rho_2}{2}\|dx^k\|^2, \tfrac{\rho_1^*}{2}\|dy^{k-1}\|^2\\ & + \tfrac{\rho_2}{2}\|dx^k\|^2, \tfrac{\rho_1^*}{2}\|dy^{k-1}\|^2 + \tfrac{\rho_2^*}{2}\|dy^k\|^2\Big\}\end{aligned}$$

hold for every k;

(ii) *The inequalities*

$$\begin{aligned}(h^*-g^*)(y^{k+1}) \leq\ & (g-h)(x^{k+1}) - \max\Big\{\tfrac{\rho_1}{2}\|dx^{k+1}\|^2, \tfrac{\rho_1^*}{2}\|dy^k\|^2\Big\}\\ \leq\ & (h^*-g^*)(y^k) - \max\Big\{\tfrac{\rho_1^*+\rho_2^*}{2}\|dy^k\|^2, \tfrac{\rho_1}{2}\times\\ & \|dx^{k+1}\|^2 + \tfrac{\rho_2}{2}\|dx^k\|^2, \tfrac{\rho_1^*}{2}\|dy^k\|^2 + \tfrac{\rho_2}{2}\|dx^k\|^2\Big\}\end{aligned}$$

hold for every k;

(iii) *If α is finite, then $\{(g-h)(x^k)\}$ and $\{(h^*-g^*)(y^k)\}$ are decreasing sequences that converge to the same limit $\beta \geq \alpha$. Furthermore,*

(a) *If $\rho(g)+\rho(h)>0$ (resp., $\rho(g^*)+\rho(h^*)>0$), then* $\lim_{k\to\infty}(x^{k+1}-x^k)=0$ *(resp.,* $\lim_{k\to\infty}(y^{k+1}-y^k)=0$*);*

(b) $\lim_{k\to\infty}[g(x^k)+g^*(y^k)-\langle x^k,y^k\rangle]=0$*;*

(c) $\lim_{k\to\infty}[h(x^{k+1})+h^*(y^k)-\langle x^{k+1},y^k\rangle]=0$.

(iv) *If α is finite, and $\{x^k\}$ and $\{y^k\}$ are bounded, then for every cluster point $\bar{x}$ of $\{x^k\}$ (resp., $\bar{y}$ of $\{y^k\}$), there is a cluster point $\bar{y}$ of $\{y^k\}$ (resp., $\bar{x}$ of $\{x^k\}$) such that:*

(d) $(\bar{x},\bar{y})\in[\partial g^*(\bar{y})\cap\partial h^*(\bar{y})]\times[\partial g(\bar{x})\cap\partial h(\bar{x})]$*;*

(e) $(g-h)(\bar{x})=(h^*-g^*)(\bar{y})=\beta$*;*

(f) $\lim_{k\to\infty}\{g(x^k)+g^*(y^k)\}=\lim_{k\to\infty}\langle x^k,y^k\rangle$.

The estimates in the assertions (i) and (ii) of the above theorem can be slightly improved as shown in the next remark.

Remark 1.1 If $\rho(h)>0$, then ρ_2 is a real number such that $\rho_2\in[0,\rho(h))$. Since the construction of the sequences $\{x^k\}$ and $\{y^k\}$ does not depend on the choice of the constants ρ_1, ρ_1^*, ρ_2, and ρ_2^*, by assertion (i) of Theorem 1.1 we have for each $k\in\mathbb{N}$ the inequality

$$(g-h)(x^{k+1})\leq(h^*-g^*)(y^k)-\max\Big\{\frac{\rho_2}{2}\|dx^k\|^2,\frac{\rho_2^*}{2}\|dy^k\|^2\Big\}.$$

Passing the last inequality to the limit as $\rho_2\to\rho(h)$, we get

$$(g-h)(x^{k+1})\leq(h^*-g^*)(y^k)-\max\Big\{\frac{\rho(h)}{2}\|dx^k\|^2,\frac{\rho_2^*}{2}\|dy^k\|^2\Big\}.$$

Using this trick simultaneously for the constants related to strongly convex functions among the family $\{g,h,g^*,h^*\}$, we can show that the following improved versions of the estimates in the assertions (i) and (ii) of Theorem 1.1 are valid:

$$\begin{aligned}(g-h)(x^{k+1})\leq\ & (h^*-g^*)(y^k)-\max\Big\{\tfrac{\rho(h)}{2}\|dx^k\|^2,\tfrac{\rho(h^*)}{2}\|dy^k\|^2\Big\}\\ \leq\ & (g-h)(x^k)-\max\Big\{\tfrac{\rho(g)+\rho(h)}{2}\|dx^k\|^2,\tfrac{\rho(g^*)}{2}\|dy^{k-1}\|^2\\ & +\tfrac{\rho(h)}{2}\|dx^k\|^2,\tfrac{\rho(g^*)}{2}\|dy^{k-1}\|^2+\tfrac{\rho(h^*)}{2}\|dy^k\|^2\Big\},\end{aligned}$$

$$
\begin{aligned}
(h^* - g^*)(y^{k+1}) \leq\ & (g-h)(x^{k+1}) - \max\left\{\tfrac{\rho(g)}{2}\|dx^{k+1}\|^2, \tfrac{\rho(g^*)}{2}\|dy^k\|^2\right\} \\
\leq\ & (h^* - g^*)(y^k) - \max\left\{\tfrac{\rho(g^*)+\rho(h^*)}{2}\|dy^k\|^2, \tfrac{\rho(g)}{2}\times\right. \\
& \left.\|dx^{k+1}\|^2 + \tfrac{\rho(h)}{2}\|dx^k\|^2, \tfrac{\rho(g^*)}{2}\|dy^k\|^2 + \tfrac{\rho(h)}{2}\|dx^k\|^2\right\}.
\end{aligned}
$$

The forthcoming example is designed as an illustration for Theorem 1.1.

Example 1.4 Consider the function $f(x) = g(x) - h(x)$ in Example 1.1, where $g(x) = |x-1|$ and $h(x) = (x-1)^2$ for all $x \in \mathbb{R}$. Here $Y = X = \mathbb{R}$ and we have

$$
h^*(y) = \sup\{xy - h(x) \mid x \in \mathbb{R}\} = \sup\{xy - (x-1)^2 \mid x \in \mathbb{R}\} = \frac{1}{4}y^2 + y.
$$

Using DCA Scheme 1.2, we calculate DCA sequences $\{x^k\}$ and $\{y^k\}$ by solving, respectively, the convex programs (1.4) and (1.5) for $k \in \mathbb{N}$. Choose $\varepsilon = 0$. First, select $x^0 = \frac{2}{3}$. Since y^0 is a solution of (1.4) for $k = 0$, we get $y^0 = -\frac{2}{3}$. As x^1 is a solution of (1.5) for $k = 0$, one has $x^1 = 1$. Continuing the calculation, we obtain $y^k = 0$ for $k \geq 1$ and $x^k = 1$ for $k \geq 2$. The condition in Step 3 of DCA Scheme 1.2 is satisfied at $k = 1$, so the algorithm stops after one step and yields the point $\bar{x} = x^2$, which is the unique local solution of (P). It is not difficult to show that one has the same result for any initial point $x^0 \in \left(\frac{1}{2}, \frac{3}{2}\right)$. If $x^0 \in \left\{\frac{1}{2}, \frac{3}{2}\right\}$, then the algorithm stops at $k = 0$ and one gets the point $\bar{x} = x^1 = x^0$. Note that this $\bar{x}$ is a stationary point of (P), which is not a local solution. If $x^0 < \frac{1}{2}$ or $x^0 > \frac{3}{2}$, then $f(x_k) \to -\infty$ as $k \to \infty$. So, $\{x^k\}$ does not have any cluster point.

1.4 Convergence Rates

In Chapter 2 and Chapter 4, we will prove several results on convergence rates of iterative sequences. The following two types of linear convergence will be discussed in the sequel: *Q-linear convergence* and *R-linear convergence*. Let us recall these notions.

Definition 1.9 (See, e.g., [70, p. 28] and [88, pp. 293–294]) One says that a sequence $\{x^k\} \subset \mathbb{R}^n$ *converges Q-linearly* to a vector $\bar{x} \in \mathbb{R}^n$ if there exits $\beta \in (0,1)$ such that $\|x^{k+1} - \bar{x}\| \leq \beta\|x^k - \bar{x}\|$ for all k sufficiently large.

Clearly, if $x^k \neq \bar{x}$, then the relation $\|x^{k+1} - \bar{x}\| \leq \beta\|x^k - \bar{x}\|$ in Definition 1.9

can be rewritten equivalently as $\dfrac{\|x^{k+1}-\bar{x}\|}{\|x^k-\bar{x}\|} \leq \beta$. The word "$Q$", which stands for "quotient", comes from this context.

Definition 1.10 (See, e.g., [70, p. 30]) One says that a sequence $\{x^k\} \subset \mathbb{R}^n$ *converges R-linearly* to a vector $\bar{x} \in \mathbb{R}^n$ if there is a sequence of nonnegative scalars $\{\mu_k\}$ such that $\|x^k - \bar{x}\| \leq \mu_k$ for all k sufficiently large, and $\{\mu_k\}$ converges Q-linearly to 0.

If a sequence $\{x^k\}$ converges Q-linearly to a vector $\bar{x}$, then it converges R-linearly to $\bar{x}$. To see this, it suffices to select a constant $\beta \in (0,1)$ satisfying the condition stated in Definition 1.9, put $\mu_k = \beta\|x^{k-1} - \bar{x}\|$ for all $k \geq 1$, and note that $\|x^k - \bar{x}\| \leq \mu_k$ for all k sufficiently large, while $\{\mu_k\}$ converges Q-linearly to 0 because $\mu_{k+1} \leq \mu_k$ for all k sufficiently large. It well known that the R-linear convergence may not imply the Q-linear convergence. As an example, one may follow [70, p. 30] to consider the sequence of positive scalars

$$x^k = \begin{cases} 1 + (0.5)^k, & k \text{ is even}, \\ 1, & k \text{ is odd}, \end{cases}$$

and observe that $\{x^k\}$ converges R-linearly to 1, while the sequence does not converge Q-linearly to 1.

Sometimes, one says that a sequence $\{x^k\} \subset \mathbb{R}^n$ converges R-linearly to a vector $\bar{x} \in \mathbb{R}^n$ whenever

$$\limsup_{k\to\infty} \|x^k - \bar{x}\|^{1/k} < 1 \tag{1.7}$$

(see, e.g., [92]). The word "R", which stands for "root", comes from this context.

The next proposition clarifies *the equivalence* between the definition of $Q-$linear convergence in (1.7) and the one given in Definition 1.9.

Proposition 1.5 *A sequence $\{x^k\} \subset \mathbb{R}^n$ converges R-linearly to a vector $\bar{x} \in \mathbb{R}^n$ if and only if the strict inequality* (1.7) *holds.*

Proof. First, to prove the necessity, suppose that $\{x^k\}$ converges Q-linearly to a vector $\bar{x}$. Then, there is a sequence of nonnegative scalars $\{\mu_k\}$ such that $\|x^k - \bar{x}\| \leq \mu_k$ for all k sufficiently large, and $\{\mu_k\}$ converges Q-linearly to 0. Therefore, we can find a constant $\beta \in (0,1)$ and a number $k_1 \in \mathbb{N}$ such

that $\mu_{k+1} \leq \beta\mu_k$ for all $k \geq k_1$. Without loss of generality, we may assume that $\mu_{k_1} > 0$. For any $k > k_1$, one has

$$\|x^k - \bar{x}\| \leq \mu_k \leq \beta\mu_{k-1} \leq \cdots \leq \beta^{k-k_1}\mu_{k_1}.$$

It follows that $\|x^k - \bar{x}\| \leq \dfrac{\mu_{k_1}}{\beta^{k_1}}\beta^k$ for all $k > k_1$. Therefore,

$$\limsup_{k\to\infty} \|x^k - \bar{x}\|^{1/k} \leq \beta \limsup_{k\to\infty} \left[\left(\frac{\mu_{k_1}}{\beta^{k_1}}\right)^{1/k}\right] = \beta < 1.$$

Thus, the inequality (1.7) holds.

Now, to prove the sufficiency, suppose (1.7) is valid. Then, there exist a constant $\gamma \in (0, 1)$ and a natural number $k_2 \in \mathbb{N}$ such that $\|x^k - \bar{x}\|^{1/k} \leq \gamma$ for all $k \geq k_2$. Hence, $\|x^k - \bar{x}\| \leq \gamma^k$ for all $k \geq k_2$. Setting $\mu_k = \gamma^k$ for $k \in \mathbb{N}$, we have $\{\mu_k\}$ such that $\|x^k - \bar{x}\| \leq \mu_k$ for all $k \geq k_2$. In addition, the fulfillment of the equality $\mu_{k+1} = \gamma\mu_k$ for all $k \geq k_2$ together with the property $\lim\limits_{k\to\infty} \mu_k = 0$ shows that $\{\mu_k\}$ converges Q-linearly to 0. Hence, the sequence $\{x^k\}$ converges R-linearly to $\bar{x}$.

The proof is complete. □

1.5 Conclusions

In this chapter, we have recalled basic facts concerning the DCA theory from [55, 77, 93] and analyzed some fundamental properties of DC programming and DCA by presenting various remarks and examples. In addition, two types of linear convergence of vector sequences have been defined and compared.

The facts formulated in Remark 1.1 are new. They will be useful for our investigations in the next chapter.

Example 1.4 has shown that the performance of DCA depends greatly on the chosen d.c. decomposition of the objective function and the selection of the initial point.

Chapter 2

Analysis of an Algorithm in Indefinite Quadratic Programming

In this chapter, we will study two algorithms for solving the indefinite quadratic programming problem: the *Projection DC decomposition algorithm* (Algorithm A) and the *Proximal DC decomposition algorithm* (Algorithm B).

Our first aim is to prove that *any DCA sequence generated by Algorithm B converges R-linearly to a KKT point*. Hence, combining this with Theorem 2.1 from [92], we have a complete solution for the Conjecture in [58, p. 489]. Our result is obtained by applying some arguments of [92] and a new technique in dealing with implicitly defined DCA sequences.

By [58, Theorem 3], we know that DCA sequences generated by the Algorithm A converge to a locally unique solution of the IQP if the initial points are taken from a suitably-chosen neighborhood of it. In the terminology of [59], this means that the locally unique solutions of the IQP are *asymptotically stable* with respect to Algorithm A. The open question of [58, p. 488] can be reformulated as follows: *Is it true that the locally unique solutions of the IQP are asymptotically stable with respect to Algorithm B?* The second aim of the present chapter is to use a novel technique to establish the asymptotic stability of the locally unique solutions with respect to Algorithm B under a mild additional assumption on the DCA decomposition parameter. It is still unclear to us whether that assumption can be dropped, or not.

Our third aim is to analyze the influence of the decomposition parameter on the rates of convergence of DCA sequences and compare the performances of the algorithms A and B upon randomly generated data sets.

This chapter is written on the basis of paper No. 1 in the List of author's related papers (see p. 112).

2.1 Indefinite Quadratic Programs and DCAs

The importance of the indefinite quadratic programming problem under linear constraints (IQP for brevity) in optimization theory and its various applications is well known. Roughly speaking, the sequential quadratic programming methods (Wilson's method, Pang's method, the local Maratos-Mayne-Polak method, global MMP method, the Maratos-Mayne-Polak-Pang method, etc.) [83, Section 2.9] reduce the given nonlinear mathematical programming program with smooth data to solving a sequence of IQPs. For other theoretical aspects of IQP, we refer to [16]. Gupta [30] gives a review on applications of IQP in finance, agriculture, economics, production operations, marketing, and public policy. Chapters 5 and 6 of the book by Cornuéjols *et al.* [21] are devoted to quadratic programming models in finance. Jen and Wang [41] shows that the image enhancement problem can be formulated as a quadratic programming problem. Both the methodological and functional applications of quadratic programming are reviewed by McCarl *et al.* [68]. Akoa [2] discusses the IQP in the context of training support vector machines of nonpositive–semidefinite kernels by using the Difference-of-Convex algorithms. Recently, similar questions in machine learning have been studied by Xu *et al.* [99] and Xue *et al.* [100] by other methods. Liu *et al.* [62, 63] have studied the IQP associated with support vector machine with indefinite kernel - a model that has attracted increasing attentions in machine learning. For applications of quadratic programming under quadratic constraints, we refer to the paper by Wiebking [96].

Numerical methods for solving IQP have been addressed in many research works; see, e.g., [17, 19, 78–80, 82, 101–103]. Note that most of the known algorithms yield just stationary points (that is, the Karush-Kuhn-Tucker points, or KKT points for short), or local minimizers. In other words, most of the known algorithms are *local solution methods*. Since the IQP is NP-hard (see [72] and also [17]), finding its *global solutions* remains a challenging question.

We are interested in studying and implementing two methods to solve the

IQP, that are based on a general scheme for solving DC (Difference-of-Convex-functions) programs due to Pham Dinh and Le Thi [77, 79] (see also [54, 81]). A combination of DCA (DC Algorithms) with interior point techniques for solving large-scale nonconvex quadratic programming has been proposed in [82]. The two DC decompositions suggested in [82] are the projection DC decomposition and the proximal DC decomposition. They lead to two algorithms for solving the IQP: the *Projection DC decomposition algorithm* (Algorithm A) and the *Proximal DC decomposition algorithm* (Algorithm B); see [57, 82], and the detailed descriptions given below. It is worthy to stress the following features of these algorithms:

- The algorithm descriptions are simple;
- The implementation is easy;
- No line searches are required.

Nevertheless, using the DCA theory one can only assert [58, Theorem 1] that any cluster point of a DCA sequence generated by the above-mentioned algorithms is a KKT point of the IQP. To be sure that such cluster points do exist, one must establish the boundedness of the DCA sequence. In general, DCA sequences need not be bounded [58, Example 1]. But there is a Conjecture [58, p. 489] saying that *if the IQP has global solutions, then every DCA sequence generated by one of the algorithms A and B must be bounded.* Recently, the Conjecture has been solved in the affirmative for the two-dimensional IQP by Tuan [91]. To solve it in the general case, Tuan [92] has used a local error bound for affine variational inequalities and several specific properties of the KKT point set of the IQP which were obtained by Luo and Tseng [65] (see also Tseng [90] and Luo [64]). The main result of [92] is the following theorem: *If the IQP has a nonempty solution set, then every DCA sequence generated by Algorithm A converges R-linearly to a KKT point.*

Numerous numerical tests, which will be reported in Section 2.4, lead us to the following observations:

- For both the the algorithms A and B, the closer is the positive decomposition parameter to the lower bound of the admissible parameter interval, the higher is the convergence rate of DCA sequences;
- Applied to the same problem with the same initial point, Algorithm B

is more efficient than Algorithm A in terms of the number of computation steps and the execution time.

Our results complement a recent paper of Le Thi, Huynh, and Pham Dinh [53], where by original proofs the authors have obtained a series of important convergence theorems for DCA algorithms, which solve optimization problems with subanalytic data. To be more precise, from Theorems 3.4, 3.5, and 4.2 of [53] it follows that any DCA sequence generated by Algorithm B converges R-linearly to a KKT point, *if the sequence is bounded*. Since the boundedness of DCA sequences cannot be obtained by the Lojasiewicz inequality (see [53, Theorem 2.1]) and the related results on Kurdyka-Lojasiewicz properties (see [4] and the references therein), Theorem 2.2 and its proof are new contributions to the analysis of the existing solution algorithms in indefinite quadratic programming.

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.

Now we describe some standard notations that will be used later on. The unit matrix in $\mathbb{R}^{n\times n}$ is denoted by E. The *eigenvalues* of a symmetric matrix $M \in \mathbb{R}^{n\times n}$ are ordered in the sequence $\lambda_1(M) \leq ... \leq \lambda_n(M)$ with counting multiplicities. For an index set $\alpha \subset \{1,\ldots,m\}$, by A_α we denote the matrix composed by the rows A_i, $i \in \alpha$, of A. Similarly, b_α is the vector composed by the components b_i, $i \in \alpha$, of b. The *pseudo-face* of C corresponding to α is the set

$$\big\{x \in \mathbb{R}^n \mid A_\alpha x = b_\alpha,\ A_{\bar\alpha}x > b_{\bar\alpha}\big\},$$

where $\bar\alpha := \{1,\ldots,m\}\backslash\alpha$. Let $B(x,\varepsilon)$ (resp., $\bar B(x,\varepsilon)$) denote the open (resp., closed) ball with center x and radius $\varepsilon > 0$. Given s vectors $v^1,\ldots,v^s$ in $\mathbb{R}^n$, we denote by $\mathrm{pos}\{v^1,\ldots,v^s\}$ the closed convex cone generated by $v^1,\ldots,v^s$,

that is

$$\text{pos}\{v^1,\ldots,v^s\} = \Big\{v = \sum_{i=1}^{s} \lambda_i v^i \mid \lambda_i \geq 0 \ \text{ for } \ i = 1,\ldots,s\Big\}.$$

The *metric projection* of $u \in \mathbb{R}^n$ onto C is denoted by $P_C(u)$, that is $P_C(u)$ belongs to C and

$$\big\|u - P_C(u)\big\| = \min_{x\in C} \|u - x\|.$$

The *tangent cone* to C at $x \in C$ is denoted by $T_C(x)$, i.e.,

$$T_C(x) = \{t(y-x) \mid t \geq 0,\ y \in C\} = \{v \in \mathbb{R}^n \mid A_\alpha v \geq 0\},$$

where $\alpha = \{i \mid A_i x = b_i\}$. The *normal cone* to C at $x \in C$ is denoted by $N_C(x)$, that is

$$\begin{aligned} N_C(x) = \big(T_C(x)\big)^* &= \{\xi \in \mathbb{R}^n \mid \langle \xi, v\rangle \leq 0 \ \ \forall v \in T_C(x)\} \\ &= -\text{pos}\{A_i \mid i \in \alpha\}. \end{aligned}$$

Following [82], 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

$$f(x) = \varphi(x) - \psi(x) \tag{2.2}$$

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*

$$\min\big\{g(x) - h(x) \mid x \in \mathbb{R}^n\big\}$$

with $g(x) := \varphi(x) + \delta_C(x)$, $h(x) := \psi(x)$, where $\delta_C(x) = 0$ for $x \in C$ and $\delta_C(x) = +\infty$ for $x \notin C$ is the indicator function of C. Let $x^0 \in \mathbb{R}^n$ be a given initial point. In accordance with the general solution method of [79,81] (see Schemes 1 and 2 in Chapter 1), at every step $k \geq 0$ one computes $y^k = \big(\nabla h(x^k)\big)^T = Q_2 x^k$ and finds the unique solution, denoted by x^{k+1} of the convex minimization problem

$$\min\Big\{g(x) - [h(x^k) + \langle x - x^k, y^k\rangle] \mid x \in \mathbb{R}^n\Big\}.$$

The latter is equivalent to the *strongly convex quadratic program*

$$\min\Big\{\frac{1}{2}x^T Q_1 x + q^T x - x^T Q_2 x^k \mid x \in C\Big\}. \tag{2.3}$$

The obtained sequence $\{x^k\}$ is called *the DCA sequence generated by the DC algorithm and the initial point* x^0.

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

$$\begin{cases} Qx + q - A^T\lambda = 0, \\ Ax \geq b, \quad \lambda \geq 0, \quad \lambda^T(Ax - b) = 0, \end{cases} \tag{2.4}$$

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

This definition can be rephrased (see, e.g., [50]) as follows: If $x \in C$ and

$$\langle \nabla f(x), v \rangle = (Qx + q)^T v \geq 0 \quad \forall v \in T_C(x), \tag{2.5}$$

then x is said to be a KKT point of (2.1). Since condition (2.5) is equivalent to $\langle \nabla f(x), y - x \rangle \geq 0$ for all $y \in C$, $x \in C$ is a KKT point of the IQP in (2.1) if and only if it is a solution of the *affine variational inequality*

$$x \in C, \quad \langle Qx + q, u - x \rangle \geq 0 \ \ \forall u \in C. \tag{2.6}$$

Denote the KKT point set (resp., the global solution set) of IQP by C^* (resp., $\mathcal{S}$). It is well known (see, e.g., [50]) that $\mathcal{S} \subset C^*$.

We now recall some basic properties of DCA sequences which follow from applying the fundamental theorem on DCAs (see Theorem 1.1) and Remark 1.1 to the IQP in (2.1). In doing so, we observe that the modulus of convexity of the function $g(x) = \varphi(x) + \delta_C(x)$ with $\varphi(x) = \frac{1}{2}x^T Q_1 x + q^T x$ is larger or equal to $\lambda_1(Q_1)$. Similarly, the modulus of convexity of the function $h(x) = \psi(x)$ with and $\psi(x) = \frac{1}{2}x^T Q_2 x$ is equal to $\lambda_1(Q_2)$.

Theorem 2.1 (See [81, Theorem 3] and [82, Theorem 2.1]) *Every DCA sequence $\{x^k\}$ generated by the above DC algorithm and an initial point $x^0 \in \mathbb{R}^n$ has the following properties:*

(i) $f(x^{k+1}) \leq f(x^k) - \dfrac{1}{2}[\lambda_1(Q_1) + \lambda_1(Q_2)]\|x^{k+1} - x^k\|^2$ *for every* $k \geq 1$;

(ii) $\{f(x^k)\}$ *converges to an upper bound* f_* *for the optimal value of* (2.1);

(iii) *Every cluster point* $\bar{x}$ *of* $\{x^k\}$ *is a KKT point of* (2.1);

(iv) *If* $\inf\limits_{x \in C} f(x) > -\infty$, *then* $\lim\limits_{k \to \infty} \|x^{k+1} - x^k\| = 0$.

Remark 2.1 By [81, Theorem 3], if $x^0 \in C$ then we have the inequality in (i) for every $k \geq 0$. To see this, it suffices to note that $x^0 \in C = \text{dom} g$, where $g = \varphi + \delta_C$ and dom $:= \{x \mid g(x) < +\infty\}$.

As one can easily compute the smallest eigenvalue $\lambda_1(Q)$ and the largest eigenvalue $\lambda_n(Q)$ of $Q = Q_1 - Q_2$ by some algorithm (for instance, by the Newton-Raphson algorithm in [88]) or software, the next realizations of the DC decomposition (2.2) can be done:

(a) $Q_1 := \rho E$, $Q_2 := \rho E - Q$, where ρ is a positive real value satisfying the condition $\rho \geq \lambda_n(Q)$;

(b) $Q_1 := Q + \rho E$, $Q_2 := \rho E$, where ρ is a positive real value satisfying the condition $\rho > -\lambda_1(Q)$.

The number ρ is called *the decomposition parameter*. The following algorithms appear on the basis of (a) and (b), respectively.

Algorithm A. (*Projection DC decomposition algorithm*) Fix a positive number $\rho \geq \lambda_n(Q)$ and choose an initial point $x^0 \in \mathbb{R}^n$. For every $k \geq 0$, compute the point

$$x^{k+1} := P_C\Big(x^k - \frac{1}{\rho}(Qx^k + q)\Big)$$

which is the unique solution of (2.3), where $Q_1 = \rho E$ and $Q_2 := \rho E - Q$. The latter can be rewritten in the form

$$\min\Big\{\Big\|x - \frac{1}{\rho}(y^k - q)\Big\|^2 \mid Ax \geq b\Big\} \tag{2.7}$$

with

$$y^k := (\rho E - Q)x^k. \tag{2.8}$$

The scheme of the algorithm with a stopping criterion is as follows. (To have an infinite DCA sequence, one has to choose $\varepsilon = 0$.)

Input: *$Q \in \mathbb{R}^{n\times n}$, $A \in \mathbb{R}^{m\times n}$, $q \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $\rho > 0$ and $\rho \geq \lambda_n(Q)$, and a tolerance $\varepsilon > 0$.*
Output: *$\{x^k\}$ and $\{y^k\}$.*
Step 1. Choose $x^0 \in \mathbb{R}^n$, and set $k := 0$.
Step 2. Calculate y^k by using (2.8).
Step 3. Calculate x^{k+1} by solving the convex program (2.7).
Step 4. If $\|x^{k+1} - x^k\| \leq \varepsilon$ then stop, else go to Step 5.
Step 5. Set $k = k + 1$ and go to Step 2.

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

$$\min \Big\{ \psi(x) := \frac{1}{2} x^T Q x + q^T x + \frac{\rho}{2} \|x - x^k\|^2 \mid Ax \geq b \Big\}. \tag{2.9}$$

(Note that, up to adding a real constant, the objective function of (2.9) 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 $Q_2 = \rho E$.) The scheme of the algorithm with a stopping criterion is as follows. (To have an infinite DCA sequence, one has to choose $\varepsilon = 0$.)

Input: $Q \in \mathbb{R}^{n \times n}$, $A \in \mathbb{R}^{m \times n}$, $q \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $\rho > 0$ *and* $\rho > -\lambda_1(Q)$, *and a tolerance* $\varepsilon > 0$.
Output: $\{x^k\}$.
Step 1. *Choose* $x^0 \in \mathbb{R}^n$ *and put* $k := 0$.
Step 2. *Calculate* x^{k+1} *by solving the convex program* (2.9).
Step 3. *If* $\|x^{k+1} - x^k\| \leq \varepsilon$ *then stop, else go to Step 4.*
Step 4. *Set* $k = k + 1$ *and go to Step 2.*

Let $\{x^k\}$ be a DCA sequence generated by one of the last two algorithms and an initial point x^0. If $\{x^k\}$ is bounded, then it has a convergent subsequence $x^{k_j} \to \bar{x}$. According to Theorem 2.1, $\bar{x}$ is a KKT point of IQP. Since one wants to find a global solution, one has to *restart* the algorithm if $\bar{x} \notin \mathcal{S}$. To do so, we must find some $u \in C$ such that $f(u) < f(\bar{x})$, put $x^0 = u$ and construct a new DCA sequence. If the latter is again bounded, one finds a new KKT point $\bar{u} \in C^*$ with $f(\bar{u}) \leq f(u) < f(\bar{x})$ (see Theorem 2.1). The process is continued until finding a point $\bar{x} \in \mathcal{S}$. Since the distinct values of f on C^* does not exceed 2^m (see [17, Lemma 4]), the upper bound for the number of restarts of any DC algorithm is 2^m.

Before proving the convergence and the $R-$linear convergence rate of Algorithm B, let us consider one example, which is designed to show how Algorithms A and B are performed in practice.

Example 2.1 (see [50, Example 11.5, p. 209]) Consider problem (2.1) with

$n = 2$, $m = 3$, $Q = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, $A = \begin{bmatrix} 1 & -2 \\ 1 & 2 \\ 1 & 0 \end{bmatrix}$, $q = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$, $b = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$. Here, we have $f(x) = \frac{1}{2}(x_1^2 - x_2^2) - x_1$ on the set

$$C = \big\{x \in \mathbb{R}^2 \mid x_1 - 2x_2 \geq 0,\ x_1 + 2x_2 \geq 0,\ x_1 \geq 2\big\}.$$

The eigenvalues of Q are $\lambda_1 = -1$ and $\lambda_2 = 1$. Denote by C^* the KKT point set of (P). A direct computation using (2.4) gives

$$C^* = \big\{(2,0),\ (2,1),\ (2,-1)\big\},\quad \mathcal{S} = \mathrm{loc(P)} = \big\{(2,1),\ (2,-1)\big\}.$$

To implement the algorithms A and B, we choose $\varepsilon = 10^{-6}$.

For Algorithm A, one can choose $\rho = 1$. The objective function of the problem (P) can be decomposed as follows $f(x) = g(x) - h(x)$, where

$$g(x) = \frac{1}{2}x^T(\rho E)x + q^T x = \frac{1}{2}(x_1^2 + x_2^2) - x_1,$$

and $h(x) = \frac{1}{2}x^T(\rho E - Q)x = x_2^2$. The implementation of Algorithm A begins with selecting an initial point, say, $x^0 = (2,2)$, and setting $k = 0$. Using (2.8), one obtains $y^0 = (0,4)$. By solving the convex program (2.7), one gets $x^1 = \big(\frac{12}{5}, \frac{6}{5}\big)$. Since $\|x^1 - x^0\| > \varepsilon$, one increases k by 1 and computes y^1. By (2.8), one has $y^1 = (0, \frac{12}{5})$. Using (2.7), one obtains $x^2 = (2,1)$. The stopping criterion in Step 4 is not satisfied, so one sets $k = 2$ and goes to Step 2. By the rule (2.8), one has $y^2 = (0,2)$. Using (2.7) again, one gets $x^3 = (2,1)$. Thus, the condition $\|x^3 - x^2\| \leq \varepsilon$ is satisfied. So, the computation stops after 3 steps and one has $\bar{x} = (2,1)$, which belongs to $\mathcal{S}$ (see Table 2.1 a)). Similarly, selecting the initial point $x^0 = (2,-2)$, one gets the point $\bar{x} = (2,-1)$, which also belongs to $\mathcal{S}$ (see Table 2.1 b)).

For Algorithm B, one can select $\rho = 2$. Then, the objective function of (P) can be represented as $f(x) = g(x) - h(x)$, where

$$g(x) = \frac{1}{2}x^T(Q + \rho E)x + q^T x = \frac{1}{2}(3x_1^2 + x_2^2) - x_1$$

and $h(x) = \frac{1}{2}x^T(\rho E)x = x_1^2 + x_2^2$. To implement Algorithm B, put $x^0 = (2,2)$ and set $k = 0$. One solves the convex program (2.9) and gets $x^1 = \big(\frac{28}{13}, \frac{14}{13}\big)$. Since $\|x^1 - x^0\| > \varepsilon$, one increase k by 1 and computes x^2. By solving the problem (2.9), one obtains $x^2 = (2,1)$. The stopping criterion in Step 3 is not satisfied. Therefore, one sets $k = 2$ and goes to Step 2. Using (2.9), one has $x^3 = (2,1)$. Hence, the condition $\|x^3 - x^2\| \leq \varepsilon$ is satisfied. So,

Table 2.1: The performance of Algorithm A

k	x^k	y^k	$f(x^k)$
0	(2.000000, 2.000000)	(0.000000, 4.000000)	-2.000000
1	(2.400000, 1.200000)	(0.000000, 2.400000)	-0.240000
2	(2.000000, 1.000000)	(0.000000, 2.000000)	-0.500000
3	(2.000000, 1.000000)	(0.000000, 2.000000)	-0.500000

a) $x^0 = (2, 2)$

k	x^k	y^k	$f(x^k)$
0	(2.000000, -2.000000)	(0.000000, -4.000000)	-2.000000
1	(2.400000, -1.200000)	(0.000000, -2.400000)	-0.240000
2	(2.000000, -1.000000)	(0.000000, -2.000000)	-0.500000
3	(2.000000, -1.000000)	(0.000000, -2.000000)	-0.500000

b) $x^0 = (2, -2)$

Table 2.2: The performance of Algorithm B

k	x^k	$f(x^k)$
0	(2.000000, 2.000000)	-2.000000
1	(2.023530, 1.011765)	-0.488028
2	(2.000000, 1.000000)	-0.500000
3	(2.000000, 1.000000)	-0.500000

a) $x^0 = (2, 2)$

k	x^k	$f(x^k)$
0	(2.000000, -2.000000)	-2.000000
1	(2.023530, -1.011765)	-0.488028
2	(2.000000, -1.000000)	-0.500000
3	(2.000000, -1.000000)	-0.500000

b) $x^0 = (2, -2)$

the computation stops after 3 steps and one gets the point $\bar{x} = (2, 1)$, which belongs to $\mathcal{S}$ (see Table 2.2 a)). Similarly, with the initial point $x^0 = (2, -2)$, one gets the point $\bar{x} = (2, -1)$, which also belongs to $\mathcal{S}$ (see Table 2.2 b)).

2.2 Convergence and Convergence Rate of the Algorithm

As noted in Section 2.1, the KKT point set C^* of (2.1) is the solution set of the affine variational inequality (2.6), so C^* is the union of finitely many polyhedral convex sets (see, e.g., [65, Lemma 3.1] and [50, Sections 3.1 and 5.3]). In particular, C^* has finitely many connected components. Since the solution set $\mathcal{S}$ of (2.1) is a subset of C^*, if $\mathcal{S}$ is nonempty then $C^* \neq \emptyset$. For any given subset $M \subset \mathbb{R}^n$, by $d(x, M) := \inf\{\|x - y\| \mid y \in M\}$ we denote the *distance* from $x \in \mathbb{R}^n$.

We will need two auxiliary results. The next lemma gives a local error bound for the distance $d(x, C^*)$ form a feasible point $x \in C$ to C^*.

Lemma 2.1 ([92, Lemma 2.1]; cf. [65, Lemma 3.1]) *For any $\rho > 0$, if $C^* \neq \emptyset$, then there exist scalars $\varepsilon > 0$ and $\ell > 0$ such that*

$$d(x, C^*) \leq \ell \left\| x - P_C\Big(x - \frac{1}{\rho}(Qx + q)\Big)\right\| \tag{2.10}$$

for all $x \in C$ with

$$\left\| x - P_C\Big(x - \frac{1}{\rho}(Qx + q)\Big)\right\| \leq \varepsilon. \tag{2.11}$$

Lemma 2.2 ([65, Lemma 3.1]; see also [92, Lemma 2.2]) *Let $C_1, C_2, \cdots, C_r$ denote the connected components of C^*. Then we have*

$$C^* = \bigcup_{i=1}^{r} C_i,$$

and the following properties are valid:

(a) *each C_i is the union of finitely many polyhedral convex sets;*

(b) *the sets C_i, $i = 1, \ldots r$, are properly separated each from others, that is, there exists $\delta > 0$ such that if $i \neq j$ then*

$$d(x, C_j) \geq \delta \quad \forall x \in C_i;$$

(c) *f is constant on each C_i.*

The necessary and sufficient condition for x^{k+1} to be the unique solution of (2.9) is the following

$$\langle \nabla\psi(x^{k+1}), x - x^{k+1}\rangle \geq 0 \quad \forall x \in C,$$

where $\nabla\psi(x^{k+1}) = Qx^{k+1} + q + \rho x^{k+1} - \rho x^k$. Equivalently, x^{k+1} is the unique solution of the strongly monotone affine variational inequality given by the affine operator $x \mapsto (Q + \rho E)x + q - \rho x^k$ and the polyhedral convex set C. Therefore, applying Theorem 2.3 from [45, p. 9] we see that x^{k+1} is *the unique fixed point* of the map $G_k(x) := P_C(x - \mu(Mx + q^k))$, where $\mu > 0$ is arbitrarily chosen, $M := Q + \rho E$, and $q^k := q - \rho x^k$. In what follows, we choose $\mu = \rho^{-1}$. Then

$$x^{k+1} = P_C\Big(x^{k+1} - \frac{1}{\rho}(Mx^{k+1} + q^k)\Big). \tag{2.12}$$

The convergence and the rate of convergence of Algorithm B, the Proximal DC decomposition algorithm, can be formulated as follows.

Theorem 2.2 *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 $\limsup_{k\to\infty} \|x^k - \bar{x}\|^{1/k} < 1$.*

Proof. Since (2.1) has a solution, $C^* \neq \emptyset$. Hence, by Lemma 2.1 there exist $\ell > 0$ and $\varepsilon > 0$ such that (2.10) is fulfilled for any x satisfying (2.11). As $\inf_{x\in C} f(x) > -\infty$, assertion (iv) of Theorem 2.1 gives

$$\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0. \tag{2.13}$$

Choose $k_0 \in \mathbb{N}$ as large as $\|x^{k+1} - x^k\| < \varepsilon$ for all $k \geq k_0$.

If it holds that

$$\|x^{k+1} - P_C(x^{k+1} - \frac{1}{\rho}(Qx^{k+1} + q))\| \leq \varepsilon \quad \forall k \geq k_0, \tag{2.14}$$

then by (2.10) one has

$$d(x^{k+1}, C^*) \leq \ell\|x^{k+1} - P_C\Big(x^{k+1} - \frac{1}{\rho}(Qx^{k+1} + q)\Big)\| \quad \forall k \geq k_0. \tag{2.15}$$

To obtain (2.14), for any $k \geq k_0$, we recall that

$$x^{k+1} = G_k(x^{k+1}) = P_C\Big(x^{k+1} - \frac{1}{\rho}(Mx^{k+1} + q^k)\Big), \tag{2.16}$$

Combining this with the nonexpansiveness of $P_C(.)$ [45, Corollary 2.4, p. 10] yields

$$\begin{aligned}
&\|x^{k+1} - P_C(x^{k+1} - \tfrac{1}{\rho}(Qx^{k+1} + q))\| \\
&\leq \|P_C\Big(x^{k+1} - \frac{1}{\rho}(Mx^{k+1} + q^k)\Big) - P_C\Big(x^{k+1} - \frac{1}{\rho}(Qx^{k+1} + q)\Big)\| \\
&\leq \|[x^{k+1} - \frac{1}{\rho}(Mx^{k+1} + q^k)] - [x^{k+1} - \frac{1}{\rho}(Qx^{k+1} + q)]\| \\
&= \|[x^{k+1} - \frac{1}{\rho}(Qx^{k+1} + \rho x^{k+1} + q - \rho x^k)] - [x^{k+1} - \frac{1}{\rho}(Qx^{k+1} + q)]\| \\
&= \|x^{k+1} - x^k\| < \varepsilon.
\end{aligned}$$

Hence (2.14) is valid and, in addition, we have

$$\|x^{k+1} - P_C\Big(x^{k+1} - \frac{1}{\rho}(Qx^{k+1} + q)\Big)\| \leq \|x^{k+1} - x^k\|.$$

From this and (2.15) it follows that

$$d(x^{k+1}, C^*) \leq \ell\|x^{k+1} - x^k\| \quad \forall k \geq k_0. \tag{2.17}$$

Since C^* is closed and nonempty, for each $k \in \{0, 1, 2, \dots\}$ we can find $y^k \in C^*$ such that $d(x^k, C^*) = \|x^k - y^k\|$. Then (2.17) implies that

$$\|x^{k+1} - y^{k+1}\| \leq \ell\|x^{k+1} - x^k\| \quad \forall k \geq k_0. \tag{2.18}$$

So, as consequence of (2.13),

$$\lim_{k\to\infty} \|y^{k+1} - x^{k+1}\| = 0. \tag{2.19}$$

Since

$$\|y^{k+1} - y^k\| \leq \|y^{k+1} - x^{k+1}\| + \|x^{k+1} - x^k\| + \|x^k - y^k\|,$$

it follows that

$$\lim_{k\to\infty} \|y^{k+1} - y^k\| = 0. \tag{2.20}$$

Let $C_1, C_2, \cdots, C_r$ be the connected components of C^*. By Lemma 2.2 and (2.20), there exist $i_0 \in \{1, \dots, r\}$ and $k_1 \geq k_0$ such that $y^k \in C_{i_0}$ for every $k \geq k_1$. Hence, according to the third assertion of Lemma 2.2,

$$f(y^k) = c \quad \forall k \geq k_1 \tag{2.21}$$

for some $c \in \mathbb{R}$.

Since (2.1) has a solution, by Theorem 2.1 we can find a real value f_* such that $\lim_{k\to\infty} f(x^k) = f_*$.

By the classical Mean Value Theorem and by the formula $\nabla f(x) = Qx+q$, for every k there is $z^k \in (x^k, y^k) := \{(1-t)x^k + ty^k \mid 0 < t < 1\}$ such that

$$f(y^k) - f(x^k) = \langle Qz^k + q, y^k - x^k \rangle.$$

Since y^k is a KKT point, it holds that $0 \le \langle Qy^k + q, x^k - y^k \rangle$. Adding this inequality and the preceding equality, we get

$$\begin{aligned} f(y^k) - f(x^k) &\le \langle Q(z^k - y^k), y^k - x^k \rangle \\ &\le \|Q\| \|z^k - y^k\| \|y^k - x^k\| \\ &\le \|Q\| \|y^k - x^k\|^2. \end{aligned} \tag{2.22}$$

On one hand, from (2.21) and (2.22) it follows that

$$c = f(y^k) \le f(x^k) + \|Q\| \, \|y^k - x^k\|^2.$$

As $\lim\limits_{k\to\infty} \left[f(x^k) + \|Q\| \, \|y^k - x^k\|^2\right] = f_*$ due to (2.19), this forces

$$c \le f_*. \tag{2.23}$$

On the other hand, since $x^{k+1} = P_C\Big(x^{k+1} - \dfrac{1}{\rho}(Mx^{k+1} + q^k)\Big)$ by (2.16), the characterization of the metric projection on a closed convex set [45, Theorem 2.3, p. 9] gives us

$$\left\langle \left[x^{k+1} - \frac{1}{\rho}(Mx^{k+1} + q^k)\right] - x^{k+1}, y - x^{k+1} \right\rangle \le 0 \quad \forall y \in C.$$

Therefore,

$$\left\langle Mx^{k+1} + q^k, y^{k+1} - x^{k+1} \right\rangle \ge 0 \quad \forall k \in \mathbb{N}.$$

From this and (2.18) we get

$$\begin{aligned} &\langle My^{k+1} + q^k, x^{k+1} - y^{k+1} \rangle \\ &\le \langle My^{k+1} + q^k, x^{k+1} - y^{k+1} \rangle + \langle Mx^{k+1} + q^k, y^{k+1} - x^{k+1} \rangle \\ &= \langle M(y^{k+1} - x^{k+1}), x^{k+1} - y^{k+1} \rangle \\ &\le \|M\| \|y^{k+1} - x^{k+1}\|^2 \\ &\le \ell^2 \|M\| \|x^{k+1} - x^k\|^2 \end{aligned}$$

for all $k \ge k_0$. So, setting $\alpha = \ell^2 \|M\|$, we have

$$\langle My^{k+1} + q^k, x^{k+1} - y^{k+1} \rangle \le \alpha \|x^{k+1} - x^k\|^2. \tag{2.24}$$

For each $k \ge k_1$, since $M = Q + \rho E$ and $q^k = q - \rho x^k$, invoking (2.24) and

using (2.18) once more, we have

$$\begin{aligned} f(x^{k+1}) - c &= f(x^{k+1}) - f(y^{k+1}) \\ &\leq \tfrac{1}{2}\langle Qx^{k+1}, x^{k+1}\rangle + \langle q, x^{k+1}\rangle - \tfrac{1}{2}\langle Qy^{k+1}, y^{k+1}\rangle - \langle q, y^{k+1}\rangle \\ &= \langle My^{k+1} + q^k, x^{k+1} - y^{k+1}\rangle + \tfrac{1}{2}\langle Q(x^{k+1} - y^{k+1}), x^{k+1} - y^{k+1}\rangle \\ &\quad +\rho\langle x^k - y^{k+1}, x^{k+1} - y^{k+1}\rangle \\ &= \langle My^{k+1} + q^k, x^{k+1} - y^{k+1}\rangle + \tfrac{1}{2}\langle Q(x^{k+1} - y^{k+1}), x^{k+1} - y^{k+1}\rangle \\ &\quad +\rho\langle x^k - x^{k+1}, x^{k+1} - y^{k+1}\rangle + \rho\langle x^{k+1} - y^{k+1}, x^{k+1} - y^{k+1}\rangle \\ &\leq \alpha\|x^{k+1} - x^k\|^2 + \tfrac{1}{2}\|Q\|\|x^{k+1} - y^{k+1}\|^2 \\ &\quad +\rho\|x^{k+1} - x^k\|\|x^{k+1} - y^{k+1}\| + \rho\|x^{k+1} - y^{k+1}\|^2 \\ &\leq \left[\alpha + \tfrac{1}{2}\|Q\|\ell^2 + \rho\ell(1+\ell)\right]\|x^{k+1} - x^k\|^2. \end{aligned}$$

Therefore, with $\beta := \alpha + \frac{1}{2}\|Q\|\ell^2 + \rho\ell(1+\ell)$, we get

$$f(x^{k+1}) \leq c + \beta\|x^{k+1} - x^k\|^2. \tag{2.25}$$

Letting $k \to \infty$, from (2.25) we can deduce that

$$f_* = \lim_{k\to\infty} f(x^{k+1}) \leq c.$$

Combining the last expression with (2.23) yields $f_* = c$. Therefore, by (2.25) and the first assertion of Theorem 2.1 we obtain

$$f(x^{k+1}) - f_* \leq \beta\|x^{k+1} - x^k\|^2 \leq \frac{2\beta}{\lambda_1(Q_1) + \lambda_1(Q_2)}(f(x^k) - f(x^{k+1})),$$

where $Q_1 = Q + \rho E$ and $Q_2 = \rho E$. Putting $\gamma = \lambda_1(Q_1) + \lambda_1(Q_2)$, from the condition $\rho > -\lambda_1(Q)$ we get $\gamma = (\lambda_1(Q) + \rho) + \rho > 0$. Therefore,

$$f(x^{k+1}) - f_* \leq \frac{2\beta}{\gamma}\left[(f(x^k) - f_*) - (f(x^{k+1}) - f_*)\right].$$

Hence

$$f(x^{k+1}) - f_* \leq \frac{2\beta}{2\beta + \gamma}(f(x^k) - f_*).$$

So we have

$$|f(x^{k+1}) - f_*| \leq \mu_0|f(x^k) - f_*| \quad \forall\, k \geq k_1,$$

where $\mu_0 := \frac{2\beta}{2\beta+\gamma} \in (0,1)$. Thus,

$$|f(x^k) - f_*| \leq \mu_0^{k-k_1}|f(x^{k_1}) - f_*| \quad \forall\, k > k_1,$$

or

$$|f(x^k) - f_*| \leq r_0\,\mu^{2k} \quad \forall\, k > k_1,$$

where $r_0 := \mu_0^{-k_1}|f(x^{k_1}) - f_*|$ and $\mu := \mu_0^{1/2}$. Hence,

$$\begin{aligned} |f(x^{k+1}) - f(x^k)| &\le |f(x^{k+1}) - f_*| + |f(x^k) - f_*| \\ &\le r_0\,\mu^{2k+2} + r_0\,\mu^{2k} = r_1\mu^{2k} \quad \forall k > k_1, \end{aligned}$$

where $r_1 := r_0(\mu^2+1)$. Consequently, using the first assertion of Theorem 2.1 once more, we see that

$$\|x^{k+1} - x^k\|^2 \le \frac{2}{\gamma}(f(x^k) - f(x^{k+1})) \le \frac{2r_1}{\gamma}\mu^{2k} \quad \forall k > k_1.$$

Thus

$$\|x^{k+1} - x^k\| \le r\,\mu^k \quad \forall k > k_1,$$

where $r := \left(\frac{2r_1}{\gamma}\right)^{\frac{1}{2}}$ and $\mu \in (0,1)$. Let $\varepsilon > 0$ be given arbitrarily. For each positive integer p, we have

$$\begin{aligned} \|x^{k+p} - x^k\| &\le \|x^{k+p} - x^{k+p-1}\| + \cdots + \|x^{k+1} - x^k\| \\ &\le r\,\mu^{k+p-1} + \cdots + r\mu^k \\ &= r\frac{1-\mu^p}{1-\mu}\mu^k \le \frac{r}{1-\mu}\,\mu^k < \varepsilon, \end{aligned}$$

provided that k is large enough. Hence $\{x^k\}$ is a Cauchy sequence, and we may assume that it converges to a point $\bar{x} \in C$. By the third assertion of Theorem 2.1, $\bar{x} \in C^*$. Moreover, passing the inequality

$$\|x^{k+p} - x^k\| \le \frac{r}{1-\mu}\,\mu^k$$

to the limit as $p \to \infty$, we get

$$\|x^k - \bar{x}\| \le \frac{r}{1-\mu}\,\mu^k$$

for all k large enough. So,

$$\|x^k - \bar{x}\|^{1/k} \le \left(\frac{r}{1-\mu}\right)^{1/k} \mu$$

for all k large enough. Therefore,

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

This proves that $\{x^k\}$ converges R-linearly to a KKT point of (2.1). □

Remark 2.2 According to Theorem 2.2, one can find a constant C such that $\limsup_{k\to\infty} \|x^k - \bar{x}\|^{1/k} < C < 1$. So, if the computation is terminated at step

k, provided that k is sufficiently large, then one has $\|x^k - \bar{x}\|^{1/k} < C$. That is, $\|x^k - \bar{x}\| < C^k$. Therefore, the *computation error* between the obtained approximate solution x^k and the exact limit point $\bar{x}$ of the sequence $\{x^k\}$ is smaller than the number C^k. Since $C \in (0, 1)$, one sees that the *computation error bound* C^k tends to 0 as $k \to \infty$.

2.3 Asymptotical Stability of the Algorithm

We will prove that DCA sequences generated by Algorithm B converge to a locally unique solution of (2.1) if the initial points are taken from a suitably-chosen neighborhood of it.

First, we have to recall a stability concept that works for discrete dynamical system. Consider *an iteration algorithm* which generates a unique point x^{k+1}, provided that the preceding iteration point x^k, $k \in \{0, 1, 2, \dots\}$, has been defined. Following Leong and Goh [59, Definition 2], we can present the concept of *asymptotic stability* of a KKT point as follows.

Definition 2.2 The KKT point $\bar{x}$ of (2.1) is:

(i) *stable* w.r.t. the iteration algorithm if for any given $\varepsilon > 0$ there exists $\delta > 0$ such that whenever $x^0 \in B(\bar{x}, \delta)$, the DCA sequence generated by the iteration algorithm and the initial point x^0 has the property $x^k \in B(\bar{x}, \varepsilon)$ for all $k \geq 0$;

(ii) *attractive* if there exists $\delta > 0$ such that whenever $x^0 \in B(\bar{x}, \delta)$, the DCA sequence generated by the iteration algorithm and the initial point x^0 has the property $\lim\limits_{k\to\infty} x_k = \bar{x}$;

(iii) *asymptotically stable* w.r.t. the iteration algorithm if it is stable and attractive w.r.t. to that algorithm.

As usual, for an optimization problem $\min\{g(x) \mid x \in \Omega\}$ with $g : \mathbb{R}^n \to \mathbb{R}$ and $\Omega \subset \mathbb{R}^n$ being respectively a real function and an arbitrary subset, one says that $\bar{x} \in \Omega$ is a *locally unique solution* of if there exists $\varepsilon > 0$ such that

$$g(x) > g(\bar{x}) \quad \forall x \in (\Omega \cap B(\bar{x}, \varepsilon)) \setminus \{\bar{x}\}.$$

The next two lemmas express some well-known facts.

Lemma 2.3 (See, e.g., [50, Theorem 3.8]) *If $\bar{x} \in C$ is a locally unique solution of (2.1), then there exist $\mu > 0$ and $\eta > 0$ such that*

$$f(x) - f(\bar{x}) \geq \eta \|x - \bar{x}\|^2 \quad \text{for every } x \in C \cap B(\bar{x}, \mu). \tag{2.26}$$

Lemma 2.4 (See, e.g., [17, Proof of Lemma 4] and [57, Lemma 1]) *If the KKT point set C^* contains a segment $[u, x]$, then the restriction of f on that segment is a constant function.*

The main result of this section can be formulated as follows.

Theorem 2.3 *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 $\gamma > 0$ 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) $x^k \in C \cap B(\bar{x}, \gamma)$ *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.

Proof. Suppose that $\rho > \|Q\|$ and $\bar{x}$ is a locally unique solution of (2.1). By Lemma 2.3 we can select constants $\mu > 0$ and $\eta > 0$ such that (2.26) holds. For any given $\gamma > 0$, by replacing γ with a smaller one (if necessary), we may assume that $\gamma \in (0, \mu)$ and $\gamma < \mu(1 - \rho^{-1}\|Q\|)$. Since

$$f(x) - f(\bar{x}) > 0 \quad \forall x \in \big(C \cap B(\bar{x}, \gamma)\big) \setminus \{\bar{x}\}$$

by (2.26), the continuity of f implies the existence of $\delta \in (0, \mu)$ satisfying

$$\frac{1}{\eta^{1/2}}\big(f(x) - f(\bar{x})\big)^{1/2} < \gamma \quad \forall x \in C \cap B(\bar{x}, \delta). \tag{2.27}$$

First, let us show that the assertion about stability of DCA sequences generated by Algorithm B is valid for the chosen number $\delta > 0$. Fix any $x^0 \in C \cap B(\bar{x}, \delta)$. As $\delta < \gamma$, for $k = 0$ we have $x^k \in C \cap B(\bar{x}, \gamma)$. To proceed by induction, suppose that the last inclusion holds for some $k \geq 0$. Since $\bar{x}$ is a locally unique solution of (2.1), it is a KKT point of that problem, i.e.,

$$\big(Q\bar{x} + q\big)^T(x - \bar{x}) \geq 0 \quad \forall x \in C. \tag{2.28}$$

It follows that

$$\bar{x} = P_C\big(\bar{x} - \frac{1}{\rho}(Q\bar{x} + q)\big). \tag{2.29}$$

Indeed, by the characterization of the metric projection [45, Theorem 2.3, p. 9], (2.29) is valid if and only if

$$\Big(\big[\bar{x} - \frac{1}{\rho}(Q\bar{x} + q)\big] - \bar{x}\Big)^T (x - \bar{x}) \le 0 \quad \forall x \in C.$$

The latter is equivalent to (2.28). Using (2.12), (2.29), and the nonexpansiveness of the metric projection [45, Corollary 2.4, p. 10], we have

$$\begin{aligned}
\|x^{k+1} - \bar{x}\| &= \big\|P_C\big(x^{k+1} - \frac{1}{\rho}(Mx^{k+1} + q^k)\big) - P_C\big(\bar{x} - \frac{1}{\rho}(Q\bar{x} + q)\big)\big\| \\
&\le \big\|\big[x^{k+1} - \frac{1}{\rho}(Mx^{k+1} + q^k)\big] - \big[\bar{x} - \frac{1}{\rho}(Q\bar{x} + q)\big]\big\| \\
&= \big\|\big[x^{k+1} - \frac{1}{\rho}\big((\rho E + Q)x^{k+1} + q - \rho x^k\big)\big] - \big[\bar{x} - \frac{1}{\rho}(Q\bar{x} + q)\big]\big\| \\
&= \big\|(x^k - \bar{x}) + \frac{1}{\rho}Q(\bar{x} - x^{k+1})\big\| \\
&\le \|x^k - \bar{x}\| + \frac{1}{\rho}\|Q\|\|\bar{x} - x^{k+1}\|.
\end{aligned}$$

Then we obtain

$$\|x^{k+1} - \bar{x}\| \le (1 - \frac{1}{\rho}\|Q\|)^{-1}\|x^k - \bar{x}\| \le (1 - \frac{1}{\rho}\|Q\|)^{-1}\gamma < \mu,$$

where the strict inequality follows from the property $\gamma < \mu(1-\rho^{-1}\|Q\|)$. Thus, $x^{k+1} \in C \cap B(\bar{x}, \mu)$. Applying (2.26) and the inequality $f(x^k) \ge f(x^{k+1})$ which holds for any $k \ge 0$ (see Remark 2.1), we get

$$\begin{aligned}
\|x^{k+1} - \bar{x}\|^2 &\le \frac{1}{\eta}\big(f(x^{k+1}) - f(\bar{x})\big) \\
&\le \frac{1}{\eta}\big(f(x^k) - f(\bar{x})\big) \\
&\vdots \\
&\le \frac{1}{\eta}\big(f(x^0) - f(\bar{x})\big).
\end{aligned}$$

Hence,

$$\|x^{k+1} - \bar{x}\| \le \frac{1}{\eta^{1/2}}\big(f(x^0) - f(\bar{x})\big)^{1/2}.$$

Since $x^0 \in C \cap B(\bar{x}, \delta)$, combining this with (2.27) we obtain $\|x^{k+1} - \bar{x}\| < \gamma$ which means that $x^{k+1} \in C \cap B(\bar{x}, \gamma)$. Thus, we have proved that

$$x^k \in C \cap B(\bar{x}, \gamma)$$

for every $k \ge 0$.

Next, to obtain the assertion about the attractiveness of DCA sequences generated by Algorithm B, we observe by the just obtained stability result that for any $\gamma > 0$ there exists $\delta = \delta(\gamma) > 0$ 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 the property in (a) is valid. Without loss of generality, we may assume that $\gamma \in (0, \mu)$ and $\delta \in (0, \gamma)$. By taking a smaller positive $\gamma > 0$ and choosing the corresponding $\delta = \delta(\gamma)$ such that the property in (a) is valid, we can have the following: 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 the property in (b) holds. Indeed, if this claim was false, we would find sequences $\gamma_j \to 0^+$ and $\delta_j \to 0^+$ such that for each $j \in \mathbb{N}$ we have $\gamma_j \in (0, \mu)$, $\delta_j \in (0, \gamma_j)$, and the stability assertion is valid for the pair $(\delta, \gamma) := (\delta_j, \gamma_j)$. Moreover, for each j, there exists some $x^{0,j} \in C \cap B(\bar{x}, \delta_j)$ such that the DCA sequence $\{x^{k,j}\}$ generated by Algorithm B and the initial point $x^{0,j}$ does not converge to $\bar{x}$. Then we can select a subsequence of $\{x^{k,j}\}$ which converges to a point

$$\widetilde{x}^j \in C \cap \bar{B}(\bar{x}, \gamma_j) \subset C \cap B(\bar{x}, \mu), \tag{2.30}$$

where $\widetilde{x}^j \neq \bar{x}$. By Theorem 2.1 we have $\widetilde{x}^j \in C^*$ for $j = 1, 2, \ldots$. Observe that

$$\lim_{j \to \infty} \widetilde{x}^j = \bar{x}. \tag{2.31}$$

For each j, one can find an integer $k(j) \geq 1$ such that $\gamma_{j+k(j)} < \|\widetilde{x}^j - \bar{x}\|$. Then, by (2.30) one has

$$\|\widetilde{x}^{j+k(j)} - \bar{x}\| < \|\widetilde{x}^j - \bar{x}\|.$$

Choose $z^1 = \widetilde{x}^1$ and set $z^{p+1} := \widetilde{x}^{p+k(p)}$ for $p = 1, 2, \ldots$. It is clear that $\{z^p\}$ is a subsequence of $\{\widetilde{x}^j\}$ and $z^p \neq z^{p'}$ whenever $p' \neq p$. Hence, by considering a subsequence (if necessary), we can assume that $\widetilde{x}^j \neq \widetilde{x}^\ell$ whenever $j \neq \ell$. Since the number of pseudo-faces of C is finite, by (2.31) there must exists an index set $\alpha \subset \{1, \ldots, m\}$ such that the pseudo-face

$$F_\alpha := \{x \in \mathbb{R}^n \mid A_\alpha x = b_\alpha,\ A_{\bar{\alpha}} x > b_{\bar{\alpha}}\}$$

of C contains infinite number of the members of the sequence $\{\widetilde{x}^j\}$. Without loss of generality, we may assume that the whole sequence $\{\widetilde{x}^j\}$ is contained in F_α. By [50, Lemma 4.1], the intersection $C^* \cap F_\alpha$ is a convex set. Hence, according to Lemma 2.4, the restriction of f on $C^* \cap F_\alpha$ is a constant function. Using (2.31), from this we can deduce that the equality $f(\widetilde{x}^j) = f(\bar{x})$ holds

for all j. As $\widetilde{x}^j \neq \bar{x}$ for every j, the last equality contradicts (2.26). Our claim has been proved. □

To illustrate asymptotical stability of Algorithmthe B, let us consider the following example.

Example 2.2 (see [50, Example 11.3, p. 207]) Consider problem (2.1) with $n = 2$, $m = 2$, $Q = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, $A = \begin{bmatrix} 1 & -2 \\ 1 & 2 \end{bmatrix}$, $q = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$, $b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. Here, one has the objective function $f(x) = \frac{1}{2}(x_1^2 - x_2^2) - x_1$ over the set

$$C = \{x \in \mathbb{R}^2 \mid x_1 - 2x_2 \geq 0,\ x_1 + 2x_2 \geq 0\}.$$

Since $\lambda_1 = -1$ and $\lambda_2 = 1$ are eigenvalues of Q, one can choose $\rho = 2$. Using (2.4), one obtains the KKT point set $C^* = \left\{(1,0), (\frac{4}{3}, \frac{2}{3}), (\frac{4}{3}, -\frac{2}{3})\right\}$. For this problem, one has $\mathcal{S}$=loc(P)=$\{(\frac{4}{3}, \frac{2}{3}), (\frac{4}{3}, -\frac{2}{3})\}$. One selects initial point, say, $x^0 = (\frac{3}{2}, \frac{1}{2})$, and chooses $\bar{x} = (\frac{4}{3}, \frac{2}{3}) \in \mathcal{S}$. Let the tolerance $\varepsilon > 0$ be small enough and put $\delta = \|x^0 - \bar{x}\| = \frac{\sqrt{2}}{6}$. Hence, $x^0 \in C \cap B(\bar{x}, \delta)$. Let $\{x^k\}$ be the DCA sequence generated by Algorithm B and the initial point x^0. For each $k \in \{0, \ldots, 27\}$, one has $\|x^k - \bar{x}\| \leq \varepsilon$; so $x^k \in C \cap B(\bar{x}, \gamma)$, where $\gamma = \varepsilon$ (see Table 2.3 a) and Figure 2.1). Similar results are valid if one choses $x^0 = (\frac{3}{2}, -\frac{1}{5})$ and $\bar{x} = (\frac{4}{3}, -\frac{2}{3})$ (see Table 2.3 b)).

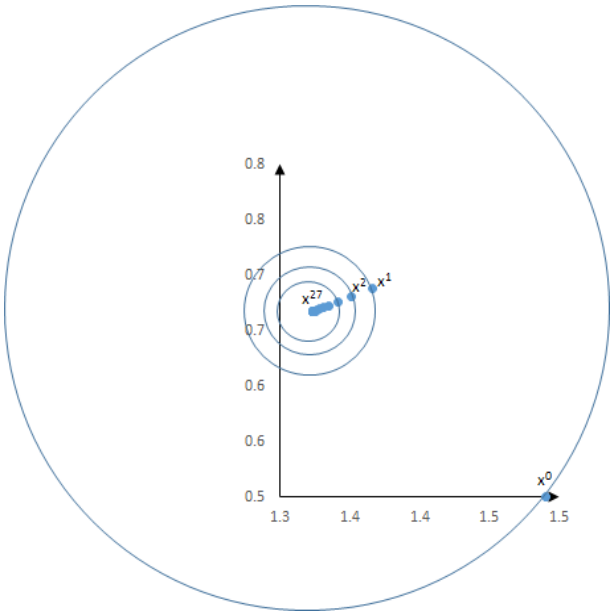

Figure 2.1: The DCA sequence generated by Algorithm B and $x^0 = (1.5, 0.5)$

Table 2.3: Asymptotical stability of Algorithm B

k	x^k	$f(x^k)$	$\delta(\gamma)$
0	(1.500000, 0.500000)	-0.500000	0.235702
1	(1.376471, 0.688235)	-0.665969	0.048228
2	(1.361246, 0.680623)	-0.666375	0.031206
3	(1.351394, 0.675697)	-0.666544	0.020192
4	(1.345020, 0.672510)	-0.666615	0.013065
5	(1.340895, 0.670448)	-0.666645	0.008454
6	(1.338226, 0.669113)	-0.666658	0.005470
7	(1.336499, 0.668250)	-0.666663	0.003539
8	(1.335382, 0.667691)	-0.666665	0.002290
9	(1.334659, 0.667329)	-0.666666	0.001481
10	(1.334191, 0.667096)	-0.666666	0.000958
11	(1.333888, 0.666944)	-0.666667	0.000620
12	(1.333692, 0.666846)	-0.666667	0.000401
13	(1.333566, 0.666783)	-0.666667	0.000259
14	(1.333484, 0.666742)	-0.666667	0.000167
15	(1.333431, 0.666715)	-0.666667	0.000108
16	(1.333396, 0.666698)	-0.666667	0.000070
17	(1.333374, 0.666687)	-0.666667	0.000045
18	(1.333360, 0.666680)	-0.666667	0.000029
19	(1.333351, 0.666675)	-0.666667	0.000019
20	(1.333345, 0.666672)	-0.666667	0.000012
21	(1.333341, 0.666670)	-0.666667	0.000007
22	(1.333338, 0.666669)	-0.666667	0.000005
23	(1.333336, 0.666668)	-0.666667	0.000003
24	(1.333335, 0.666668)	-0.666667	0.000001
25	(1.333335, 0.666667)	-0.666667	0.000001
26	(1.333334, 0.666667)	-0.666667	0.000000
27	(1.333334, 0.666667)	-0.666667	0.000000

a) $x^0 = (1.5, 0.5)$

x^k	$f(x^k)$	$\delta(\gamma)$
(1.500000, -0.500000)	-0.500000	0.235702
(1.376471, -0.688235)	-0.665969	0.048228
(1.361246, -0.680623)	-0.666375	0.031206
(1.351394, -0.675697)	-0.666544	0.020192
(1.345020, -0.672510)	-0.666615	0.013065
(1.340895, -0.670448)	-0.666645	0.008454
(1.338226, -0.669113)	-0.666658	0.005470
(1.336499, -0.668250)	-0.666663	0.003539
(1.335382, -0.667691)	-0.666665	0.002290
(1.334659, -0.667329)	-0.666666	0.001481
(1.334191, -0.667096)	-0.666666	0.000958
(1.333888, -0.666944)	-0.666667	0.000620
(1.333692, -0.666846)	-0.666667	0.000401
(1.333566, -0.666783)	-0.666667	0.000259
(1.333484, -0.666742)	-0.666667	0.000167
(1.333431, -0.666715)	-0.666667	0.000108
(1.333396, -0.666698)	-0.666667	0.000070
(1.333374, -0.666687)	-0.666667	0.000045
(1.333360, -0.666680)	-0.666667	0.000029
(1.333351, -0.666675)	-0.666667	0.000019
(1.333345, -0.666672)	-0.666667	0.000012
(1.333341, -0.666670)	-0.666667	0.000007
(1.333338, -0.666669)	-0.666667	0.000005
(1.333336, -0.666668)	-0.666667	0.000003
(1.333335, -0.666668)	-0.666667	0.000001
(1.333335, -0.666667)	-0.666667	0.000001
(1.333334, -0.666667)	-0.666667	0.000000
(1.333334, -0.666667)	-0.666667	0.000000

b) $x^0 = (1.5, -0.5)$

2.4 Influence of the Decomposition Parameter

In this section, we will analyze the influence of the decomposition parameter ρ for the rates of convergence of the algorithms A and B. We also compare the effectiveness of Algorithm B with that of Algorithm A. These algorithms were implemented in the Visual C++ 2010 environment, and performed on a PC Intel Core^{TM} i7 (4×2.0 GHz) processor, 4GB RAM. The CPLEX 11.2 solver is used to solve linear and convex quadratic problems.

Recall that, for Algorithm A, the parameter $\rho > 0$ has to satisfy the inequality $\rho \geq \lambda_n(Q)$. For Algorithm B, $\rho > 0$ must satisfy the strict inequality $\rho > -\lambda_1(Q)$.

We have used the algorithms A and B to solve some test problems of the form (2.1) for the dimensions $n = 10$, $n = 20$, $n = 40$, $n = 60$, $n = 80$. With $\beta_i \in [0, 10]$ for $i = 1, \ldots, n$ being generated randomly, the following two types of constraint sets have been considered:

$$C = \Big\{x \in \mathbb{R}^n \ : \ x \geq 0, \ ix_i \geq \beta_i, \ i = 1, \ldots, n, \ \sum_{i=1}^{n} ix_i \leq 5000\Big\}$$

and

$$C = \Big\{x \in \mathbb{R}^n \ : \ x \geq 0, \ ix_i \geq \beta_i, \ i = 1, \ldots, n, \ 10 \leq x_1 + \sum_{i=2}^{n} 0.1ix_i \leq 100\Big\}.$$

Each of these sets can be represented as the solution set of the linear inequality system $Ax \geq b$ with a suitably chosen matrix $A \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^m$. Fixing a dimension $n \in \{10, 20, 40, 60, 80\}$, we generate randomly a symmetric matrix $Q \in \mathbb{R}^{n\times n}$ and a vector $q \in \mathbb{R}^n$ with the requirement that all their components belong to the segment $[0, 10]$. The initial point $x^0 \in \mathbb{R}^n$ is generated randomly with the requirement that all its components belong to the segment $[0, 5]$. Then, we start testing Algorithm A with $\rho = \lambda_n(Q)$ if $\lambda_n(Q) > 0$ and $\rho = 0.1$ otherwise. For our convenience, this ρ is called *the smallest decomposition parameter* for Algorithm A. Similarly, we start testing Algorithm B with $\rho = -\lambda_1(Q) + 0.1$ if $\lambda_1(Q) < 0$ and $\rho = 0.1$ otherwise. This ρ is said to be *the smallest decomposition parameter* for Algorithm B. *The stopping criterion is* $\|x^{k+1} - x^k\| \leq 10^{-6}$ *and the largest allowed number of steps is 1000.* After testing Algorithm A (resp., Algorithm B) for a decomposition parameter ρ, we increase ρ by 1.5 times and let the algorithm to run again.

Due to the space limitation, we only present the test results for $n = 10, 40, 80$.

In Table 2.4, the second rows of the sub-tables a) and b) correspond to the smallest decomposition parameters for Algorithm A and Algorithm B, respectively. The decomposition parameters of the test reported in the third rows are 1.5 times of the smallest decomposition parameters. The decomposition parameters of the test reported in the fourth rows are 1.5 times of the just mentioned decomposition parameters; and so on... In the sub-tables a) and b), the first column presents the ordinal number of the tests. The second one indicates the numbers of iterations. The third one reports the running times, while the fourth column contains the decomposition parameters. Table 2.4 reports the computation results when Algorithm A and Algorithm B are applied to the same problem with the same initial point. Only 11 records are shown, because the 12th record would tell us that Algorithm A requires more than 1000 steps to complete the computation.

The contents of Tables 2.5–2.9 are similar to those of Table 2.4.

With any n belonging to the set $\{10, 20, 40, 60, 80\}$, a careful analysis of these Tables allows us to observe that:

• *For both algorithms, if ρ increases, then the running time, as well as the number of computation steps, increases;*

• *For any row of the sub-tables a) and b) with the same ordinal number, the number of steps required by Algorithm B is much smaller than that required by Algorithm A.*

• *For any row of the sub-tables a) and b) with the same ordinal number, the running time of Algorithm B is much smaller than that of Algorithm A.*

Thus, in terms of the number of computation steps and the execution time, Algorithm B is much more efficient than Algorithm A when the algorithms are applied to the same problem.

Table 2.4: The test results for $n = 10$ with the 1st type constraint

No.	Step	Time	ρ
1	5	0.239	48.802
2	12	0.222	73.203
3	22	0.274	109.805
4	37	0.416	164.707
5	59	0.718	247.060
6	91	0.947	370.590
7	139	1.364	555.886
8	210	2.050	833.829
9	316	3.019	1250.743
10	474	4.593	1876.114
11	710	7.006	2814.171

a)

No.	Step	Time	ρ
1	4	0.127	9.380
2	4	0.125	14.070
3	5	0.114	21.105
4	6	0.135	31.658
5	8	0.210	47.487
6	10	0.227	71.231
7	13	0.296	106.846
8	17	0.419	160.269
9	24	0.576	240.404
10	34	0.787	360.606
11	49	1.312	540.909

b)

Table 2.5: The test results for $n = 10$ with the 2nd type constraint

No.	Step	Time	ρ
1	3	0.189	47.763
2	7	0.210	71.644
3	13	0.285	107.467
4	21	0.233	161.200
5	33	0.335	241.800
6	51	0.527	362.700
7	77	0.729	544.049
8	115	1.029	816.074
9	171	1.802	1224.111
10	255	2.363	1836.167
11	380	3.637	2754.250
12	567	5.133	4131.375
13	847	7.546	6197.063

a)

No.	Step	Time	ρ
1	3	0.131	15.645
2	4	0.175	23.468
3	4	0.167	35.201
4	6	0.252	52.802
5	7	0.206	79.203
6	9	0.329	118.805
7	12	0.298	178.207
8	16	0.506	267.310
9	22	0.830	400.966
10	31	1.073	601.449
11	44	1.043	902.173
12	65	1.543	1353.259
13	95	2.628	2029.889

b)

Table 2.6: The test results for $n = 40$ with the 1st type constraint

No.	Step	Time	ρ
1	8	0.621	194.883
2	20	0.664	292.324
3	65	1.498	657.729
4	106	2.256	986.594
5	167	3.255	1479.891
6	259	4.925	2219.837
7	397	7.451	3329.755
8	604	11.236	4994.632
9	915	17.078	7491.948

a)

No.	Step	Time	ρ
1	5	0.320	32.917
2	6	0.386	49.375
3	7	0.454	74.062
4	8	0.509	111.094
5	11	0.670	166.641
6	15	0.947	249.961
7	20	1.238	374.941
8	28	1.734	562.412
9	40	2.477	843.618

b)

Table 2.7: The test results for $n = 40$ with the 2nd type constraint

No.	Step	Time	ρ
1	6	0.357	207.869
2	43	1.078	701.557
3	69	1.563	1052.336
4	107	2.408	1578.504
5	163	3.438	2367.756
6	373	7.227	5327.451
7	561	10.695	7991.177
8	843	15.936	11986.766

a)

No.	Step	Time	ρ
1	4	0.271	31.539
2	4	0.311	47.308
3	5	0.350	70.962
4	6	0.469	106.444
5	7	0.477	159.665
6	10	0.666	239.498
7	12	0.795	359.247
8	17	1.129	538.870

b)

Table 2.8: The test results for $n = 80$ with the 1st type constraint

No.	Step	Time	ρ
1	17	2.257	398.858
2	42	3.590	598.287
3	80	5.654	897.430
4	137	8.608	1346.145
5	223	12.446	2019.218
6	351	18.653	3028.826
7	543	29.408	4543.240
8	831	43.965	6814.859

a)

No.	Step	Time	ρ
1	6	1.329	46.645
2	6	1.309	69.967
3	8	1.904	104.951
4	11	2.415	157.426
5	14	3.210	236.139
6	19	4.730	354.208
7	27	6.244	531.312
8	38	7.713	796.969

b)

Table 2.9: The test results for $n = 80$ with the 2nd type constraint

No.	Step	Time	ρ
1	17	2.424	396.403
2	43	3.025	594.605
3	81	4.447	891.908
4	138	6.908	1337.862
5	222	9.914	2006.793
6	348	15.201	3010.189
7	536	22.813	4515.283
8	818	33.261	6772.925

a)

No.	Step	Time	ρ
1	7	1.787	109.550
2	10	2.545	164.325
3	14	3.285	246.488
4	19	4.677	369.732
5	26	6.597	554.598
6	38	8.805	831.898
7	56	13.169	1247.846
8	82	20.989	1871.770

b)

2.5 Conclusions

We have established two properties of Algorithm B for the IQP problem:

- Every DCA sequence generated by the Algorithm B must be bounded and, moreover, it converges R-linearly to a KKT point of the problem in question.

- Algorithm B that is asymptotically stable, provided that the initial point is close enough to a locally unique solution of the given problem and the DCA decomposition parameter satisfies a mild additional assumption.

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. Therefore, one should choose the smallest possible decomposition parameter.

- Algorithm B is more efficient 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 problem will be established in this chapter. Among other things, we will clarify the solution existence, properties of the global solutions, characteristic properties of the local solutions, locally Lipschitz property of the optimal value function, locally upper Lipschitz property of the global solution map, and the Aubin property of the local solution map.

This chapter is written on the basis of paper No. 2 in the List of author's related papers (see p. 112).

3.1 Clustering Problems

Clustering is an important task in data mining and it is a powerful tool for automated analysis of data. Cluster is a subset of the data set. The elements of a cluster are similar in some sense (see, e.g., [1, p. 32] and [43, p. 250]).

There are many kinds of clustering problems, where different criteria are used such as Euclidean distance [95], L_1-distance [8, 10], and square of the Euclidean distance. Among these criteria, *the* ***M****inimum* ***S****um-of-****S****quares* ***C****lustering* (MSSC for short) criterion is one of the most used [15, 18, 22, 28, 48, 60, 75, 87]. Biding by this criterion, one tries to make the sum of the

squared Euclidean distances from each data point to the centroid of its cluster as small as possible. The *MSSC problem* requires to partition a finite data set into a given number of clusters in order to minimize the just mentioned sum.

The importance of the MSSC problem was noticed by researchers long time ago and they have developed many algorithms to solve it (see, e.g., [6, 7, 9, 12, 13, 61, 71, 98], and the references therein). Since this is a NP-hard problem [3, 67], the effective existing algorithms reach at most local solutions. These algorithms may include certain techniques for improving the current data partition to seek better solutions. For example, in [71], the authors proposed a method to find good starting points that is based on the DCA (Difference-of Convex-functions Algorithms). The latter has been applied to the MSSC problem in [7, 52, 60].

The first aim of the present chapter is to prove some basic properties of the above problem. We begin with clarifying the equivalence between the mixed integer programming formulation and the unconstrained nonsmooth nonconvex optimization formulation of the problem, that were given in [71]. Then we prove that the MSSC problem always has a global solution and, under a mild condition, the global solution set is finite and the components of each global solution can be computed by an explicit formula.

The second aim of this chapter is to characterize the local solutions of the MSSC problem. Based on the necessary optimality condition in DC programming [26], some arguments of [71], and a newly introduced concept of *nontrivial local solution*, we get necessary conditions for a system of centroids to be a nontrivial local solution. Interestingly, we are able to prove that these necessary conditions are also sufficient ones. Since the known algorithms for solving the MSSC problem focus on the local solutions, our characterizations may lead to a better understanding and further refinements of the existing algorithms. Here, by constructing a suitable example, we investigate the performance of the k-means algorithm, which can be considered as a basic solution method for the MSSC problem.

The third aim of this chapter is to analyze the changes of the optimal value, the global solution set, and the local solution set of the MSSC problem with respect to small changes in the data set. Three principal stability properties will be established. Namely, we will prove that the optimal value function is

locally Lipschitz, the global solution map is locally upper Lipschitz, and the local solution map has the Aubin property, provided that the original data points are pairwise distinct.

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., [15, p. 266])

$$\psi(x, \alpha) := \frac{1}{m} \sum_{i=1}^{m} \Big(\sum_{j=1}^{k} \alpha_{ij} \|a^i - x^j\|^2 \Big) \quad \longrightarrow \quad \min,$$

where $\alpha_{ij} = 1$ if $a^i \in A^j$ and $\alpha_{ij} = 0$ otherwise, is used. Thus, the above partitioning problem can be formulated as the constrained optimization problem

$$\begin{aligned} \min \Big\{ \psi(x, \alpha) \;\; | \;\; & x \in \mathbb{R}^{nk}, \;\; \alpha = (\alpha_{ij}) \in \mathbb{R}^{m \times k}, \;\; \alpha_{ij} \in \{0, 1\}, \\ & \sum_{j=1}^{k} \alpha_{ij} = 1, \;\; i = 1, \ldots, m, \;\; j = 1, \ldots, k \Big\}, \end{aligned} \tag{3.1}$$

where the centroid system $x = (x^1, \ldots, x^k)$ and the incident matrix $\alpha = (\alpha_{ij})$ are to be found.

Since (3.1) is a difficult *mixed integer programming problem*, instead of it one usually considers (see, e.g., [71, p. 344]) the next *unconstrained nonsmooth nonconvex optimization problem*

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

Both models (3.1) and (3.2) are referred to as *the minimum sum-of-squares clustering problem* (the MSSC problem). As the decision variables of (3.1) and (3.2) belong to different Euclidean spaces, the equivalence between these minimization problems should be clarified. For our convenience, let us put $I = \{1, \ldots, m\}$ and $J = \{1, \ldots, k\}$.

3.2 Basic Properties of the MSSC Problem

Given a vector $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k) \in \mathbb{R}^{nk}$, we inductively construct k subsets $A^1, \ldots, A^k$ 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 $j \in J$. This means that, for every $i \in I$, the data point a^i belongs to the cluster A^j if and only if the distance $\|a^i - \bar{x}^j\|$ is the minimal one among the distances $\|a^i - \bar{x}^q\|$, $q \in J$, and a^i does not belong to any cluster A^p with $1 \le p \le j-1$. We will call this family $\{A^1, \ldots, A^k\}$ *the natural clustering associated with* $\bar{x}$.

Definition 3.1 Let $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k) \in \mathbb{R}^{nk}$. 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 $\bar{x}^j$.

Clearly, the cluster A^j in (3.3) can be represented as follows:

$$A^j = A[\bar{x}^j] \setminus \Big(\bigcup_{p=1}^{j-1} A^p \Big).$$

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)*.*

Proof. First, suppose that $(\bar{x}, \bar{\alpha})$ is a solution of the optimization problem (3.1). As $\psi(\bar{x}, \bar{\alpha}) \le \psi(\bar{x}, \alpha)$ for every $\alpha = (\alpha_{ij}) \in \mathbb{R}^{m \times k}$ with $\alpha_{ij} \in \{0, 1\}$, $\sum_{j=1}^{k} \alpha_{ij} = 1$ for all $i \in I$ and $j \in J$, one must have

$$\sum_{j=1}^{k} \bar{\alpha}_{ij} \|a^i - \bar{x}^j\|^2 = \min_{j \in J} \|a^i - \bar{x}^j\|^2 \quad (\forall i \in I).$$

Hence, $\psi(\bar{x}, \bar{\alpha}) = f(\bar{x})$. If $\bar{x}$ is not a solution of (3.2), then one can find some $\tilde{x} = (\tilde{x}^1, \ldots, \tilde{x}^k) \in \mathbb{R}^{nk}$ such that $f(\tilde{x}) < f(\bar{x})$. Let $\{A^1, \ldots, A^k\}$ be the natural clustering associated with $\tilde{x}$. For any $i \in I$ and $j \in J$, set $\tilde{\alpha}_{ij} = 1$ if $a^i \in A^j$ and $\tilde{\alpha}_{ij} = 0$ if $a^i \notin A^j$. Let $\tilde{\alpha} = (\tilde{\alpha}_{ij}) \in \mathbb{R}^{m \times k}$. From the definition of

natural clustering and the choice of $\tilde{\alpha}$ it follows that $\psi(\tilde{x}, \tilde{\alpha}) = f(\tilde{x})$. Then, we have

$$\psi(\bar{x}, \bar{\alpha}) = f(\bar{x}) > f(\tilde{x}) = \psi(\tilde{x}, \tilde{\alpha}),$$

contrary to the fact that $(\bar{x}, \bar{\alpha})$ is a solution of (3.1).

Now, suppose that $\bar{x}$ is a solution of (3.2). Let $\{A^1, \ldots, A^k\}$ be the natural clustering associated with $\bar{x}$. Put $\bar{\alpha} = (\bar{\alpha}_{ij})$, where $\bar{\alpha}_{ij} = 1$ if $a^i \in A^j$ and $\bar{\alpha}_{ij} = 0$ if $a^i \notin A^j$. It is easy to see that $\psi(\bar{x}, \bar{\alpha}) = f(\bar{x})$. If there is a feasible point (x, α) of (3.1) such that $\psi(x, \alpha) < \psi(\bar{x}, \bar{\alpha})$ then, by considering the natural clustering $\{\tilde{A}^1, \ldots, \tilde{A}^k\}$ associated with x and letting $\tilde{\alpha} = (\tilde{\alpha}_{ij})$ with $\tilde{\alpha}_{ij} = 1$ if $a^i \in \tilde{A}^j$ and $\tilde{\alpha}_{ij} = 0$ if $a^i \notin \tilde{A}^j$, we have $f(x) = \psi(x, \tilde{\alpha}) \leq \psi(x, \alpha)$. Then we get

$$f(x) \leq \psi(x, \alpha) < \psi(\bar{x}, \bar{\alpha}) = f(\bar{x}),$$

contrary to the global optimality of $\bar{x}$ for (3.2). One has thus proved that $(\bar{x}, \bar{\alpha})$ is a solution of (3.1). □

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

Proof. Indeed, if there is some $j_0 \in J$ with $A^{j_0} = \emptyset$, then the assumption $k \leq m$ implies the existence of an index $j_1 \in J$ such that A^{j_1} contains at least two points. Since the elements of A^{j_1} are pairwise distinct, one could find $a^{i_1} \in A^{j_1}$ with $a^{i_1} \neq \bar{x}^{j_1}$. Setting $\tilde{x}^j = \bar{x}^j$ for $j \in J \setminus \{j_0\}$ and $\tilde{x}^{j_0} = a^{i_1}$, one can easily show that

$$f(\tilde{x}) - f(\bar{x}) \leq -\frac{1}{m}\|a^{i_1} - \bar{x}^{j_1}\|^2 < 0.$$

This is impossible because $\bar{x}$ is a global solution of (3.2). □

Remark 3.1 In practical measures, some data points can coincide. Naturally, if $a^{i_1} = a^{i_2}$, $i_1 \neq i_2$, then a^{i_1} and a^{i_2} must belong to the same cluster. Procedure (3.3) guarantees the fulfillment of this natural requirement. By grouping identical data points and choosing from each group a unique representative, we obtain a new data set having pairwise distinct data points. Thus, *there is no loss of generality in assuming that $a^1, ..., a^m$ are pairwise distinct points.*

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 $\bar{x}=(\bar{x}^1,\ldots,\bar{x}^k)\in\mathbb{R}^{nk}$ *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*.*

Proof. a) *Solution existence:* By the second assertion of Proposition 3.1, it suffices to show that (3.2) has a solution. Since the minimum of finitely many continuous functions is a continuous function, the objective function of (3.2) is continuous on $\mathbb{R}^{nk}$. If $k=1$, then the formula for f can be rewritten as $f(x^1)=\dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\|a^i-x^1\|^2$. This smooth, strongly convex function attains its unique global minimum on $\mathbb{R}^n$ at the point $\bar{x}^1=a^0$, where

$$a^0:=\frac{1}{m}\sum_{i\in I}a^i \tag{3.5}$$

is the *barycenter* of the data set A (see, e.g., [50, pp. 24–25] for more details). To prove the solution existence of (3.2) for any $k\geq 2$, put $\rho=\max\limits_{i\in I}\|a^i-a^0\|$, where a^0 is defined by (3.5). Denote by $\bar{B}(a^0,2\rho)$ the closed ball in $\mathbb{R}^n$ centered at a^0 with radius 2ρ, and consider the optimization problem

$$\min\Big\{f(x)\ \mid\ x=(x^1,\ldots,x^k)\in\mathbb{R}^{nk},\ x^j\in\bar{B}(a^0,2\rho),\ \forall j\in J\Big\}. \tag{3.6}$$

By the Weierstrass theorem, (3.6) has a solution $\bar{x}=(\bar{x}^1,\ldots,\bar{x}^k)$ with $\bar{x}^j$ satisfying the inequality $\|\bar{x}^j-a^0\|\leq 2\rho$ for all $j\in J$. Take an arbitrary point $x=(x^1,\ldots,x^k)\in\mathbb{R}^{nk}$ and notice by the choice of $\bar{x}$ that $f(\bar{x})\leq f(x)$ if $\|x^j-a^0\|\leq 2\rho$ for all $j\in J$. If there exists at least one index $j\in J$ with $\|x^j-a^0\|>2\rho$, then denote the set of such indexes by J_1 and define a vector $\tilde{x}=(\tilde{x}^1,\ldots,\tilde{x}^k)\in\mathbb{R}^{nk}$ by putting $\tilde{x}^j=x^j$ for every $j\in J\setminus J_1$, and $\tilde{x}^j=a^0$ for all $j\in J_1$. For any $i\in I$, it is clear that $\|a^i-\tilde{x}^j\|=\|a^i-a^0\|\leq\rho<\|a^i-x^j\|$ for every $j\in J_1$, and $\|a^i-\tilde{x}^j\|=\|a^i-x^j\|$ for every $j\in J\setminus J_1$. So, we have $f(\tilde{x})\leq f(x)$. As $f(\bar{x})\leq f(\tilde{x})$, this yields $f(\bar{x})\leq f(x)$. We have thus proved that $\bar{x}$ is a solution of (3.2).

b) *Finiteness of the solution sets and formulae for the solution components:* Suppose that $a^1,\ldots,a^m$ are pairwise distinct points, $\bar{x}=(\bar{x}^1,\ldots,\bar{x}^k)\in\mathbb{R}^{nk}$ is a global solution of (3.2), and $\{A^1,\ldots,A^k\}$ is the natural clustering associated

with $\bar{x}$. By Proposition 3.2, $A^j \neq \emptyset$ for all $j \in J$. Since

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

and $A^j \neq \emptyset$ for every $j \in J$, we see that $|I(j)| \geq 1$ for every $j \in J$. This implies that right-hand-side of (3.4) is well defined for each $j \in J$. To justify that formula, we can argue as follows. Fix any $j \in J$. Since

$$\|a^i - \bar{x}^j\| > \min_{q \in J} \|a^i - \bar{x}^q\| \quad \forall i \notin I(j),$$

there exists $\varepsilon > 0$ such that

$$\|a^i - x^j\| > \min_{q \in J} \|a^i - \bar{x}^q\| \quad \forall i \notin I(j) \tag{3.7}$$

for any $x^j \in \bar{B}(\bar{x}^j, \varepsilon)$. For each $x^j \in \bar{B}(\bar{x}^j, \varepsilon)$, put $\tilde{x} = (\tilde{x}^1, \ldots, \tilde{x}^k)$ with $\tilde{x}^q := \bar{x}^q$ for every $q \in J \setminus \{j\}$ and $\tilde{x}^j := x^j$. From the inequality $f(\bar{x}) \leq f(\tilde{x})$ and the validity of (3.7) we can deduce that

$$\begin{aligned}
f(\bar{x}) &= \frac{1}{m} \sum_{i=1}^{m} \Big(\min_{q \in J} \|a^i - \bar{x}^q\|^2 \Big) \\
&= \frac{1}{m} \Big[\sum_{i \in I(j)} \|a^i - \bar{x}^j\|^2 + \sum_{i \in I \setminus I(j)} \Big(\min_{q \in J} \|a^i - \bar{x}^q\|^2 \Big) \Big] \\
&\leq f(\tilde{x}) \\
&= \frac{1}{m} \Big[\sum_{i \in I(j)} \Big(\min_{q \in J} \|a^i - \tilde{x}^q\|^2 \Big) + \sum_{i \in I \setminus I(j)} \Big(\min_{q \in J} \|a^i - \tilde{x}^q\|^2 \Big) \Big] \\
&= \frac{1}{m} \Big[\sum_{i \in I(j)} \Big(\min_{q \in J} \|a^i - \tilde{x}^q\|^2 \Big) + \sum_{i \in I \setminus I(j)} \Big(\min_{q \in J} \|a^i - \bar{x}^q\|^2 \Big) \Big] \\
&\leq \frac{1}{m} \Big[\sum_{i \in I(j)} \|a^i - x^j\|^2 + \sum_{i \in I \setminus I(j)} \Big(\min_{q \in J} \|a^i - \bar{x}^q\|^2 \Big) \Big].
\end{aligned} \tag{3.8}$$

Consider the function $\varphi(x^j) := \dfrac{1}{m} \displaystyle\sum_{i \in I(j)} \|a^i - x^j\|^2$, $x^j \in \mathbb{R}^n$. Comparing the expression on the second line of (3.8) with the one on the sixth line yields $\varphi(\bar{x}^j) \leq \varphi(x^j)$ for every $x^j \in \bar{B}(\bar{x}^j, \varepsilon)$. Hence φ attains its local minimum at $\bar{x}^j$. By the Fermat Rule we have $\nabla \varphi(\bar{x}^j) = 0$, which gives $\displaystyle\sum_{i \in I(j)} (a^i - \bar{x}^j) = 0$. This equality implies (3.4). Since there are only finitely many nonempty subsets $\Omega \subset I$, the set $\mathcal{B}$ of vectors b_Ω defined by formula $b_\Omega = \dfrac{1}{|\Omega|} \displaystyle\sum_{i \in \Omega} a^i$ is finite. (Note that b_Ω is the barycenter of the subsystem $\{a^i \in A \mid i \in \Omega\}$ of A.) According to (3.4), each component of a global

solution $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k)$ of (3.2) must belongs to $\mathcal{B}$, we can assert that the solution set of (3.2) is finite, provided that $a^1, ..., a^m$ are pairwise distinct points. By Proposition 3.1, if $(\bar{x}, \bar{\alpha})$ is a solution of (3.1), then $\bar{x}$ is a solution of (3.2). Since $\bar{\alpha} = (\bar{\alpha}_{ij}) \in \mathbb{R}^{m\times k}$ must satisfy the conditions $\bar{\alpha}_{ij} \in \{0, 1\}$ and $\sum_{j=1}^{k} \bar{\alpha}_{ij} = 1$ for all $i \in I$, $j \in J$, it follows that the solution set of (3.1) is also finite. □

Proposition 3.3 *If $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k) \in \mathbb{R}^{nk}$ 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 $j_2 \neq j_1$.*

Proof. On the contrary, suppose that there are distinct indexes $j_1, j_2 \in J$ satisfying $\bar{x}^{j_1} = \bar{x}^{j_2}$. As $k \leq m$, one has $k-1 < n$. So, there must exist $j_0 \in J$ such that $|A[\bar{x}^{j_0}]| \geq 2$. Therefore, one can find a data point $a^{i_0} \in A[\bar{x}^{j_0}]$ with $a^{i_0} \neq \bar{x}^{j_0}$. Setting $\tilde{x} = (\tilde{x}^1, \ldots, \tilde{x}^k)$ with $\tilde{x}^j = \bar{x}^j$ for every $j \in J \setminus \{j_2\}$ and $\tilde{x}^{j_2} = a^{i_0}$. The construction of $\tilde{x}$ yields

$$f(\tilde{x}) - f(\bar{x}) \leq -\frac{1}{m}\|a^{i_0} - \bar{x}^{j_0}\|^2 < 0,$$

which is impossible because $\bar{x}$ is a global solution of (3.2). □

Remark 3.2 If the points $a^1, ..., a^m$ are not pairwise distinct, then the conclusions of Theorem 3.1 do not hold in general. Indeed, let $A = \{a^1, a^2\} \subset \mathbb{R}^2$ with $a^1 = a^2$. For $k := 2$, let $\bar{x} = (\bar{x}^1, \bar{x}^2)$ with $\bar{x}^1 = a^1$ and $\bar{x}^2 \in \mathbb{R}^2$ being chosen arbitrarily. Since $f(\bar{x}) = 0$, we can conclude that $\bar{x}$ is a global solution of (3.2). So, the problem has an unbounded solution set. Similarly, for a data set $A = \{a^1, \ldots, a^4\} \subset \mathbb{R}^2$ with $a^1 = a^2$, $a^3 = a^4$, and $a^2 \neq a^3$. For $k := 3$, let $\bar{x} = (\bar{x}^1, \bar{x}^2, \bar{x}^3)$ with $\bar{x}^1 = a^1$, $\bar{x}^2 = a^3$, and $\bar{x}^3 \in \mathbb{R}^2$ being chosen arbitrarily. By the equality $f(\bar{x}) = 0$ we can assert that $\bar{x}$ is a global solution of (3.2). This shows that the solution set of (3.2) is unbounded. Notice also that, if $\bar{x}^3 \notin \{\bar{x}^1, \bar{x}^2\}$, then formula (3.4) cannot be applied to $\bar{x}^3$, because the index set $I(3) = \{i \in I \mid a^i \in A[\bar{x}^3]\} = \left\{i \in I \mid \|a^i - \bar{x}^3\| = \min_{q\in J} \|a^i - \bar{x}^q\|\right\}$ is empty.

Formula (3.4) is effective for computing certain components of any given *local solution* of (3.2). The precise statement of this result is as follows.

Theorem 3.2 *If $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k) \in \mathbb{R}^{nk}$ 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.*

Proof. It suffices to re-apply the arguments described in the second part of the proof of Theorem 3.1, noting that $f(\bar{x}) \le f(\tilde{x})$ if x^j (the j-th component of $\tilde{x}$) is taken from $\bar{B}(\bar{x}^j, \varepsilon')$ with $\varepsilon' \in (0, \varepsilon)$ being small enough. □

As in the proof of Theorem 3.1, if $\Omega = \{a^{i_1}, \ldots, a^{i_r}\} \subset A$ is a nonempty subset, then we put $b_\Omega = \dfrac{1}{r}\sum_{l=1}^{r} a^{i_l}$. Recall that the set of such points b_Ω has been denoted by $\mathcal{B}$.

Remark 3.3 Theorem 3.1 shows that if the points $a^1, ..., a^m$ are pairwise distinct, then every component of a global solution must belong to $\mathcal{B}$. It is clear that $\mathcal{B} \subset \mathrm{co}A$, where co$A$ abbreviates the *convex hull* of A. Looking back to the proof of Theorem 3.1, we see that the set A lies in the ball $\bar{B}(a^0, \rho)$. Hence $\mathcal{B} \subset \mathrm{co}A \subset \bar{B}(a^0, \rho)$. It follows that *the global solutions of* (3.2) *are contained in the set*

$$\Big\{x = (x^1, \ldots, x^k) \in \mathbb{R}^{nk} \;\mid\; x^j \in \bar{B}(a^0, \rho),\ \forall j \in J\Big\},$$

provided the points $a^1, ..., a^m$ *are pairwise distinct.* Similarly, Theorem 3.2 assures that each attractive component of a local solution of (3.2) belongs to $\mathcal{B}$, where $\mathcal{B} \subset \mathrm{co}A \subset \bar{B}(a^0, \rho)$.

Remark 3.4 If $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k) \in \mathbb{R}^{nk}$ is a global solution (resp., a local solution) of (3.2) then, for any permutation σ of J, the vector

$$\bar{x}^\sigma := (\bar{x}^{\sigma(1)}, \ldots, \bar{x}^{\sigma(k)})$$

is also a global solution (resp., a local solution) of (3.2). This observation follows easily from the fact that $f(x) = f(x^\sigma)$, where $x = (x^1, \ldots, x^k) \in \mathbb{R}^{nk}$ and $x^\sigma := (x^{\sigma(1)}, \ldots, x^{\sigma(k)})$.

To understand the importance of the above results and those to be established in the next two sections, let us recall the k-means clustering algorithm and consider an illustrative example.

3.3 The k-means Algorithm

Despite its ineffectiveness, the k-means clustering algorithm (see, e.g., [1, pp. 89–90], [39], [43, pp. 263–266], and [66]) is one of the most popular solution methods for (3.2). The convergence of this algorithms was proven in [86].

One starts with selecting k points $x^1, \ldots, x^k$ in $\mathbb{R}^n$ as the initial centroids. Then one inductively constructs k subsets $A^1, \ldots, A^k$ of the data set A by putting $A^0 = \emptyset$ and using the rule (3.3), where x^j plays the role of $\bar{x}^j$ for all $j \in J$. This means that $\{A^1, \ldots, A^k\}$ is the natural clustering associated with $x = (x^1, \ldots, x^k)$. Once the clusters are formed, for each $j \in J$, if $A^j \neq \emptyset$ then the centroid x^j is updated by the rule

$$x^j \quad \leftarrow \quad \widetilde{x}^j := \frac{1}{|I(A^j)|} \sum_{i \in I(A^j)} a^i \tag{3.9}$$

with $I(A^j) := \{i \in I \mid a^i \in A^j\}$; and x^j does not change otherwise. The algorithm iteratively repeats the procedure until the centroid system $\{x^1, \ldots, x^k\}$ is stable, i.e., $\widetilde{x}^j = x^j$ for all $j \in J$ with $A^j \neq \emptyset$. The computation procedure is described as follows.

Input: *The data set $A = \{a^1, ..., a^m\}$ and a constant $\varepsilon \geq 0$ (tolerance).*
Output: *The set of k centroids $\{x^1, ..., x^k\}$.*
Step 1. Select initial centroids $x^j \in \mathbb{R}^n$ for all $j \in J$.
Step 2. Compute $\alpha_i = \min\{\|a^i - x^j\| \mid j \in J\}$ for all $i \in I$.
Step 3. Form the clusters $A^1, \ldots, A^k$:
- Find the attraction sets

$$A[x^j] = \{a^i \in A \mid \|a^i - x^j\| = \alpha_i\} \qquad (j \in J);$$

- Set $A^1 = A[x^1]$ and

$$A^j = A[x^j] \setminus \Big(\bigcup_{p=1}^{j-1} A^p\Big) \quad (j = 2, \ldots, k). \tag{3.10}$$

Step 4. Update the centroids x^j satisfying $A^j \neq \emptyset$ by the rule (3.9), keeping other centroids unchanged.
Step 5. Check the convergence condition: If $\|\widetilde{x}^j - x^j\| \leq \varepsilon$ for all $j \in J$ with $A^j \neq \emptyset$ then stop, else go to Step 2.

The following example is designed to show how the algorithm is performed in practice.

Example 3.1 Choose $m = 3$, $n = 2$, and $k = 2$. Let $A = \{a^1, a^2, a^3\}$, where $a^1 = (0, 0)$, $a^2 = (1, 0)$, $a^3 = (0, 1)$. Apply the k-means algorithm to solve the problem (3.2) with the tolerance $\varepsilon = 0$.

(a) With the starting centroids $x^1 = a^1$, $x^2 = a^2$, one obtains the clusters $A^1 = A[x^1] = \{a^1, a^3\}$ and $A^2 = A[x^2] = \{a^2\}$. The updated centroids are $x^1 = (0, \frac{1}{2})$, $x^2 = a^2$. Then, the new clusters A^1 and A^2 coincide with the old ones. Thus, $\|\widetilde{x}^j - x^j\| = 0$ for all $j \in J$ with $A^j \neq \emptyset$. So, the computation terminates. For $x^1 = (0, \frac{1}{2})$, $x^2 = a^2$, one has $f(x) = \frac{1}{6}$.

(b) Starting with the points $x^1 = (\frac{1}{4}, \frac{3}{4})$ and $x^2 = (2, 3)$, one gets the clusters $A^1 = A[x^1] = \{a^1, a^2, a^3\}$ and $A^2 = A[x^2] = \emptyset$. The algorithm gives the centroid system $x^1 = (\frac{1}{3}, \frac{1}{3})$, $x^2 = (2, 3)$, and $f(x) = \frac{1}{3}$.

(c) Starting with $x^1 = (0, 1)$ and $x^2 = (0, 0)$, by the algorithm we are led to $A^1 = A[x^1] = \{a^3\}$, $A^2 = A[x^2] = \{a^1, a^2\}$, $x^1 = (0, 1)$, and $x^2 = (\frac{1}{2}, 0)$. The corresponding value of objective function is $f(x) = \frac{1}{6}$.

(d) Starting with $x^1 = (0, 0)$ and $x^2 = (\frac{1}{2}, \frac{1}{2})$, by the algorithm one gets the results $A^1 = A[x^1] = \{a^1\}$, $A^2 = A[x^2] = \{a^2, a^3\}$, $x^1 = (0, 0)$, and $x^2 = (\frac{1}{2}, \frac{1}{2})$. The corresponding value of objective function is $f(x) = \frac{4}{9}$.

(e) With $x^1 = (\frac{1}{3}, \frac{1}{3})$ and $x^2 = (1 + \frac{\sqrt{5}}{3}, 0)$ as the initial centroids, one obtains the results $A^1 = A[x^1] = \{a^1, a^2, a^3\}$, $A^2 = A[x^2] = \emptyset$, $x^1 = (\frac{1}{3}, \frac{1}{3})$, $x^2 = (1 + \frac{\sqrt{5}}{3}, 0)$, and $f(x) = \frac{4}{9}$.

Based on the existing knowledge on the MSSC problem and the k-means clustering algorithm, one cannot know whether the five centroid systems obtained in the items (a)–(e) *of Example 3.1 contain a global optimal solution of the clustering problem, or not. Even if one knows that the centroid systems obtained in* (a) *and* (c) *are global optimal solutions, one still cannot say definitely whether the centroid systems obtained in the items* (b), (d), (e) *are local optimal solutions of* (3.2)*, or not.*

The theoretical results in Section 3.2 and the two forthcoming ones allow us to clarify the following issues related to the MSSC problem in Example 3.1:

- The structure of the global solution set (see Example 3.2 below);

- The structure of the local solution set (see Example 3.3);

- The performance of the k-means algorithm (see Example 3.4).

In particular, it will be shown that the centroid systems in (a) and (c) are global optimal solutions, the centroid systems in (b) and (d) are local-nonglobal optimal solutions, while the centroid system in (e) is not a local solution (despite the fact that the centroid systems generated by the k-means algorithm converge to it, and the value of the objective function at it equals

to the value given by the centroid system in (d)).

3.4 Characterizations of the Local Solutions

In order to study the local solution set of (3.2) in more details, we will follow Ordin and Bagirov [71] to consider the problem in light of a well-known optimality condition in DC programming. For every $x = (x^1, ..., x^k) \in \mathbb{R}^{nk}$, we have

$$\begin{aligned} f(x) &= \frac{1}{m}\sum_{i\in I}\Big(\min_{j\in J}\|a^i - x^j\|^2\Big) \\ &= \frac{1}{m}\sum_{i\in I}\Big[\Big(\sum_{j\in J}\|a^i - x^j\|^2\Big) - \Big(\max_{j\in J}\sum_{q\in J\setminus\{j\}}\|a^i - x^q\|^2\Big)\Big]. \end{aligned} \tag{3.11}$$

Hence, the objective function f of (3.2) can be expressed [71, p. 345] as the difference of two convex functions

$$f(x) = f^1(x) - f^2(x), \tag{3.12}$$

where

$$f^1(x) := \frac{1}{m}\sum_{i\in I}\Big(\sum_{j\in J}\|a^i - x^j\|^2\Big) \tag{3.13}$$

and

$$f^2(x) := \frac{1}{m}\sum_{i\in I}\Big(\max_{j\in J}\sum_{q\in J\setminus\{j\}}\|a^i - x^q\|^2\Big). \tag{3.14}$$

It is clear that f^1 is a convex linear-quadratic function. In particular, it is differentiable. As the sum of finitely many nonsmooth convex functions, f^2 is a nonsmooth convex function, which is defined on the whole space $\mathbb{R}^{nk}$. The subdifferentials of $f^1(x)$ and $f^2(x)$ can be computed as follows. First, one has

$$\begin{aligned} \partial f^1(x) = \{\nabla f^1(x)\} &= \Big\{\frac{2}{m}\sum_{i\in I}\left(x^1 - a^i, \ldots, x^k - a^i\right)\Big\} \\ &= \{2(x^1 - a^0, \ldots, x^k - a^0)\} \end{aligned}$$

where, as before, $a^0 = b_A$ is the barycenter of the system $\{a^1, \ldots, a^m\}$. Set

$$\varphi_i(x) = \max_{j\in J} h_{i,j}(x) \tag{3.15}$$

with $h_{i,j}(x) := \sum_{q\in J\setminus\{j\}}\|a^i - x^q\|^2$ and

$$J_i(x) = \{j \in J \mid h_{i,j}(x) = \varphi_i(x)\}. \tag{3.16}$$

Proposition 3.4 *One has*

$$J_i(x) = \{j \in J \mid a^i \in A[x^j]\}. \tag{3.17}$$

Proof. From the formula of $h_{i,j}(x)$ it follows that

$$h_{i,j}(x) = \Big(\sum_{q\in J} \|a^i - x^q\|^2\Big) - \|a^i - x^j\|^2.$$

Therefore, by (3.15) we have

$$\begin{aligned}\varphi_i(x) &= \max_{j\in J}\Big[\Big(\sum_{q\in J} \|a^i - x^q\|^2\Big) - \|a^i - x^j\|^2\Big]\\ &= \Big(\sum_{q\in J} \|a^i - x^q\|^2\Big) + \max_{j\in J}\Big(-\|a^i - x^j\|^2\Big)\\ &= \Big(\sum_{q\in J} \|a^i - x^q\|^2\Big) - \min_{j\in J} \|a^i - x^j\|^2.\end{aligned}$$

Thus, the maximum in (3.15) is attained when the minimum $\min\limits_{j\in J} \|a^i - x^j\|^2$ is achieved. So, by (3.16),

$$J_i(x) = \Big\{j \in J \mid \|a^i - x^j\| = \min_{q\in J} \|a^i - x^q\|\Big\}.$$

This implies (3.17). □

Invoking the subdifferential formula for the maximum function (see [20, Proposition 2.3.12] and note that the Clarke generalized gradient coincides with the subdifferential of convex analysis if the functions in question are convex), we have

$$\partial\varphi_i(x) = \operatorname{co}\{\nabla h_{i,j}(x) \,|\, j \in J_i(x)\} = \operatorname{co}\{2\,(\widetilde{x}^j - \widetilde{a}^{i,j}) \mid j \in J_i(x)\}, \tag{3.18}$$

where

$$\widetilde{x}^j = (x^1, \ldots, x^{j-1}, 0_{\mathbb{R}^n}, x^{j+1}, \ldots, x^k) \tag{3.19}$$

and

$$\widetilde{a}^{i,j} = \Big(a^i, \ldots, a^i, \underbrace{0_{\mathbb{R}^n}}_{j-\text{th position}}, a^i, \ldots, a^i\Big). \tag{3.20}$$

By the Moreau-Rockafellar theorem [84, Theorem 23.8], one has

$$\partial f^2(x) = \frac{1}{m}\sum_{i\in I} \partial\varphi_i(x) \tag{3.21}$$

with $\partial\varphi_i(x)$ being computed by (4.50).

Now, suppose $x = (x^1, ..., x^k) \in \mathbb{R}^{nk}$ is a local solution of (3.2). By the necessary optimality condition in DC programming (see, e.g., [31] and [77]), which can be considered as a consequence of the optimality condition obtained by Dem'yanov et al. in quasidifferential calculus (see, e.g., [25, Theorem 3.1] and [26, Theorem 5.1]), we have

$$\partial f^2(x) \subset \partial f^1(x). \tag{3.22}$$

Since $\partial f^1(x)$ is a singleton, $\partial f^2(x)$ must be a singleton too. This happens if and only if $\partial \varphi_i(x)$ is a singleton for every $i \in I$. By (4.50), if $|J_i(x)| = 1$, then $|\partial \varphi_i(x)| = 1$. In the case where $|J_i(x)| > 1$, we can select two elements j_1 and j_2 from $J_i(x)$, $j_1 < j_2$. As $\partial \varphi_i(x)$ is a singleton, by (4.50) one must have $\widetilde{x}^{j_1} - \widetilde{a}^{i,j_1} = \widetilde{x}^{j_2} - \widetilde{a}^{i,j_2}$. Using (3.19) and (3.20), one sees that the latter occurs if and only if $x^{j_1} = x^{j_2} = a^i$. To proceed furthermore, we need to introduce 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*.

Remark 3.5 Proposition 3.3 shows that every global solution of (3.2) is a nontrivial local solution.

The following fundamental facts have the origin in [71, pp. 346]. Here, a more precise and complete formulation is presented. In accordance with (3.17), the first assertion of the next theorem means that if x is a nontrivial local solution, then for each data point $a^i \in A$ there is a unique component x^j of x such that $a^i \in A[x^j]$.

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*

$$x^j = \frac{1}{|I(j)|} \sum_{i \in I(j)} a^i, \tag{3.23}$$

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

$$x^j \notin \mathcal{A}[x], \tag{3.24}$$

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 $p \in I(q)$.

Proof. Suppose $x = (x^1, ..., x^k)$ is a nontrivial local solution of (3.2). Given any $i \in I$, we must have $|J_i(x)| = 1$. Indeed, if $|J_i(x)| > 1$ then, by the analysis given before the formulation of the theorem, there exist indexes j_1 and j_2 from $J_i(x)$ such that $x^{j_1} = x^{j_2} = a^i$. This contradicts the nontriviality of the local solution x. Let $J_i(x) = \{j(i)\}$ for $i \in I$, i.e., $j(i) \in J$ is the unique element of $J_i(x)$.

For each $i \in I$, observe by (3.15) that

$$h_{i,j}(x) < h_{i,j(i)}(x) = \varphi_i(x) \;\; \forall j \in J \setminus \{j(i)\}.$$

Hence, by the continuity of the functions $h_{i,j}(x)$, there exists an open neighborhood U_i of x such that

$$h_{i,j}(y) < h_{i,j(i)}(y) \;\; \forall j \in J \setminus \{j(i)\}, \; \forall y \in U_i.$$

It follows that

$$\varphi_i(y) = h_{i,j(i)}(y) \;\; \forall y \in U_i. \tag{3.25}$$

So, $\varphi_i(.)$ is continuously differentiable on U_i. Put $U = \bigcap_{i \in I} U_i$. From (3.14) and (3.25) one can deduce that

$$f^2(y) = \frac{1}{m} \sum_{i \in I} \varphi_i(y) = \frac{1}{m} \sum_{i \in I} h_{i,j(i)}(y) \quad \forall y \in U.$$

Therefore, $f^2(y)$ is continuously differentiable function on U. Moreover, the formulas (4.50)–(3.20) yield

$$\nabla f^2(y) = \frac{2}{m} \sum_{i \in I} (\widetilde{y}^{j(i)} - \widetilde{a}^{i,j(i)}) \quad \forall y \in U, \tag{3.26}$$

where

$$\widetilde{y}^{j(i)} = \big(y^1, ..., y^{j(i)-1}, 0_{\mathbb{R}^n}, y^{j(i)+1}, ..., y^k\big)$$

and

$$\widetilde{a}^{i,j(i)} = \Big(a^i, \ldots, a^i, \underbrace{0_{\mathbb{R}^n}}_{j(i)-\text{th position}}, a^i, \ldots, a^i\Big).$$

Substituting $y = x$ into (3.26) and combining the result with (3.22), we obtain

$$\sum_{i \in I} (\widetilde{x}^{j(i)} - \widetilde{a}^{i,j(i)}) = m(x^1 - a^0, ..., x^k - a^0). \tag{3.27}$$

Now, fix an index $j \in J$ with $A[x^j] \neq \emptyset$ and transform the left-hand side of (3.27) as follows:

$$\sum_{i\in I}(\widetilde{x}^{j(i)} - \widetilde{a}^{i,j(i)}) = \sum_{i\in I,\, j(i)=j}(\widetilde{x}^{j(i)} - \widetilde{a}^{i,j(i)}) + \sum_{i\in I,\, j(i)\neq j}(\widetilde{x}^{j(i)} - \widetilde{a}^{i,j(i)})$$
$$= \sum_{i\in I,\, j(i)=j}(\widetilde{x}^{j(i)} - \widetilde{a}^{i,j(i)}) + \sum_{i\notin I(j)}(\widetilde{x}^{j(i)} - \widetilde{a}^{i,j(i)}).$$

Clearly, if $j(i) = j$, then the j-th component of the vector $\widetilde{x}^{j(i)} - \widetilde{a}^{i,j(i)}$, that belongs to $\mathbb{R}^{nk}$, is $0_{\mathbb{R}^n}$. If $j(i) \neq j$, then the j-th component of the vector $\widetilde{x}^{j(i)} - \widetilde{a}^{i,j(i)}$ is $x^j - a^i$. Consequently, (3.27) gives us

$$\sum_{i\notin I(j)}(x^j - a^i) = m(x^j - a^0).$$

Since $ma^0 = a^1 + \cdots + a^m$, this yields $\sum_{i\in I(j)} a^i = |I(j)|x^j$. Thus, formula (3.23) is valid for any $j \in J$ satisfying $A[x^j] \neq \emptyset$.

For any $j \in J$ with $A[x^j] = \emptyset$, one has (3.24). Indeed, suppose to the contrary that there exits $j_0 \in J$ with $A[x^{j_0}] = \emptyset$ such that for some $p \in I$, $q \in J$, one has $p \in I(q)$ and $x^{j_0} \in \bar{B}(a^p, \|a^p - x^q\|)$. If $\|a^p - x^{j_0}\| = \|a^p - x^q\|$, then $J_p(x) \supset \{q, j_0\}$. This is impossible due to the first claim of the theorem. Now, if $\|a^p - x^{j_0}\| < \|a^p - x^q\|$, then $p \notin I(q)$. We have thus arrived at a contradiction.

The proof is complete. □

Roughly speaking, the necessary optimality condition given in the above theorem is a sufficient one. Therefore, in combination with Theorem 3.3, the next statement gives *a complete description* of the nontrivial local solutions of (3.2).

Theorem 3.4 (Sufficient conditions for nontrivial local optimality) *Suppose that a vector $x = (x^1, ..., x^k) \in \mathbb{R}^{nk}$ satisfies condition* **(C1)** *and $|J_i(x)| = 1$ for every $i \in I$. If* (3.23) *is valid for any $j \in J$ with $A[x^j] \neq \emptyset$ and* (3.24) *is fulfilled for any $j \in J$ with $A[x^j] = \emptyset$, then x is a nontrivial local solution of* (3.2).

Proof. Let $x = (x^1, ..., x^k) \in \mathbb{R}^{nk}$ be such that **(C1)** holds, $J_i(x) = \{j(i)\}$ for every $i \in I$, (3.23) is valid for any $j \in J$ with $A[x^j] \neq \emptyset$, and (3.24) is satisfied for any $j \in J$ with $A[x^j] = \emptyset$. Then, for all $i \in I$ and $j' \in J \setminus \{j(i)\}$,

one has

$$\|a^i - x^{j(i)}\| < \|a^i - x^{j'}\|.$$

So, there exist $\varepsilon > 0$, $q \in J$, such that

$$\|a^i - \widetilde{x}^{j(i)}\| < \|a^i - \widetilde{x}^{j'}\| \quad \forall i \in I,\ \forall j' \in J \setminus \{j(i)\}, \tag{3.28}$$

whenever vector $\widetilde{x} = (\widetilde{x}^1, ..., \widetilde{x}^k) \in \mathbb{R}^{nk}$ satisfies the condition $\|\widetilde{x}^q - x^q\| < \varepsilon$ for all $q \in J$. By (3.24) and by the compactness of $\mathcal{A}[x]$, reducing the positive number ε (if necessary) we have

$$\widetilde{x}^j \notin \mathcal{A}[\widetilde{x}] \tag{3.29}$$

whenever vector $\widetilde{x} = (\widetilde{x}^1, ..., \widetilde{x}^k) \in \mathbb{R}^{nk}$ satisfies the condition $\|\widetilde{x}^q - x^q\| < \varepsilon$ for all $q \in J$, where $\mathcal{A}[\widetilde{x}]$ is the union of the balls $\bar{B}(a^p, \|a^p - \widetilde{x}^q\|)$ with $p \in I$, $q \in J$ satisfying $p \in I(q) = \{i \in I \mid a^i \in A[x^q]\}$.

Fix any vector $\widetilde{x} = (\widetilde{x}^1, ..., \widetilde{x}^k) \in \mathbb{R}^{nk}$ with the property that $\|\widetilde{x}^q - x^q\| < \varepsilon$ for all $q \in J$. Then, by (3.28) and (3.29), $J_i(\widetilde{x}) = \{j(i)\}$. So,

$$\min_{j \in J} \|a^i - \widetilde{x}^j\|^2 = \|a^i - \widetilde{x}^{j(i)}\|^2.$$

Therefore, one has

$$\begin{aligned} f(\widetilde{x}) &= \frac{1}{m} \sum_{i \in I} \Big(\min_{j \in J} \|a^i - \widetilde{x}^j\|^2 \Big) \\ &= \frac{1}{m} \sum_{i \in I} \|a^i - \widetilde{x}^{j(i)}\|^2 \\ &= \frac{1}{m} \sum_{j \in J} \Big(\sum_{i \in I(j)} \|a^i - \widetilde{x}^{j(i)}\|^2 \Big) \\ &= \frac{1}{m} \sum_{j \in J} \Big(\sum_{i \in I(j)} \|a^i - \widetilde{x}^j\|^2 \Big) \\ &\geq \frac{1}{m} \sum_{j \in J} \Big(\sum_{i \in I(j)} \|a^i - x^j\|^2 \Big) = f(x), \end{aligned}$$

where the inequality is valid because (3.23) obviously yields

$$\sum_{i \in I(j)} \|a^i - x^j\|^2 \leq \sum_{i \in I(j)} \|a^i - \widetilde{x}^j\|^2$$

for every $j \in J$ such that the attraction set $A[x^j]$ of x^j is nonempty. (Note that x^j is the barycenter of $A[x^j]$.)

The local optimality of $x = (x^1, ..., x^k)$ has been proved. Hence, x is a nontrivial local solution of (3.2). □

Example 3.2 (A local solution need not be a global solution) Consider the clustering problem described in Example 3.1. Here, we have $I = \{1, 2, 3\}$ and $J = \{1, 2\}$. By Theorem 3.1, problem (3.2) has a global solution. Moreover, if $x = (x^1, x^2) \in \mathbb{R}^{2\times 2}$ is a global solution then, for every $j \in J$, the attraction set $A[x^j]$ is nonempty. Thanks to Remark 3.5, we know that x is a nontrivial local solution. So, by Theorem 3.3, the attraction sets $A[x^1]$ and $A[x^2]$ are disjoint. Moreover, the barycenter of each one of these sets can be computed by formula (3.23). Clearly, $A = A[x^1] \cup A[x^2]$. Since $A[x^j] \subset A = \{a^1, a^2, a^3\}$, allowing permutations of the components of each vector $x = (x^1, x^2) \in \mathbb{R}^{2\times 2}$ (see Remark 3.4), we can assert that the global solution set of our problem is contained in the set

$$\Big\{\bar{x} := \big((\frac{1}{2}, \frac{1}{2}), (0, 0)\big),\ \hat{x} := \big((0, \frac{1}{2}), (1, 0)\big),\ \widetilde{x} := \big((\frac{1}{2}, 0), (0, 1)\big)\Big\}. \quad (3.30)$$

Since $f(\bar{x}) = \frac{1}{3}$ and $f(\hat{x}) = f(\widetilde{x}) = \frac{1}{6}$, we infer that $\hat{x}$ and $\widetilde{x}$ are global solutions of our problem. Using Theorem 3.4, we can assert that $\bar{x}$ is a local solution. Thus, $\bar{x}$ is a local solution which does not belong to the global solution set, i.e., $\bar{x}$ is *a local-nonglobal solution* of our problem.

Example 3.3 (Complete description of the set of nontrivial local solutions) Again, consider the MSSC problem given in Example 3.1. Allowing permutations of the components of each vector in $\mathbb{R}^{2\times 2}$, by Theorems 3.3 and 3.4 we find that the set of nontrivial local solutions consists of the three vectors described in (3.30) and all the vectors of the form $x = (x^1, x^2) \in \mathbb{R}^{2\times 2}$, where $x^1 = (\frac{1}{3}, \frac{1}{3})$ and

$$x^2 \notin \bar{B}(a^1, \|a^1 - x^1\|) \cup \bar{B}(a^2, \|a^2 - x^1\|) \cup \bar{B}(a^3, \|a^3 - x^1\|).$$

This set of nontrivial local solutions is *unbounded* and *non-closed*.

Example 3.4 (Convergence analysis of the k-means algorithm) Consider once again the problem described in Example 3.1. By the results given in Example 3.3, the centroid systems in items (a), (b), (c) and (d) of Example 3.1 are local solutions. In addition, by Example 3.2, the centroid systems in the just mentioned items (a) and (c) are global solutions. Concerning the centroid system in item (e) of Example 3.1, remark that $x := \big((\frac{1}{3}, \frac{1}{3}), (1 + \frac{\sqrt{5}}{3}, 0)\big)$ is not a local solution by Theorem 3.3, because $a^2 \in A[x^1] \cap A[x^2]$, i.e., $J_2(x) = \{1, 2\}$ (see Figure 3.1). In general, with $x^1 = (\frac{1}{3}, \frac{1}{3})$ and $x^2 \in \mathbb{R}^{2\times 2}$ belonging to the boundary of the set

$$\bar{B}(a^1, \|a^1 - x^1\|) \cup \bar{B}(a^2, \|a^2 - x^1\|) \cup \bar{B}(a^3, \|a^3 - x^1\|),$$

$x := (x^1, x^2)$ is not a local solution of the MSSC problem under consideration. The above analysis shows that the k-means algorithm is very sensitive to the choice of starting centroids. The algorithm may give a global solution, a local-nonglobal solution, as well as a centroid system which is not a local solution. In other words, the quality of the obtained result greatly depends on the initial centroid system.

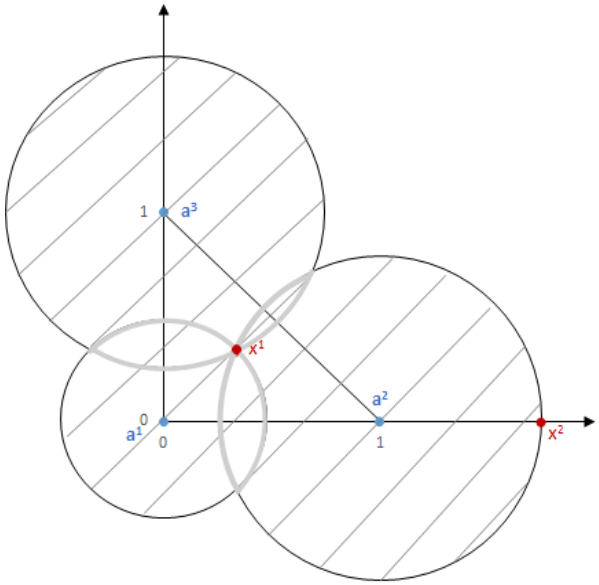

Figure 3.1: The centroids in item (e) of Example 3.1

3.5 Stability Properties

This section is devoted to establishing the local Lipschitz property of the optimal value function, the local upper Lipschitz property of the global solution map, and the local Lipschitz-like property of the local solution map of (3.2).

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 $a \in \mathbb{R}^{nm}$. Denoting by $v(a)$ the *optimal value* of (3.2), one has

$$v(a) = \min\{f(x) \mid x = (x^1, \ldots, x^k) \in \mathbb{R}^{nk}\}. \tag{3.31}$$

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

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

Let us abbreviate the *local solution set* of (3.2) to $F_1(a)$. Note that the inclusion $F(a) \subset F_1(a)$ is valid, and it may be strict.

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 $\bar{a} = (\bar{a}^1, ..., \bar{a}^m) \in \mathbb{R}^{nm}$ 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 $v : \mathbb{R}^{nm} \to \mathbb{R}$ 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 $\|a' - \bar{a}\| < \delta_0$.

Proof. Denote by Ω the set of all the partitions of I. Every element ω of Ω is a family $\{I_\omega(j) \mid j \in J\}$ of pairwise distinct, nonempty subsets of I with $\bigcup_{j\in J} I_\omega(j) = I$. We associate to each pair (ω, a), where $a = (a^1, ..., a^m) \in \mathbb{R}^{nm}$ and $\omega \in \Omega$, a vector $x_\omega(a) = (x^1_\omega(a), \ldots, x^k_\omega(a)) \in \mathbb{R}^{nk}$ with

$$x^j_\omega(a) = \frac{1}{|I_\omega(j)|} \sum_{i\in I_\omega(j)} a^i \tag{3.32}$$

for every $j \in J$. By Theorem 3.1, problem (3.2) has solutions and the number of the global solutions is finite, i.e., $F(\bar{a})$ is nonempty and finite. Moreover, for each $\bar{x} = (\bar{x}^1, ..., \bar{x}^k) \in F(\bar{a})$, one can find some $\omega \in \Omega$ satisfying $\bar{x}^j = x^j_\omega(\bar{a})$ for all $j \in J$. Let $\Omega_1 = \{\omega_1, \ldots, \omega_r\}$ be the set of the elements of Ω corresponding the global solutions. Then,

$$f(x_{\omega_1}(\bar{a}), \bar{a}) < f(x_\omega(\bar{a}), \bar{a}) \quad (\forall \omega \in \Omega \setminus \Omega_1), \tag{3.33}$$

where

$$f(x, a) = \frac{1}{m} \sum_{i\in I} \Big(\min_{j\in J} \|a^i - x^j\|^2 \Big). \tag{3.34}$$

For each pair $(i, j) \in I \times J$, the rule $(x, a) \mapsto \|a^i - x^j\|^2$ defines a polynomial function on $\mathbb{R}^{nk} \times \mathbb{R}^{nm}$. In particular, this function is locally Lipschitz on its domain. So, by [20, Prop. 2.3.6 and 2.3.12] we can assert that the function $f(x, a)$ in (3.34) is locally Lipschitz on $\mathbb{R}^{nk} \times \mathbb{R}^{nm}$.

Now, observe that for any $\omega \in \Omega$ and $j \in J$, the vector function $x^j_\omega(.)$ in (3.32), which maps $\mathbb{R}^{nm}$ to $\mathbb{R}^n$, is continuously differentiable. In particular, it is locally Lipschitz on $\mathbb{R}^{nm}$.

For every $\omega \in \Omega$, from the above observations we can deduce that the function $g_\omega(a) := f(x_\omega(a), a)$ is locally Lipschitz on $\mathbb{R}^{nm}$. Rewriting (3.33) as

$$g_{\omega_1}(\bar{a}) < g_\omega(\bar{a}) \quad (\forall \omega \in \Omega \setminus \Omega_1)$$

and using the continuity of the functions $g_\omega(.)$, we can find a number $\delta_0 > 0$ such that

$$g_{\omega_1}(a) < g_\omega(a) \quad (\forall \omega \in \Omega \setminus \Omega_1) \tag{3.35}$$

for all a satisfying $\|a - \bar{a}\| < \delta_0$. Since $\bar{a}^1, ..., \bar{a}^m$ are pairwise distinct, without loss of generality, we may assume that $a^1, ..., a^m$ are pairwise distinct for any $a = (a^1, ..., a^m)$ with $\|a - \bar{a}\| < \delta_0$.

Now, consider a vector $a = (a^1, ..., a^m)$ satisfying $\|a - \bar{a}\| < \delta_0$. By (3.35), $f(x_{\omega_1}(a), a) < f(x_\omega(a), a)$ for all $\omega \in \Omega \setminus \Omega_1$. Since $f(., a)$ is the objective function of (3.2), this implies that the set $\{x_\omega(a) \mid \omega \in \Omega \setminus \Omega_1\}$ does not contain any global solution of the problem. Thanks to Theorem 3.1, we know that the global solution set $F(a)$ of (3.2) is contained in the set

$$\{x_\omega(a) \mid \omega \in \Omega_1\}.$$

Hence,

$$F(a) \subset \{x_\omega(a) \mid \omega \in \Omega_1\} = \{x_{\omega_1}(a), \ldots, x_{\omega_r}(a)\}. \tag{3.36}$$

Since $F(a) \neq \emptyset$, by (3.36) one has

$$v(a) = \min\{f(x, a) \mid x \in F(a)\} = \min\{f(x_{\omega_\ell}(a), a) \mid \ell = 1, \ldots, r\}.$$

Thus, we have proved that

$$v(a) = \min\{g_{\omega_\ell}(a) \mid \ell = 1, \ldots, r\} \tag{3.37}$$

for all a satisfying $\|a - \bar{a}\| < \delta_0$. As it has been noted, the functions g_ω, $\omega \in \Omega$, are locally Lipschitz on $\mathbb{R}^{nm}$. Hence, applying [20, Prop. 2.3.6 and 2.3.12] to the minimum function in (3.37), we can assert that v is locally Lipschitz at $\bar{a}$.

The proof is complete. □

Theorem 3.6 (Local upper Lipschitz property of the global solution map) *The global solution map $F : \mathbb{R}^{nm} \rightrightarrows \mathbb{R}^{nk}$ is locally upper Lipschitz at $\bar{a}$, i.e.,*

there exist $L > 0$ *and* $\delta > 0$ *such that*

$$F(a) \subset F(\bar{a}) + L\|a - \bar{a}\|\bar{B}_{\mathbb{R}^{nk}} \tag{3.38}$$

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

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

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

Proof. Let Ω, $\Omega_1 = \{\omega_1, \ldots, \omega_r\}$, $x_\omega(a) = (x^1_\omega(a), \ldots, x^k_\omega(a)) \in \mathbb{R}^{nk}$, and δ_0 be constructed as in the proof of the above theorem. For any $\omega \in \Omega$, the vector function $x_\omega(.)$, which maps $\mathbb{R}^{nm}$ to $\mathbb{R}^{nk}$, is continuously differentiable. Hence, there exist $L_\omega > 0$ and $\delta_\omega > 0$ such that

$$\|x_\omega(a) - x_\omega(\widetilde{a})\| \leq L_\omega\|a - \widetilde{a}\| \tag{3.39}$$

for any a, $\widetilde{a}$ satisfying $\|a - \bar{a}\| < \delta_\omega$ and $\|\widetilde{a} - \bar{a}\| < \delta_\omega$. Set

$$L = \max\{L_{\omega_1}, \ldots, L_{\omega_r}\} \;\text{ and }\; \delta = \min\{\delta_0, \delta_{\omega_1} \ldots, \delta_{\omega_r}\}.$$

Then, for every a satisfying $\|a - \bar{a}\| < \delta$, by (3.36) and (3.39) one has

$$\begin{aligned} F(a) \subset \{x_{\omega_1}(a), \ldots, x_{\omega_r}(a)\} &\subset \{x_{\omega_1}(\bar{a}), \ldots, x_{\omega_r}(\bar{a})\} + L\|a - \bar{a}\|\bar{B}_{\mathbb{R}^{nk}} \\ &= F(\bar{a}) + L\|a - \bar{a}\|\bar{B}_{\mathbb{R}^{nk}}. \end{aligned}$$

Hence, inclusion (3.38) is valid for every a satisfying $\|a - \bar{a}\| < \delta$. □

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* $F_1 : \mathbb{R}^{nm} \rightrightarrows \mathbb{R}^{nk}$ *has the Aubin property at* $(\bar{a}, \bar{x})$, *i.e., there exist* $L_1 > 0$, $\varepsilon > 0$, *and* $\delta_1 > 0$ *such that*

$$F_1(a) \cap B(\bar{x}, \varepsilon) \subset F_1(\widetilde{a}) + L_1\|a - \widetilde{a}\|\bar{B}_{\mathbb{R}^{nk}} \tag{3.40}$$

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

Proof. Suppose that $\bar{x} = (\bar{x}^1, ..., \bar{x}^k) \in F_1(\bar{a})$ and $\bar{x}^{j_1} \neq \bar{x}^{j_2}$ for all $j_1, j_2 \in J$ with $j_2 \neq j_1$. Denote by J_1 the set of the indexes $j \in J$ such that $\bar{x}^j$ is attractive w.r.t. the data set $\{\bar{a}^1, \ldots, \bar{a}^m\}$. Put $J_2 = J \setminus J_1$. For every $j \in J_1$, by Theorem 3.3 one has

$$\|\bar{a}^i - \bar{x}^j\| < \|\bar{a}^i - \bar{x}^q\| \quad (\forall i \in I(j),\ \forall q \in J \setminus \{j\}). \tag{3.41}$$

In addition, the following holds:

$$\bar{x}^j = \frac{1}{|I(j)|} \sum_{i \in I(j)} \bar{a}^i, \tag{3.42}$$

where $I(j) = \{i \in I \mid \bar{a}^i \in A[\bar{x}^j]\}$. For every $j \in J_2$, by Theorem 3.3 one has

$$\|\bar{x}^q - \bar{a}^p\| < \|\bar{x}^j - \bar{a}^p\| \quad (\forall q \in J_1,\ \forall p \in I(q)). \tag{3.43}$$

Let $\varepsilon_0 > 0$ be such that $\|\bar{x}^{j_1} - \bar{x}^{j_2}\| > \varepsilon_0$ for all $j_1, j_2 \in J$ with $j_2 \neq j_1$.

By (3.41) and (3.43), there exist $\delta_0 > 0$ and $\varepsilon \in \left(0, \frac{\varepsilon_0}{4}\right)$ such that

$$\|a^i - x^j\| < \|a^i - x^q\| \quad (\forall j \in J_1,\ \forall i \in I(j),\ \forall q \in J \setminus \{j\}) \tag{3.44}$$

and

$$\|x^q - a^p\| < \|x^j - a^p\| \quad (\forall j \in J_2,\ \forall q \in J_1,\ \forall p \in I(q)) \tag{3.45}$$

for all $a = (a^1, ..., a^m) \in \mathbb{R}^{nm}$ and $x = (x^1, \ldots, x^k) \in \mathbb{R}^{nk}$ with $\|a - \bar{a}\| < \delta_0$ and $\|x - \bar{x}\| < 2k\varepsilon$. As $\bar{x}^{j_1} \neq \bar{x}^{j_2}$ for all $j_1, j_2 \in J$ with $j_2 \neq j_1$, by taking a smaller $\varepsilon > 0$ (if necessary), for any $x = (x^1, \ldots, x^k) \in \mathbb{R}^{nk}$ satisfying $\|x - \bar{x}\| < 2k\varepsilon$ we have $x^{j_1} \neq x^{j_2}$ for all $j_1, j_2 \in J$ with $j_2 \neq j_1$.

For every $j \in J_1$ and $a = (a^1, ..., a^m) \in \mathbb{R}^{nm}$, define

$$x^j(a) = \frac{1}{|I(j)|} \sum_{i \in I(j)} a^i. \tag{3.46}$$

Comparing (3.46) with (3.42) yields $x^j(\bar{a}) = \bar{x}^j$ for all $j \in J_1$. Then, by the continuity of the vector functions $x^j(.)$, where $j \in J_1$, we may assume that $\|x^j(\widetilde{a}) - \bar{x}^j\| < \varepsilon$ for all $j \in J_1$ and $\widetilde{a} = (\widetilde{a}^1, ..., \widetilde{a}^m) \in \mathbb{R}^{nm}$ satisfying $\|\widetilde{a} - \bar{a}\| < \delta_0$ (one can take a smaller $\delta_0 > 0$, if necessary).

Since the vector functions $x^j(.)$, $j \in J_1$, are continuously differentiable, there exist $L_1 > 0$ such that

$$\|x^j(a) - x^j(\widetilde{a})\| \leq \frac{1}{k} L_1 \|a - \widetilde{a}\| \tag{3.47}$$

for any a, $\widetilde{a}$ satisfying $\|a - \bar{a}\| < \delta_0$ and $\|\widetilde{a} - \bar{a}\| < \delta_0$ (one can take a smaller $\delta_0 > 0$, if necessary). Choose $\delta_1 \in (0, \delta_0)$ as small as $\frac{2}{k} L_1 \delta_1 < \varepsilon$.

With the chosen constants $L_1 > 0$, $\varepsilon > 0$, and $\delta_1 > 0$, let us show that the inclusion (3.40) is fulfilled for all a and $\widetilde{a}$ satisfying $\|a - \bar{a}\| < \delta_1$ and $\|\widetilde{a} - \bar{a}\| < \delta_1$.

Let a and $\widetilde{a}$ be such that $\|a - \bar{a}\| < \delta_1$ and $\|\widetilde{a} - \bar{a}\| < \delta_1$. Select an arbitrary element $x = (x^1, \ldots, x^k)$ of the set $F_1(a) \cap B(\bar{x}, \varepsilon)$. Put $\widetilde{x}^j = x^j(\widetilde{a})$ for all $j \in J_1$, where $x^j(a)$ is given by (3.46). For any $j \in J_2$, set $\widetilde{x}^j = x^j$.

CLAIM 1. *The vector* $\widetilde{x} = (\widetilde{x}^1, \ldots, \widetilde{x}^k)$ *belongs to* $F_1(\widetilde{a})$.

Indeed, the inequalities $\|a - \bar{a}\| < \delta_1$ and $\|x - \bar{x}\| < \varepsilon$ imply that both properties (3.44) and (3.45) are available. From (3.44) it follows that, for every $j \in J_1$, the attraction set $A[x^j]$ is $\{a^i \mid i \in I(j)\}$. Since $I(j) \neq \emptyset$ for each $j \in J_1$ and $x \in F_1(a)$, by Theorem 3.3 we have

$$x^j = \frac{1}{|I(j)|} \sum_{i \in I(j)} a^i. \tag{3.48}$$

Comparing (3.48) with (3.46) yields $x^j = x^j(a)$ for all $j \in J_1$. By (3.45) we see that, for every $j \in J_2$, the attraction set $A[x^j]$ is empty. Moreover, one has

$$x^j \notin \mathcal{A}[x] \quad (\forall j \in J_2) \tag{3.49}$$

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 $p \in I(q)$.

For each $j \in J_1$, using (3.47) we have

$$\begin{aligned} \|x^j(\widetilde{a}) - \bar{x}^j\| &\leq \|x^j(\widetilde{a}) - x^j(a)\| + \|x^j(a) - \bar{x}^j\| \\ &\leq \tfrac{1}{k} L_1 \|\widetilde{a} - a\| + \varepsilon \\ &\leq \tfrac{1}{k} L_1 \left(\|\widetilde{a} - \bar{a}\| + \|\bar{a} - a\|\right) + \varepsilon \\ &\leq \tfrac{2}{k} L_1 \delta_1 + \varepsilon < 2\varepsilon. \end{aligned}$$

Besides, for each $j \in J_2$, we have $\|x^j(\widetilde{a}) - \bar{x}^j\| = \|x^j - \bar{x}^j\| < \varepsilon$. Therefore,

$$\|\widetilde{x} - \bar{x}\| = \sum_{j \in J_1} \|x^j(\widetilde{a}) - \bar{x}^j\| + \sum_{j \in J_2} \|x^j - \bar{x}^j\| < 2k\varepsilon.$$

In combination with the inequality $\|\widetilde{a} - \bar{a}\| < \delta_1$, this assures that the properties (3.44) and (3.45), where $\widetilde{a}$ and $x(\widetilde{a})$ respectively play the roles of a and x, hold. In other words, one has

$$\|\widetilde{a}^i - \widetilde{x}^j\| < \|\widetilde{a}^i - \widetilde{x}^q\| \quad (\forall j \in J_1,\ \forall i \in I(j),\ \forall q \in J \setminus \{j\}) \tag{3.50}$$

and

$$\|\widetilde{x}^q - \widetilde{a}^p\| < \|\widetilde{x}^j - \widetilde{a}^p\| \quad (\forall j \in J_2,\ \forall q \in J_1,\ \forall p \in I(q)). \tag{3.51}$$

So, similar to the above case of x, for every $j \in J_1$, the attraction set $A[\widetilde{x}^j]$ is $\{\widetilde{a}^i \mid i \in I(j)\}$. Recall that $I(j) \neq \emptyset$ for each $j \in J_1$ and $\widetilde{x}^j$ was given by

$$\widetilde{x}^j = x^j(\widetilde{a}) = \frac{1}{|I(j)|} \sum_{i \in I(j)} \widetilde{a}^i. \tag{3.52}$$

In addition, for every $j \in J_2$, the attraction set $A[\widetilde{x}^j]$ is empty and one has

$$\widetilde{x}^j \notin \mathcal{A}[\widetilde{x}] \quad (\forall j \in J_2), \tag{3.53}$$

where $\mathcal{A}[\widetilde{x}]$ is the union of the balls $\bar{B}(\widetilde{a}^p, \|\widetilde{a}^p - \widetilde{x}^q\|)$ with $p \in I$, $q \in J$ satisfying $p \in I(q)$. Besides, from (3.50) and (3.51) it follows that $|J_i(\widetilde{x})| = 1$ for every $i \in I$. Since $\|\widetilde{x} - \bar{x}\| < 2k\varepsilon$, we have $\widetilde{x}^{j_1} \neq \widetilde{x}^{j_2}$ for all $j_1, j_2 \in J$ with $j_2 \neq j_1$. Due to the last two properties and (3.52), (3.53), by Theorem 3.4 we conclude that $\widetilde{x} \in F_1(\widetilde{a})$.

Claim 2. *One has* $x \in \widetilde{x} + L_1\|a - \widetilde{a}\|\bar{B}_{\mathbb{R}^{nk}}$.

Indeed, since $x^j = x^j(a)$ for all $j \in J_1$, $\widetilde{x}^j = x^j$ for any $j \in J_2$, by (3.52) and (3.47) we have

$$\begin{aligned}\|x - \widetilde{x}\| &= \sum_{j\in J_1} \|x^j - \widetilde{x}^j\| + \sum_{j\in J_2} \|x^j - \widetilde{x}^j\| \\ &= \sum_{j\in J_1} \|x^j(a) - x^j(\widetilde{a})\| \\ &\leq k.\frac{1}{k} L_1\|a - \widetilde{a}\| = L_1\|a - \widetilde{a}\|.\end{aligned}$$

It follows that $x \in \widetilde{x} + L_1\|a - \widetilde{a}\|\bar{B}_{\mathbb{R}^{nk}}$.

Combining Claim 2 with Claim 1, we have $x \in F_1(\widetilde{a}) + L_1\|a - \widetilde{a}\|\bar{B}_{\mathbb{R}^{nk}}$. Thus, property (3.40) is valid for all a and $\widetilde{a}$ satisfying $\|a - \bar{a}\| < \delta_1$ and $\|\widetilde{a} - \bar{a}\| < \delta_1$. □

3.6 Conclusions

We have proved that the minimum sum-of-squares clustering problem always has a global solution and, under a mild condition, the global solution set is finite and the components of each global solution can be computed by an explicit formula. Based on a new concept of nontrivial local solution, we have got necessary and sufficient conditions for a system of centroids to be a nontrivial local solution.

We also have established the local Lipschitz property of the optimal value function, the local upper Lipschitz property of the global solution map, and the local Lipschitz-like property of the local solution map of the MSSC problem. Thanks to the obtained complete characterizations of the nontrivial local solutions, one can understand better the performance of the k-means algorithm.

Chapter 4

Some Incremental Algorithms for the Clustering Problem

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

Based on the **D**ifference-of-**C**onvex functions **A**lgorithms (DCAs) in DC programming and the qualitative properties of the MSSC problem established in Chapter 3, we suggest several improvements of the incremental algorithms of Ordin and Bagirov [71] and of Bagirov [7]. Properties of the new algorithms, including finite convergence, convergence, and rate of convergence, are presented herein. The results of our numerical tests of these algorithms on several real-world databases are shown.

The present chapter is written on the basis of paper No. 3 and paper No. 4 in the List of author's related papers (see p. 112).

4.1 Incremental Clustering Algorithms

There are many algorithms to solve the MSSC problem (see, e.g., [6,7,9,12, 13,71,98], and the references therein). Since it is a NP-hard problem [3,67] when either the number of the data features or the number of the clusters is a part of the input, the fact that the existing algorithms can give at most some local solutions is understandable.

The k-means clustering algorithm (see Section 3.3 and see also, e.g., [1], [39], [43], and [66]) is the best known solution method for the MSSC problem.

To improve its effectiveness, the *global k-means, modified global k-means, and fast global k-means clustering algorithms* have been proposed in [6, 12, 33, 49, 61, 98].

Since the quality of the computation results greatly depends on the starting points, it is reasonable to look for good starting points. The DCA (Difference-of Convex-functions Algorithms), which has been applied to the MSSC problem in [7, 60], can be used for the purpose.

One calls a clustering algorithm *incremental* if the number of the clusters increases step by step. As noted in [71, p. 345], the available numerical results demonstrate that *incremental clustering algorithms* (see, e.g., [6, 33, 49, 71]) are efficient for dealing with large data sets.

Recently, Ordin and Bagirov [71] have proposed an incremental clustering algorithm based on control parameters to find good starting points for k-means algorithm. Note that, in his earlier paper [7], Bargirov suggested another incremental clustering algorithm based on DC programming and DCA. We will propose several improvements of the just mentioned incremental algorithms to solve the MSSC problem in (3.2).

The incremental clustering algorithms in [7, 44, 71] start with the computation of the centroid of the whole data set and attempt to optimally add one new centroid at each stage. The process is continued until finding k centroids for problem (3.2).

We are interested in analyzing and developing the *incremental heuristic clustering algorithm* of Ordin and Bagirov [71] and the *incremental DC clustering algorithm* of Bagirov [7]. By constructing some concrete MSSC problems with small data sets, we will show how these algorithms work. It turns out that, due to the exact stopping criterion, the computation by the second algorithm may not stop. We will propose one modified version for the incremental heuristic clustering algorithm of [71] and three modified versions for the incremental DC clustering algorithm of [7].

4.2 Ordin-Bagirov's Clustering Algorithm

This section is devoted to the incremental heuristic algorithm of Ordin and Bagirov [71, pp. 349–353] and some properties of the algorithm.

4.2.1 Basic constructions

Let ℓ be an index with $1 \le \ell \le k-1$ 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

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

Applying the natural clustering procedure described in (3.3) to the centroid system $\{\bar{x}^1, ..., \bar{x}^\ell\}$, one divides A into ℓ clusters with the centers $\bar{x}^1, ..., \bar{x}^\ell$. For every $i \in I$, put

$$d_\ell(a^i) = \min\left\{\|\bar{x}^1 - a^i\|^2, ..., \|\bar{x}^\ell - a^i\|^2\right\}. \quad (4.2)$$

The formula $g(y) = f_{\ell+1}(\bar{x}^1, ..., \bar{x}^\ell, y)$ where, in accordance with (4.1),

$$f_{\ell+1}(x) = \frac{1}{m}\sum_{i=1}^{m}\Big(\min_{j=1,...,\ell+1}\|a^i - x^j\|^2\Big) \quad \forall x = (x^1, \ldots, x^\ell, x^{\ell+1}) \in \mathbb{R}^{n(\ell+1)},$$

defines our *auxiliary cluster function* $g : \mathbb{R}^n \to \mathbb{R}$. From (4.2) it follows that

$$g(y) = \frac{1}{m}\sum_{i=1}^{m}\min\left\{d_\ell(a^i),\, \|y - a^i\|^2\right\}. \quad (4.3)$$

The problem

$$\min\left\{g(y) \mid y \in \mathbb{R}^n\right\} \quad (4.4)$$

is called *the auxiliary clustering problem.* For each $i \in I$, one has

$$\min\left\{d_\ell(a^i),\, \|y - a^i\|^2\right\} = \left[d_\ell(a^i) + \|y - a^i\|^2\right] - \max\left\{d_\ell(a^i), \|y - a^i\|^2\right\}.$$

So, the objective function of (4.4) can be represented as $g(y) = g^1(y) - g^2(y)$, where

$$g^1(y) = \frac{1}{m}\sum_{i=1}^{m} d_\ell(a^i) + \frac{1}{m}\sum_{i=1}^{m}\|y - a^i\|^2 \quad (4.5)$$

is a smooth convex function and

$$g^2(y) = \frac{1}{m}\sum_{i=1}^{m}\max\left\{d_\ell(a^i), \|y - a^i\|^2\right\}. \quad (4.6)$$

is a nonsmooth convex function. Consider the open set

$$Y_1 := \bigcup_{i\in I} B\big(a^i, d_\ell(a^i)\big) = \left\{y \in \mathbb{R}^n \mid \exists i \in I \text{ with } \|y - a^i\|^2 < d_\ell(a^i)\right\}, \quad (4.7)$$

which is the finite union of certain open balls with the centers a^i $(i \in I)$, and put

$$Y_2 := \mathbb{R}^n \setminus Y_1 = \big\{y \in \mathbb{R}^n \mid \|y - a^i\|^2 \geq d_\ell(a^i),\ \forall i \in I\big\}.$$

One sees that all the points $\bar{x}^1, ..., \bar{x}^\ell$ are contained in Y_2. Since $\ell < k \leq m$ and the data points $a^1, \ldots, a^m$ are pairwise distinct, there must exist at least one $i \in I$ with $d_\ell(a^i) > 0$ (otherwise, every data point coincides with a point from the set $\{\bar{x}^1, ..., \bar{x}^\ell\}$, which is impossible). Hence $Y_1 \neq \emptyset$. By (4.5) and (4.6), we have

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

and

$$g(y) = \frac{1}{m}\sum_{i=1}^{m} d_\ell(a^i) \quad \forall y \in Y_2.$$

Therefore, any *iteration process* for solving (4.4) 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

$$\min\Big\{f_{\ell+1}(x) := \frac{1}{m}\sum_{i=1}^{m}\Big(\min_{j=1,\ldots,\ell+1}\|a^i - x^j\|^2\Big) \mid x = (x^1, \ldots, x^{\ell+1}) \in \mathbb{R}^{n(\ell+1)}\Big\}, \tag{4.8}$$

we can use the following procedure [71, pp. 349–351]. Fixing any $y \in Y_1$, one divides the data set A into two disjoint subsets

$$A_1(y) := \{a^i \in A \mid \|y - a^i\|^2 < d_\ell(a^i)\} \tag{4.9}$$

and

$$A_2(y) := \{a^i \in A \mid \|y - a^i\|^2 \geq d_\ell(a^i)\}.$$

Clearly, $A_1(y)$ consists of all the data points standing closer to y than to their cluster centers. Since $y \in Y_1$, the set $A_1(y)$ is nonempty. Note that

$$g(y) = \frac{1}{m}\Big(\sum_{a^i \in A_1(y)}\|y - a^i\|^2 + \sum_{a^i \in A_2(y)} d_\ell(a^i)\Big). \tag{4.10}$$

Put $z_{\ell+1}(y) = f_\ell(\bar{x}) - g(y)$. Since $f_\ell(\bar{x}) = f_\ell(\bar{x}^1, ..., \bar{x}^\ell)$ and

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

the quantity $z_{\ell+1}(y) > 0$ expresses *the decrease of the minimum sum-of-squares clustering criterion* when one replaces the current centroid system

$\{\bar{x}^1, ..., \bar{x}^\ell\}$ with ℓ centers by the new one $\{\bar{x}^1, ..., \bar{x}^\ell, y\}$ with $\ell + 1$ centers. Thanks to the formula

$$f_\ell(\bar{x}) = \frac{1}{m} \sum_{a^i \in A} d_\ell(a^i)$$

and (4.10), one has the representation

$$z_{\ell+1}(y) = \frac{1}{m} \sum_{a^i \in A_1(y)} \left(d_\ell(a^i) - \|y - a^i\|^2\right),$$

which can be rewritten as

$$z_{\ell+1}(y) = \frac{1}{m} \sum_{i \in I} \max\left\{0, d_\ell(a^i) - \|y - a^i\|^2\right\}. \tag{4.11}$$

Further operations depend greatly on the data points belonging to Y_1. It is easy to show that $a \in A \cap Y_1$ if and only if $a \in A$ and $a \notin \{\bar{x}^1, ..., \bar{x}^\ell\}$. For every point $y = a \in A \cap Y_1$, one computes $z_{\ell+1}(a)$ by (4.11). Then, one finds the value

$$z^1_{\max} := \max\left\{z_{\ell+1}(a) \mid a \in A \cap Y_1\right\}. \tag{4.12}$$

The selection of 'good' starting points to solve (4.8) is controlled by two parameters: $\gamma_1 \in [0, 1]$ and $\gamma_2 \in [0, 1]$. The role of each of them will be explained later. Since the choice of these parameters can be made from the computational experience of applying the algorithm in question, the authors of [71] call their algorithm *heuristic*.

Using γ_1, one can find the set

$$\bar{A}_1 := \{a \in A \cap Y_1 \mid z_{\ell+1}(a) \geq \gamma_1 z^1_{\max}\}. \tag{4.13}$$

For $\gamma_1 = 0$, one has $\bar{A}_1 = A \cap Y_1$, i.e., $\bar{A}_1$ consists of *all* the data points belonging to Y_1. In contrast, for $\gamma_1 = 1$, the set $\bar{A}_1$ just consists of the data points yielding the *largest decrease* $z^1_{\max}$. (As noted by Ordin and Bagirov [71], the *global k-means algorithm* in [61] uses one of such data points for finding a $(\ell + 1)$-th centroid.) *Thus, γ_1 represents the tolerance in choosing appropriate points from $A \cap Y_1$.* For each $a \in \bar{A}_1$, one finds the set $A_1(a)$ and computes its barycenter, which is denoted by $c(a)$. Then, one replaces a by $c(a)$, because $c(a)$ represents the set $A_1(a)$ better than a. Since $g(c(a)) \leq g(a) < f_\ell(\bar{x})$, one must have $c(a) \in Y_1$. Put

$$\bar{A}_2 = \{c(a) \mid a \in \bar{A}_1\}. \tag{4.14}$$

For each $c \in \bar{A}_2$, one computes the value $z_{\ell+1}(c)$ by using (4.11). Then, we find

$$z^2_{\max} := \max\left\{z_{\ell+1}(c) \mid c \in \bar{A}_2\right\}. \tag{4.15}$$

Clearly, $z^2_{\max}$ is the largest decrease among the values $f_{\ell+1}(\bar{x}^1, ..., \bar{x}^\ell, c)$, where $c \in \bar{A}_2$, in comparison with the value $f_\ell(\bar{x})$.

Using γ_2, one computes

$$\bar{A}_3 = \left\{c \in \bar{A}_2 \mid z_{\ell+1}(c) \geq \gamma_2 z^2_{\max}\right\}. \tag{4.16}$$

For $\gamma_2 = 0$, one has $\bar{A}_3 = \bar{A}_2$. For $\gamma_2 = 1$, one sees that $\bar{A}_3$ just contains the barycenters $c \in \bar{A}_2$ with the largest decrease of the objective function $g(y) = f_{\ell+1}(\bar{x}^1, ..., \bar{x}^\ell, y)$ of (4.4). (As noted in [71, p. 315], for $\gamma_1 = 0$ and $\gamma_2 = 1$, one recovers the selection of a 'good' starting point in the *modified global k-means algorithm* suggested by Bargirov in [6].) *Thus, γ_2 represents the tolerance in selecting appropriate points from $\bar{A}_2$.* The set

$$\Omega := \left\{(\bar{x}^1, ..., \bar{x}^\ell, c) \mid c \in \bar{A}_3\right\} \tag{4.17}$$

contains the 'good' starting points to solve (4.8).

4.2.2 Version 1 of Ordin-Bagirov's algorithm

On the basis of the set Ω in (4.17), the computation of a set of starting points to solve problem (4.8) is controlled by a parameter $\gamma_3 \in [1, \infty)$. One applies the k-means algorithm to problem (4.8) for each initial centroid system $(\bar{x}^1, ..., \bar{x}^\ell, c) \in \Omega$. In result, one obtains a set of vectors $x = (x^1, \ldots, x^{\ell+1})$ from $\mathbb{R}^{n(\ell+1)}$. Denote by $\bar{A}_4$ the set of the components $x^{\ell+1}$ of these vectors. Then, one computes the number

$$f^{\min}_{\ell+1} := \min\left\{g(y) \mid y \in \bar{A}_4\right\}. \tag{4.18}$$

Using γ_3, one finds the set

$$\bar{A}_5 = \left\{y \in \bar{A}_4 \mid g(y) \leq \gamma_3 f^{\min}_{\ell+1}\right\}. \tag{4.19}$$

For $\gamma_3 = 1$, one sees that $\bar{A}_5$ contains all the points $x \in \bar{A}_4$ at which the function $f_{\ell+1}(x)$ attains its minimum value. In contrast, if γ_3 is large enough, then $\bar{A}_5 = \bar{A}_4$. *Thus, γ_3 represents the tolerance in choosing appropriate points from $\bar{A}_4$.* To solve problem (4.8), one will use the points from $\bar{A}_5$.

The process of finding starting points is summarized as follows.

Procedure 4.1 (for finding starting points)

Input: *An approximate solution* $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ *of problem* (4.1), $\ell \geq 1$.
Output: *A set* $\bar{A}_5$ *of starting points to solve problem* (4.8).
Step 1. Select three control parameters: $\gamma_1 \in [0, 1]$, $\gamma_2 \in [0, 1]$, $\gamma_3 \in [1, \infty)$.
Step 2. Compute $z^1_{\max}$ *by* (4.12) *and the set* $\bar{A}_1$ *by* (4.13).
Step 3. Compute the set $\bar{A}_2$ *by* (4.14), $z^2_{\max}$ *by* (4.15), *and the set* $\bar{A}_3$ *by* (4.16).
Step 4. Using (4.17), *form the set* Ω.
Step 5. Apply the k-means algorithm to problem (4.8) *for each initial centroid system* $(\bar{x}^1, ..., \bar{x}^\ell, c) \in \Omega$ *to get the set* $\bar{A}_4$.
Step 6. Compute the value $f^{\min}_{\ell+1}$ *by* (4.18).
Step 7. Form the set $\bar{A}_5$ *by* (4.19).

Now we are able to present the original version of Ordin-Bagirov's algorithm [71, Algorithm 2, p. 352] for solving problem (3.2).

Algorithm 4.1 (Ordin-Bagirov's Algorithm, Version 1)

Input*: The data set* $A = \{a^1, \ldots, a^m\}$.
Output*: A centroid system* $\{\bar{x}^1, \ldots, \bar{x}^k\}$.
Step 1. Compute the barycenter $a^0 = \dfrac{1}{m}\sum\limits_{i=1}^{m} a^i$ *of the data set A, put* $\bar{x}^1 = a^0$, *and set* $\ell = 1$.
Step 2. If $\ell = k$, *then stop. Problem* (3.2) *has been solved.*
Step 3. Apply ***Procedure*** *4.1 to compute the set* $\bar{A}_5$ *of starting points.*
Step 4. For each $\bar{y} \in \bar{A}_5$, *apply the k-means algorithm to* (4.8) *with the starting point* $(\bar{x}^1, ..., \bar{x}^\ell, \bar{y})$ *to find an approximate solution* $x = (x^1, \ldots, x^{\ell+1})$. *Denote by* $\bar{A}_6$ *the set of these solutions.*
Step 5. Select a point $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ *from* $\bar{A}_6$ *satisfying*

$$f_{\ell+1}(\hat{x}) = \min\left\{f_{\ell+1}(x) \mid x \in \bar{A}_6\right\}. \tag{4.20}$$

Define $\bar{x}^j = \hat{x}^j$, $j = 1, \ldots, \ell + 1$. *Set* $\ell = \ell + 1$ *and go to Step 2.*

Depending on the sizes of the data sets, the following rule to choose the control parameters triple $\gamma = (\gamma_1, \gamma_2, \gamma_3)$ can be used [71, p. 352]:

• For small data sets (with the number of data points $m \leq 200$), choose $\gamma = (0.3, 0.3, 3)$;

• For medium size data sets ($200 < m \leq 6000$), choose $\gamma = (0.5, 0.8, 1.5)$, or $\gamma = (0.5, 0.9, 1.5)$;

• For large data sets (with $m > 6000$), choose $\gamma = (0.85, 0.99, 1.1)$, or $\gamma = (0.9, 0.99, 1.1)$.

Going back to Procedure 4.1 and Algorithm 4.1, we have the following remarks. When one applies the k-means algorithm to problem (4.8) for an initial centroid system $(\bar{x}^1, ..., \bar{x}^\ell, c) \in \Omega$ to get the new centroid system $x = (x^1, \ldots, x^{\ell+1})$ and put $\bar{y} = x^{\ell+1}$, then $\bar{y}$ is good just in the combination with the centroids $x^1, \ldots, x^\ell$. If one combines $\bar{y}$ with the given centroids $\bar{x}^1, ..., \bar{x}^\ell$, as it is done in Step 4 of the above algorithm, then it may happen that $f_{\ell+1}(x^1, \ldots, x^\ell, \bar{y}) < f_{\ell+1}(\bar{x}^1, ..., \bar{x}^\ell, \bar{y})$. If so, one wastes the available centroid system $(x^1, \ldots, x^\ell, \bar{y})$ with $\bar{y} \in \bar{A}_5$. And the application of the k-means algorithm to problem (4.8) with the starting point $(\bar{x}^1, ..., \bar{x}^\ell, \bar{y})$ to find an approximate solution $x = (x^1, \ldots, x^{\ell+1})$, as suggested in Step 4 of the above algorithm, is not very suitable. These remarks lead us to proposing Version 2 of Ordin-Bagirov's algorithm, which is simpler than the original version.

4.2.3 Version 2 of Ordin-Bagirov's algorithm

The computation of an approximate solution of problem (4.8) on the basis of the set Ω in (4.17) is controlled by a parameter $\gamma_3 \in [1, \infty)$. One applies the k-means algorithm to problem (4.8) for each initial centroid system $(\bar{x}^1, ..., \bar{x}^\ell, c) \in \Omega$. In result, one obtains a set of points $x = (x^1, \ldots, x^{\ell+1})$ from $\mathbb{R}^{n(\ell+1)}$, which is denoted by $\widetilde{A}_4$. Then, one computes the number

$$\widetilde{f}_{\ell+1}^{\min} := \min \left\{ f_{\ell+1}(x) \mid x \in \widetilde{A}_4 \right\}. \tag{4.21}$$

Using γ_3, one finds the set

$$\widetilde{A}_5 = \left\{ x \in \widetilde{A}_4 \mid f_{\ell+1}(x) \leq \gamma_3 \widetilde{f}_{\ell+1}^{\min} \right\}. \tag{4.22}$$

For $\gamma_3 = 1$, one sees that $\widetilde{A}_5$ contains all the points $x \in \widetilde{A}_4$ at which the function $f_{\ell+1}(x)$ attains its minimum value. In contrast, if γ_3 is large enough, then $\widetilde{A}_5 = \widetilde{A}_4$. *Thus, γ_3 represents the tolerance in choosing appropriate*

points from $\widetilde{A}_4$. Selecting an arbitrary point $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ from $\widetilde{A}_5$, one has an approximate solution of problem (4.8).

The above procedure for finding a new centroid system $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ with $\ell + 1$ centers, starting from a given centroid system $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ with ℓ centers, can be described as follows.

Procedure 4.2 (for finding a new centroid system)

Input: *An approximate solution* $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ *of problem* (4.1), $\ell \geq 1$.
Output: *An approximate solution* $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ *of problem* (4.8).
Step 1. Select three control parameters: $\gamma_1 \in [0, 1]$, $\gamma_2 \in [0, 1]$, $\gamma_3 \in [1, \infty)$.
Step 2. Compute $z^1_{\max}$ *by* (4.12) *and the set* $\bar{A}_1$ *by* (4.13).
Step 3. Compute the set $\bar{A}_2$ *by* (4.14), $z^2_{\max}$ *by* (4.15), *and the set* $\bar{A}_3$ *by* (4.16).
Step 4. Using (4.17), *form the set* Ω.
Step 5. Apply the k-means algorithm to problem (4.8) *for each initial centroid system* $(\bar{x}^1, ..., \bar{x}^\ell, c) \in \Omega$ *to get the set* $\widetilde{A}_4$ *of candidates for approximate solutions of* (4.8).
Step 6. Compute the value $\widetilde{f}^{\min}_{\ell+1}$ *by* (4.21) *and the set* $\widetilde{A}_5$ *by* (4.22).
Step 7. Pick a point $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ *from* $\widetilde{A}_5$.

Now we are able to present Version 2 of Ordin-Bagirov's algorithm [71, Algorithm 2, p. 352] for solving problem (3.2).

Algorithm 4.2 (Ordin-Bagirov's Algorithm, Version 2)

Input*: The parameters* n, m, k, *and the data set* $A = \{a^1, \ldots, a^m\}$.
Output*: A centroid system* $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k)$ *and the corresponding clusters* $A^1, ..., A^k$.
Step 1. Compute the barycenter $a^0 = \dfrac{1}{m} \sum_{i=1}^{m} a^i$ *of the data set A, put* $\bar{x}^1 = a^0$, *and set* $\ell = 1$.
Step 2. If $\ell = k$, *then go to Step 5.*
Step 3. Use ***Procedure*** *4.2 to find an approximate solution* $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ *of problem* (4.8).
Step 4. Put $\bar{x}^j := \hat{x}^j$, $j = 1, \ldots, \ell + 1$. *Set* $\ell = \ell + 1$ *and go to Step 2.*

Step 5. Select an element $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k)$ *from the set*

$$\widetilde{A}_6 := \{x \in \widetilde{A}_5 \mid f_{\ell+1}(x) = \widetilde{f}_{\ell+1}^{\min}\}. \tag{4.23}$$

Using the centroid system $\bar{x}$*, apply the natural clustering procedure to partition* A *into* k *clusters* $A^1, ..., A^k$*. Print* $\bar{x}$ *and* $A^1, ..., A^k$*. Stop.*

To understand the performances of Algorithms 4.1 and 4.2, let us analyze two useful numerical examples of the MSSC problem in the form (3.2). For the sake of clarity and simplicity, data sets with only few data points, each has just two features, are considered.

Example 4.1 Choose $n = 2$, $m = 3$, $k = 2$, $A = \{a^1, a^2, a^3\}$, where

$$a^1 = (0, 0), \quad a^2 = (1, 0), \quad a^3 = (0, 1).$$

Let $\gamma_1 = \gamma_2 = 0.3, \gamma_3 = 3$. The barycenter of A is $a^0 = (\frac{1}{3}, \frac{1}{3})$.

The implementation of Algorithm 4.1 begins with computing $\bar{x}^1 = a^0$ and setting $\ell = 1$. Since $\ell < k$, we apply Procedure 4.1 to compute the set $\bar{A}_5$. By (4.2), one has $d_1(a^1) = \frac{2}{9}$, $d_1(a^2) = \frac{5}{9}$, $d_1(a^3) = \frac{5}{9}$. Using (4.11), we get $z_{\ell+1}(a^1) = \frac{2}{27}$, $z_{\ell+1}(a^2) = \frac{5}{27}$, and $z_{\ell+1}(a^3) = \frac{5}{27}$. So, by (4.12) and (4.13), one has $z^1_{\max} = \max\{\frac{2}{27}, \frac{5}{27}, \frac{5}{27}\} = \frac{5}{27}$ and $\bar{A}_1 = A$. Since $A_1(a^i) = \{a^i\}$ for $i \in I$, one obtains $c(a^i) = a^i$ for all $i \in I$. Therefore, by (4.14) and (4.15), $\bar{A}_2 = A$ and

$$z^2_{\max} = \max\Big\{\frac{2}{27}, \frac{5}{27}, \frac{5}{27}\Big\} = \frac{5}{27}.$$

It follows that $\bar{A}_3 = \{a^1, a^2, a^3\}$. Next, one applies the k-means algorithm to problem (4.8) with initial points from the Ω defined by (4.17) to compute $\bar{A}_4$. Starting from $(\bar{x}^1, a^1) \in \Omega$, one obtains the centroid system $\{(\frac{1}{2}, \frac{1}{2}), (0, 0)\}$. Starting from $(\bar{x}^1, a^2)$ and $(\bar{x}^1, a^3)$, one gets, respectively, the centroid systems $\{(\frac{1}{2}, \frac{1}{2}), (0, 0)\}$, $\{(0, \frac{1}{2}), (1, 0)\}$, and $\{(\frac{1}{2}, 0), (0, 1)\}$. Therefore,

$$\bar{A}_4 = \{(0, 0), (1, 0), (0, 1)\}.$$

By (4.3), we have $g(a^1) = \frac{10}{27}$, $g(a^2) = \frac{7}{27}$, and $g(a^3) = \frac{7}{27}$. So, by (4.18) one obtains $f^{\min}_{\ell+1} = \frac{7}{27}$. So, from (4.19) it follows that $\bar{A}_5 = \{(0, 0), (1, 0), (0, 1)\}$. Applying again the k-means algorithm to problem (4.8) with the initial points $(\bar{x}^1, \bar{y})$, $\bar{y} \in \bar{A}_5$, one gets

$$\bar{A}_6 = \Big\{\Big((\frac{1}{2}, \frac{1}{2}), (0, 0)\Big), \Big((0, \frac{1}{2}), (1, 0)\Big), \Big((\frac{1}{2}, 0), (0, 1)\Big)\Big\}.$$

The set of the values $f_{\ell+1}(x)$, $x \in \bar{A}_6$, is $\{\frac{1}{3}, \frac{1}{6}, \frac{1}{6}\}$. Then, there are two centroid systems in $\bar{A}_6$ satisfying the condition (4.20):

$$\hat{x} = \Big((0, \frac{1}{2}),\, (1, 0)\Big) \quad \text{and} \quad \hat{x} = \Big((\frac{1}{2}, 0), (0, 1)\Big). \tag{4.24}$$

Select any one from these centroid systems and increase ℓ by 1. Since $\ell = 2$, i.e., $\ell = k$, the computation ends. In result, one of the two centroid systems described by (4.24) is found.

The implementation of Algorithm 4.2 begins with putting $\bar{x}^1 = a^0$ and setting $\ell = 1$. Since $\ell < k$, we apply Procedure 4.2 to compute an approximate solution $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ of problem (4.8). The sets $\bar{A}_1, \bar{A}_2$ and $\bar{A}_3$ are defined as in Algorithm 4.1. Hence, $\bar{A}_3 = \bar{A}_2 = \bar{A}_1 = A = \{a^1, a^2, a^3\}$. Next, we apply the k-means algorithm to problem (4.8) with initial points from the set Ω defined by (4.17) to find $\widetilde{A}_4$. Since $\Omega = \{(\bar{x}^1, a^1), (\bar{x}^1, a^2), (\bar{x}^1, a^3)\}$, one gets

$$\widetilde{A}_4 = \Big\{\big((\frac{1}{2}, \frac{1}{2}), (0, 0)\big),\, \big((0, \frac{1}{2}), (1, 0)\big),\, \big((\frac{1}{2}, 0), (0, 1)\big)\Big\}.$$

By (4.21), the set of the values $\widetilde{f}_{\ell+1}(x)$, $x \in \widetilde{A}_4$, is $\Big\{\frac{1}{3}, \frac{1}{6}, \frac{1}{6}\Big\}$. Using (4.21), one gets $\widetilde{f}^{\min}_{\ell+1} = \frac{1}{6}$. Since $\gamma_3 = 3$, by (4.22) we have $\widetilde{A}_5 = \widetilde{A}_4$. Pick a point $\hat{x} = (\hat{x}^1, \hat{x}^2)$ from $\widetilde{A}_5$. Put $\bar{x}^j := \hat{x}^j,\ j = 1, 2$. Set $\ell := \ell + 1$. Since $\ell = k$, we use (4.23) to form the set

$$\widetilde{A}_6 = \Big\{\big((0, \frac{1}{2}), (1, 0)\big),\, \big((\frac{1}{2}, 0), (0, 1)\big)\Big\}.$$

Select any element $\bar{x} = (\bar{x}^1, \bar{x}^2)$ from $\widetilde{A}_6$ and stop. In result, we get one of the two centroid systems in (4.24).

In the above example, centroid systems resulted from both Algorithm 4.1 and Algorithm 4.2 belong to the global solution set of (3.2), which consists of the two centroid systems in (4.24).

We now present a modified version of Example 4.1 to show that *by Algorithm 4.1 (resp., Algorithm 4.2) one may not find a global solution of problem* (3.2). In other words, even for a very small data set, Algorithm 4.1 (resp., Algorithm 4.2) may yield a local, non-global solution of (3.2).

Example 4.2 Choose $n = 2$, $m = 4$, $k = 2$, $A = \{a^1, a^2, a^3, a^4\}$, where $a^1 = (0, 0)$, $a^2 = (1, 0)$, $a^3 = (0, 1)$, $a^4 = (1, 1)$. Let $\gamma_1 \in [0, 1]$, $\gamma_2 \in [0, 1]$, $\gamma_3 \in [1, \infty)$ be chosen arbitrarily. The barycenter of A is $a^0 = (\frac{1}{2}, \frac{1}{2})$.

To implement Algorithm 4.1, we put $\bar{x}^1 = a^0$ and set $\ell = 1$. By (4.2), one has $d_1(a^i) = \frac{1}{2}$ for $i \in I$. Using (4.11), we find that $z_{\ell+1}(a^i) = \frac{1}{8}$ for $i \in I$. So, by (4.12) and (4.13), one gets $z^1_{\max} = \frac{1}{8}$ and $\bar{A}_1 = A$. Since $A_1(a^i) = \{a^i\}$ for $i \in I$, one has $c(a^i) = a^i$ for $i \in I$. Therefore, by (4.14) and (4.15), $\bar{A}_2 = \{a^1, a^2, a^3, a^4\}$ and $z^2_{\max} = \frac{1}{8}$. So, $\bar{A}_3 = \{a^1, a^2, a^3, a^4\}$. Applying the k-means algorithm with the starting points $(\bar{x}^1, c) \in \Omega$, $c \in \bar{A}_3$, one obtains the centroid systems $\big((\frac{2}{3}, \frac{2}{3}), (0,0)\big)$, $\big((\frac{1}{3}, \frac{2}{3}), (1,0)\big)$, $\big((\frac{2}{3}, \frac{1}{3}), (0,1)\big)$, and $\big((\frac{1}{3}, \frac{1}{3}), (1,1)\big)$. Therefore, we have

$$\bar{A}_4 = \{(0,0),\ (1,0),\ (0,1),\ (1,1)\}.$$

Due to (4.3), one has $g((0,0)) = g((0,1)) = g((1,0)) = g((1,1)) = \frac{3}{8}$. So, by (4.2) one obtains $f^{\min}_{\ell+1} = \frac{3}{8}$. Thus, by (4.19), $\bar{A}_5 = \bar{A}_4$. For each $\bar{y} \in \bar{A}_5$, we apply the k-means algorithm with the starting point $(\bar{x}^1, \bar{y})$ to solve (4.8). In result, we get

$$\bar{A}_6 = \Big\{\big((\frac{2}{3}, \frac{2}{3}), (0,0)\big),\ \big((\frac{1}{3}, \frac{2}{3}), (1,0)\big),\ \big((\frac{2}{3}, \frac{1}{3}), (0,1)\big),\ \big((\frac{1}{3}, \frac{1}{3}), (1,1)\big)\Big\}. \quad (4.25)$$

Since $f_{\ell+1}(x) = \frac{1}{3}$ for every $x \in \bar{A}_6$, to satisfy condition (4.20), one can select any point $\hat{x} = (\hat{x}^1, \hat{x}^2)$ from $\bar{A}_6$. Define $\bar{x}^j := \hat{x}^j,\ j = 1, 2$. Set $\ell := \ell + 1$. Since $\ell = k$, the computation is completed. Thus, Algorithm 4.1 yields one of the four centroid systems in (4.25), which is a local, non-global solution of our clustering problem (see Remark 4.1 for detailed explanations).

The implementation of Algorithm 4.2 begins with putting $\bar{x}^1 = a^0$ and setting $\ell = 1$. Since $\ell < k$, we apply Procedure 4.2 to compute an approximate solution $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ of problem (4.8). The sets $\bar{A}_1, \bar{A}_2$ and $\bar{A}_3$ are defined as in Algorithm 4.1. Hence, $\bar{A}_3 = \bar{A}_2 = \bar{A}_1 = A = \{a^1, a^2, a^3, a^4\}$. Next, we apply the k-means algorithm to problem (4.8) with initial points from the set Ω defined by (4.17) to find $\widetilde{A}_4$. Since $\Omega = \{(\bar{x}^1, a^1), (\bar{x}^1, a^2), (\bar{x}^1, a^3), (\bar{x}^1, a^4)\}$, one gets

$$\widetilde{A}_4 = \Big\{\big((\frac{2}{3}, \frac{2}{3}), (0,0)\big),\ \big((\frac{1}{3}, \frac{2}{3}), (1,0)\big),\ \big((\frac{2}{3}, \frac{1}{3}), (0,1)\big),\ \big((\frac{1}{3}, \frac{1}{3}), (1,1)\big)\Big\}.$$

By (4.21), the set of the values $\widetilde{f}_{\ell+1}(x)$, $x \in \widetilde{A}_4$, is $\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$. Using (4.21), one gets $\widetilde{f}^{\min}_{\ell+1} = \frac{1}{3}$. By (4.22), we obtain $\widetilde{A}_5 = \widetilde{A}_4$. Set $\bar{x} = \hat{x}$, $\hat{x} \in \widetilde{A}_4$ and $\ell = 2$. Since $\ell = k$, $\widetilde{A}_6 = \widetilde{A}_5$. Select any centroid system from $\widetilde{A}_6$, e.g., $\bar{x} = \big((\frac{2}{3}, \frac{2}{3}), (0,0)\big)$. Applying the natural clustering procedure, one gets the clusters $A^1 = \{a^2, a^3, a^4\}$, $A^2 = \{a^1\}$, then stop.

Remark 4.1 Concerning the analysis given in Example 4.2, observe that every centroid system in $\bar{A}_6$ is a nontrivial local solution of problem (3.2). This assertion can be verified by Theorem 3.4. The value of the objective function at these centroid systems is $\frac{1}{3}$. Consider a partition $A = A^1 \cup A^2$, where A^1 and A^2 are disjoint nonempty subsets of A, then compute the barycenter x^j of A^j for $j = 1, 2$, and put $x = (x^1, x^2)$. According to Theorem 3.1 and Proposition 3.2, global solutions of (3.2) do exist and belong to the set of those points x. Hence, by symmetry, it is easy to see that the clustering problem in question has two global solutions: $\bar{x} = \big((\frac{1}{2}, 0), (\frac{1}{2}, 1)\big)$ and $\hat{x} = \big((0, \frac{1}{2}), (1, \frac{1}{2})\big)$. As $f(\bar{x}) = f(\hat{x}) = \frac{1}{4}$, the four centroid systems in $\bar{A}_6$ is a global solution of (3.2) are all local, non-global solutions of (3.2). Similarly, the four centroid systems in $\widetilde{A}_6 = \widetilde{A}_5 = \widetilde{A}_4$ are all local, non-global solutions of (3.2).

Remark 4.2 In both Algorithm 4.1 and Algorithm 4.2, one starts with $\bar{x}^1 = a^0$, where $a^0 = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m} a^i$ is the barycenter of the data set A. As it has been shown in Remark 4.1, for the clustering problem in Example 4.2 and for arbitrarily chosen control parameters $\gamma_1 \in [0, 1],\ \gamma_2 \in [0, 1],\ \gamma_3 \in [1, \infty)$, Algorithm 4.1 (resp., Algorithm 4.2) yields a local, non-global solution of (3.2). Anyway, if one starts with a data point a^i, $i \in I$, then by Algorithm 4.1 (resp., Algorithm 4.2) one can find a global solution of (3.2).

To proceed furthermore, we need the next lemma.

Lemma 4.1 *Let $x = (x^1, \ldots, x^k) \in \mathbb{R}^{r\times k}$ be a centroid system, where the centroids $x^1, \ldots, x^k$ are pairwise distinct. Then, after one step of applying the k-means Algorithm, one gets a new centroid system $\widetilde{x} = (\widetilde{x}^1, \ldots, \widetilde{x}^k)$ with pairwise distinct centroids, i.e., $\widetilde{x}^{j_1} \neq \widetilde{x}^{j_2}$ for any $j_1, j_2 \in J$ with $j_1 \neq j_2$.*

Proof. Let us denote by $\{A^1, \ldots, A^k\}$ the natural clustering associated with $x = (x^1, \ldots, x^k)$. For each $j \in J$, if $A^j \neq \emptyset$ then the centroid x^j is updated by the rule (3.9), and x^j does not change otherwise. This means that

$$\widetilde{x}^j = \frac{1}{|I(A^j)|} \sum_{i\in I(A^j)} a^i \tag{4.26}$$

if $A^j \neq \emptyset$, where $I(A^j) = \{i \in I \mid a^i \in A^j\}$, and $\widetilde{x}^j = x^j$ if $A^j = \emptyset$. Now, suppose that $j_1, j_2 \in J$ are such that $j_1 \neq j_2$. We may assume that $j_1 < j_2$.

Let $y^0 := \frac{1}{2}(x^{j_2} + x^{j_1})$ and $L := \{y \in \mathbb{R}^n \mid \langle y - y^0, x^{j_2} - x^{j_1} \rangle = 0\}$. Then, any point $y \in \mathbb{R}^n$ having equal distances to x^{j_1} and x^{j_2} lies in L. Denote by P_1 (resp., P_2) the open half-space with the boundary L that contains x^{j_1} (resp., x^{j_2}).

If the clusters A^{j_1} and A^{j_2} are both nonempty, then $\widetilde{x}^{j_1}$ and $\widetilde{x}^{j_2}$ are defined by formula (4.26). Since $\{A^1, \ldots, A^k\}$ is the natural clustering associated with the centroid system $x = (x^1, \ldots, x^k)$ and $j_1 < j_2$, one must have $A^{j_1} \subset \bar{P}_1$, where $\bar{P}_1 := P_1 \cup L$ is the closure of P_1, while $A^{j_2} \subset P_2$. The formulas

$$\widetilde{x}^{j_1} = \frac{1}{|I(A^{j_1})|} \sum_{i \in I(A^{j_1})} a^i, \quad \widetilde{x}^{j_2} = \frac{1}{|I(A^{j_2})|} \sum_{i \in I(A^{j_2})} a^i$$

show that $\widetilde{x}^{j_1}$ (resp., $\widetilde{x}^{j_2}$) is a convex combination of the points from A^{j_1} (resp., A^{j_2}). Hence, by the convexity of $\bar{P}_1$ (resp., P_2), we have $\widetilde{x}^{j_1} \in \bar{P}_1$ (resp., $\widetilde{x}^{j_2} \in P_2$). Then, the property $\widetilde{x}^{j_1} \neq \widetilde{x}^{j_2}$ follows from the fact that $\bar{P}_1 \cap P_2 = \emptyset$.

If the clusters A^{j_1} and A^{j_2} are both empty, then $\widetilde{x}^{j_1} = x^{j_1}$ and $\widetilde{x}^{j_2} = x^{j_2}$. Since $x^1, \ldots, x^k$ are pairwise distinct, we have $\widetilde{x}^{j_1} \neq \widetilde{x}^{j_2}$.

If $A^{j_1} \neq \emptyset$ and $A^{j_2} = \emptyset$, then $\widetilde{x}^{j_1} \in \bar{P}_1$ and $\widetilde{x}^{j_2} = x^{j_2} \in P_2$. Since $\bar{P}_1 \cap P_2 = \emptyset$, one must have $\widetilde{x}^{j_1} \neq \widetilde{x}^{j_2}$. The situation $A^{j_1} = \emptyset$ and $A^{j_2} \neq \emptyset$ is treated similarly.

The proof is complete. □

Remarkable properties of Algorithm 4.2 are described in forthcoming theorems, where the following assumption is used:

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

Note that, given any data set, one can apply the trick suggested in Remark 3.1 to obtain a data set satisfying **(C2)**.

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* (4.8)*, which is obtained by Procedure 4.2, are also pairwise distinct.*

Proof. Since $1 \leq \ell \leq k-1$, $k \leq m$, and data points $a^1, ..., a^m$ in the given data set A are pairwise distinct, one can find a data point $a^{i_0} \in A$, which is not

contained in the set $\{\bar{x}^1, ..., \bar{x}^\ell\}$. Then the set Y_1 defined by (4.7) is nonempty, because $d_\ell(a^{i_0}) > 0$ (hence the open ball $B\big(a^{i_0}, d_\ell(a^{i_0})\big)$ is nonempty). Moreover, $A \cap Y_1 \neq \emptyset$. So, from (4.12) and (4.13) it follows that $\bar{A}_1 \neq \emptyset$. Then, one easily deduces from (4.14)–(4.16) that the sets $\bar{A}_2$ and $\bar{A}_3$ are nonempty.

By the construction (4.7) of Y_1, one has $Y_1 \cap \{\bar{x}^1, ..., \bar{x}^\ell\} = \emptyset$. It follows that $\bar{A}_1 \cap \{\bar{x}^1, ..., \bar{x}^\ell\} = \emptyset$. (Actually, this property has been noted before.) Since

$$z_{\ell+1}(c(a)) \geq z_{\ell+1}(a) > 0 \quad \forall a \in \bar{A}_1,$$

and $z_{\ell+1}(\bar{x}^j) = 0$ for every $j \in \{1, \ldots, \ell\}$, we have $\bar{A}_2 \cap \{\bar{x}^1, ..., \bar{x}^\ell\} = \emptyset$. As $\bar{A}_3 \subset \bar{A}_2$, one sees that $\bar{A}_3 \cap \{\bar{x}^1, ..., \bar{x}^\ell\} = \emptyset$. Consequently, by (4.17), the centroids in any centroid system $(\bar{x}^1, ..., \bar{x}^\ell, c) \in \Omega$ are pairwise distinct. Since the approximate solution $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ of (4.8) is obtained from one centroid system $(\bar{x}^1, ..., \bar{x}^\ell, c) \in \Omega$ after applying finitely many steps of KM, thanks to Lemma 4.1 we can assert that the centroids $\hat{x}^1, \ldots, \hat{x}^{\ell+1}$ are 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.*

Proof. Algorithm 4.2 starts with computing the barycenter $a^0 = \dfrac{1}{m}\sum_{i=1}^{m} a^i$ of the data set A, put $\bar{x}^1 = a^0$, and set $\ell = 1$. Then, one applies Procedure 4.2 to find an approximate solution $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ of (4.8). By Theorem 4.1, the centroids $\hat{x}^1, ..., \hat{x}^{\ell+1}$ are pairwise distinct. Since Procedure 4.2 ends at Step 7 by picking any point $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ from the set $\widetilde{A}_5$, which is defined by (4.22), Theorem 4.1 assures that every centroid system $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ in $\widetilde{A}_5$ consists of pairwise distinct centroids.

In Step 4 of Algorithm 4.2, after putting $\bar{x}^j = \hat{x}^j$ for $j = 1, \ldots, \ell + 1$, one sets $\ell = \ell + 1$ and goes to Step 2. If $\ell < k$, then the computation continues, and one gets a approximate solution $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ of (4.8) with $\hat{x}^1, \ldots, \hat{x}^{\ell+1}$ being pairwise distinct by Theorem 4.1. If $\ell = k$, then the computation terminates by selecting an element $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k)$ from the set $\widetilde{A}_6$, which is defined by (4.23). Since $\widetilde{A}_6 \subset \widetilde{A}_5$ and we have shown that every centroid system in $\widetilde{A}_5$ consists of pairwise distinct centroids, the obtained centroids $\bar{x}^1, \ldots, \bar{x}^k$ are pairwise distinct. □

On one hand, if $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k)$ is global solution of (3.2), then by Propo-

sition 3.3 we know that the centroids $\bar{x}^1, \ldots, \bar{x}^k$ are pairwise distinct. More general, by Definition 3.2, the components of any nontrivial local solution are pairwise distinct. On the other hand, according to Theorem 4.2, Algorithm 4.2 yields a centroid system having pairwise distinct components. Thus, *the centroid system resulted from Algorithm 4.2 is a very good candidate for being a nontrivial local solution of* (3.2). As the global solutions are among the nontrivial local solutions, *Theorem 4.2 reveals a nice feature of Algorithm 4.2.*

4.2.4 The ε-neighborhoods technique

The ε-neighborhoods technique [12, pp. 869–870] (see also [71, pp. 352–353]) allows one to reduce the computation volume of Algorithm 4.1 (as well as that of Algorithm 4.2, or another incremental clustering algorithm based on the sets $\bar{A}_1$), when it is applied to large data sets. The procedure of removing data points from A to get a smaller set $\bar{A}_1$ is as follows. Choose a sufficiently small number $\delta \in (0, \ell^{-1})$ (for example, $\delta = \min\{10^{-3}, \ell^{-1}\}$). In the notations of Subsection 4.2.1, let $\{A^1, \ldots, A^\ell\}$ be the natural clustering associated with the centroid system $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$. For every $j \in \{1, \ldots, \ell\}$, if $A^j \neq \emptyset$, then one defines

$$\alpha_j = \frac{1}{|A^j|} \sum_{a \in A^j} \|\bar{x}^j - a\|^2$$

and $\beta_j = \max\{\|\bar{x}^j - a\|^2 \mid a \in A^j\}$. Set $\mu_j = \dfrac{\beta_j}{\alpha_j}$ and observe that $\mu_j \geq 1$. Let

$$A^j_\delta := \{a \in A^j \mid \|\bar{x}^j - a\|^2 \geq \eta_j \alpha_j\},$$

where $\eta_j = 1 + \ell\delta(\mu_j - 1)$. One has $A^j_\delta \neq \emptyset$. Indeed, if $\bar{a} \in A^j$ is such a data point that $\|\bar{x}^j - \bar{a}\|^2 = \beta_j$, then $\|\bar{x}^j - a\|^2 = \beta_j \geq \eta_j \alpha_j$; hence $\bar{a} \in A^j_\delta$. To proceed furthermore, denote by A_δ the union of all the sets A^j_δ, where $j \in \{1, \ldots, \ell\}$ is such that $A^j \neq \emptyset$. Now, instead of $\bar{A}_1$ given by (4.13), we use the set

$$\bar{A}_{1,\delta} := \{a \in A_\delta \cap Y_1 \mid z_{\ell+1}(a) \geq \eta_1 z^1_{\max}\}, \tag{4.27}$$

which is a subset of $\bar{A}_1$. In the construction $\bar{A}_{1,\delta}$ by (4.27), we have removed from A all the data points a with $\|\bar{x}^j - a\|^2 < \eta_j \alpha_j$, where $j \in \{1, \ldots, \ell\}$ is such that $A^j \neq \emptyset$.

4.3 Incremental DC Clustering Algorithms

Some incremental clustering algorithms based on Ordin-Bagirov's clustering algorithm and the DCA [77] 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 [7] suggested an improvement of Algorithm 4.1 by using DCA (see [51, 60, 77]) twice at each clustering level $\ell \in \{1, \ldots, k\}$. First, let us recall the specific DCA scheme presented in [7, p. 6]. Consider a DC program of the form

$$\min \{\varphi(x) := g(x) - h(x) \mid x \in \mathbb{R}^n\}, \tag{4.28}$$

where g, h are continuous convex functions on $\mathbb{R}^n$. It is assumed that g is differentiable. Then, one has $\partial g(x) = \{\nabla g(x)\}$ for every $x \in \mathbb{R}^n$. If $\bar{x} \in \mathbb{R}^n$ is a *local solution* of (4.28), then by the necessary optimality condition in DC programming (see, e.g., [77] and [31]) one has $\partial h(\bar{x}) \subset \partial g(\bar{x})$. The latter is equivalent to saying that $\partial h(\bar{x})$ is a singleton and the unique element of $\partial h(\bar{x})$, denoted by $\bar{y}$, satisfies the condition $\bar{y} = \nabla g(\bar{x})$. If $\bar{x} \in \mathbb{R}^n$ is such that $\partial h(\bar{x})$ is a singleton and the unique element $\bar{y}$ of $\partial h(\bar{x})$ satisfies the last equality, then $\bar{x}$ is said to be a *stationary point* of (4.28). If $\bar{x} \in \mathbb{R}^n$ is such that $\nabla g(\bar{x}) \in \partial h(\bar{x})$, then $\bar{x}$ is said to be a *critical point* of (4.28). Obviously, a stationary point is a critical point. Note that (4.28) may possess some critical points which are not stationary points. The just mentioned necessary condition for local minimizers of (4.28) is the motivation for the *stopping criterion* in the second step of the next procedure.

Procedure 4.3 (A specific DCA scheme [7, p. 6])

Input: *A starting point* $x^1 \in \mathbb{R}^n$.
Output: *An approximate solution* x^p *of* (4.28).
Step 1. Select any starting point $x^1 \in \mathbb{R}^n$ *and set* $p = 1$.
Step 2. Compute $y^p \in \partial h(x^p)$.
Step 3. If $y^p = \nabla g(x^p)$, *then stop.*

Step 4. Find a solution x^{p+1} of the convex optimization problem

$$\min\{g(x) - \langle y^p, x\rangle \mid x \in \mathbb{R}^n\}. \tag{4.29}$$

Step 5. Set $p = p + 1$ and go to Step 2.

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.

Procedure 4.4 (A modified version of Procedure 4.3)

Input: *A starting point $x^1 \in \mathbb{R}^n$.*
Output: *An approximate solution x^{p+1} of (4.28).*
Step 1. Select any starting point $x^1 \in \mathbb{R}^n$, a tolerance $\varepsilon > 0$, and set $p := 1$.
Step 2. Compute $y^p \in \partial h(x^p)$.
Step 3. Find a solution x^{p+1} of the convex optimization problem (4.29).
Step 4. If $\|x^{p+1} - x^p\| \leq \varepsilon$, then stop.
Step 5. Set $p = p + 1$ and go to Step 2.

Now we turn our attention back to problem (4.4) 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 (4.5) and (4.6). Clearly,

$$\partial g^1(y) = \{\nabla g^1(y)\} = \Big\{\frac{2}{m}\sum_{i\in I}(y - a^i)\Big\}. \tag{4.30}$$

To compute the subdifferential of $g^2(.)$ at $y \in \mathbb{R}^n$, consider the sets $A_1(y)$ and $A_2(y)$ defined in (4.9) and (4.2.1). Let

$$A_3(y) := \{a^i \in A \mid \|y - a^i\|^2 > d_\ell(a^i)\}. \tag{4.31}$$

Set $A_4(y) = A_2(y) \setminus A_3(y)$ and observe that

$$A_4(y) = \{a^i \in A \mid \|y - a^i\|^2 = d_\ell(a^i)\}. \tag{4.32}$$

Taking account of (4.6), (4.9), (4.31), and (4.32), one has

$$g^2(y) = \frac{1}{m}\Big(\sum_{a^i \in A_1(y)} d_\ell(a^i) + \sum_{a^i \in A_3(y)} \|y - a^i\|^2 + \sum_{a^i \in A_4(y)} \max\big\{d_\ell(a^i), \|y - a^i\|^2\big\}\Big).$$

Thus,

$$\partial g^2(y) = \frac{2}{m}\Big(\sum_{a^i \in A_3(y)} (y - a^i) + \sum_{a^i \in A_4(y)} \mathrm{co}\{0, y - a^i\}\Big). \tag{4.33}$$

(If $A_4(y) = \emptyset$, then the second sum in (4.33) is absent.) If y is a *local solution* of (4.4), then we have $\partial g^2(y) \subset \partial g^1(y)$. Since $\partial g^1(y)$ is a singleton by (4.30), the last inclusion is fulfilled only if $\partial g^2(y)$ is singleton. Hence, from (4.33) it follows that $a^i = y$ whenever $a^i \in A_4(y)$. This means that either $A_4(y) = \emptyset$, or $y \in A$ and $A_4(y) = \{y\}$. So,

$$\partial g^2(y) = \Big\{\frac{2}{m} \sum_{a^i \in A_3(y)} (y - a^i)\Big\}.$$

Therefore, the inclusion $\partial g^2(y) \subset \partial g^1(y)$ is fulfilled if and only if either

$$\begin{cases} A_4(y) = \emptyset \\ y = |A_1(y)|^{-1} \displaystyle\sum_{a^i \in A_1(y)} a^i, \end{cases}$$

or

$$\begin{cases} y \in A,\ A_4(y) = \{y\} \\ y = (|A_1(y)| + 1)^{-1}\Big(\displaystyle\sum_{a^i \in A_1(y)} a^i + y\Big), \end{cases}$$

where $|\Omega|$ denotes the number of elements of a set Ω. For $\varphi := g$, $g := g^1$, and $h := g^2$, our problem (4.4) has the form (4.28). Thus, both Procedures 3a and 3b can be used to solve (4.4). Thanks to (4.33), one has

$$\frac{2}{m} \sum_{a^i \in A_3(y)} (y - a^i) \in \partial g^2(y) \quad (\forall y \in \mathbb{R}^n).$$

In particular, for any given vector $x^p \in \mathbb{R}^n$, $\dfrac{2}{m} \displaystyle\sum_{a^i \in A_3(x^p)} (x^p - a^i) \in \partial g^2(x^p)$. So, vector y^p in Step 2 of Procedures 4.3 and 4.4 can be chosen as

$$y^p = \frac{2}{m} \sum_{a^i \in A_3(x^p)} (x^p - a^i). \tag{4.34}$$

In Step 4 of Procedure 4.3 (resp., Step 3 of Procedure 4.4), one has to solve the differentiable convex program

$$\min\big\{\psi(x) := g^1(x) - \langle y^p, x\rangle \mid x \in \mathbb{R}^n\big\}. \tag{4.35}$$

From (4.5) and (4.34) it follows that

$$\psi(x) = \frac{1}{m}\sum_{a^i\in A} d_\ell(a^i) + \frac{1}{m}\sum_{a^i\in A}\|x-a^i\|^2 - \frac{2}{m}\Big\langle \sum_{a^i\in A_3(x^p)}(x^p-a^i), x\Big\rangle$$

By the Fermat Rule, $x \in \mathbb{R}^n$ solves (4.35) if and only if $\nabla\psi(x) = 0$. This condition can be rewritten equivalently as

$$\begin{aligned}
&\frac{2}{m}\sum_{a^i\in A}(x-a^i) - \frac{2}{m}\sum_{a^i\in A_3(x^p)}(x^p-a^i) = 0\\
\iff\ & mx - \sum_{a^i\in A_1(x^p)\cup A_4(x^p)} a^i - |A_3(x^p)|x^p = 0\\
\iff\ & x = \frac{1}{m}\Big(|A_3(x^p)|x^p + \sum_{a^i\in A_1(x^p)\cup A_4(x^p)} a^i\Big).
\end{aligned}$$

Consequently, (4.35) has the unique solution

$$x^{p+1} = \frac{1}{m}\Big(|A_3(x^p)|x^p + \sum_{a^i\in A_1(x^p)\cup A_4(x^p)} a^i\Big). \tag{4.36}$$

To solve (4.4) by a DCA, Bagirov [7, p. 7] suggests to use the stopping criterion $y^p = \nabla g^1(x^p)$, where y^p is given by (4.34). Since $\nabla g^1(x^p) = \dfrac{2}{m}\sum_{i\in I}(x^p - a^i)$ by (4.30) and $A = A_1(x^p)\cup A_3(x^p)\cup A_4(x^p)$, one has $y^p = \nabla g^1(x^p)$ if and only if $\sum_{a^i\in\Omega_p}(x^p-a^i) = 0$, where $\Omega_p := A_1(x^p)\cup A_4(x^p)$. It follows that

$$x^p = \frac{1}{|\Omega_p|}\sum_{a^i\in\Omega_p} a^i.$$

Thus, x^p is the barycenter of Ω_p. By the necessary optimality condition in DC programming [77], one obtains $\partial h(x^p) \subset \partial g(x^p)$, where $g = g^1$ and $h = g^2$ are respectively given by (4.5) and (4.6). Since $\partial g^1(x^p)$ is a singleton, $\partial g^2(x^p)$ is also a singleton. As $y^p \in \partial g^1(x^p)$, one can compute y^p by formula (4.34).

The iteration formula (4.36) shows that, applied to problem (4.4) with the DC decomposition $g(y) = g^1(y) - g^2(y)$, Procedures 3a and 3b have the next simplified formulations.

Procedure 4.5 (A DCA scheme for solving (4.4); see [7, p. 7])

Input: *A starting point* $x^1 \in \mathbb{R}^n$.
Output: *An approximate solution* x^p *of* (4.4).
Step 1. Select any starting point $x^1 \in \mathbb{R}^n$ *and set* $p = 1$.
Step 2. Compute the numbers $d_\ell(a^i)$, $i \in I$, *by formula* (4.2).
Step 3. Compute the sets $A_1(x^p)$, $A_3(x^p)$, *and* $A_4(x^p)$ *by using* (4.9), (4.31), *and* (4.32), *respectively.*
Step 4. Compute y^p *by* (4.34).
Step 5. If $y^p = \nabla g^1(x^p)$, *i.e.,* x^p *is the barycenter of* $\Omega_p := A_1(x^p) \cup A_4(x^p)$, *then stop.*
Step 6. Compute x^{p+1} *by formula* (4.36).
Step 7. Set $p = p + 1$ *and go to Step 2.*

Procedure 4.6 (A modified version of Procedure 4.5)

Input: *A starting point* $x^1 \in \mathbb{R}^n$.
Output: *An approximate solution* x^{p+1} *of* (4.4).
Step 1. Select any starting point $x^1 \in \mathbb{R}^n$, *a tolerance* $\varepsilon \geq 0$, *and set* $p = 1$.
Step 2. Compute $d_\ell(a^i)$, $i \in I$, *by formula* (4.2).
Step 3. Compute $A_1(x^p)$, $A_3(x^p)$, *and* $A_4(x^p)$ *by using* (4.9), (4.31), *and* (4.32), *respectively.*
Step 4. Compute x^{p+1} *by formula* (4.36).
Step 5. If $\|x^{p+1} - x^p\| \leq \varepsilon$, *then stop.*
Step 6. Set $p = p + 1$ *and go to Step 2.*

The following natural questions arise:

(Q1) *Whether the computation in* Procedure 4.5 (*resp., in* Procedure 4.6) *terminates after finitely many steps?*

(Q2) *If the computation in* Procedure 4.5 (*resp., in* Procedure 4.6 *with a tolerance* $\varepsilon = 0$) *does not terminate after finitely many steps, then the iteration sequence* $\{x^p\}$ *converges to a stationary point of* (4.35)*?*

Partial answers to (Q1) and (Q2) are given in the forthcoming theorem.

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* $\bar{x} \in \mathcal{B}$

Proof. (i) To prove this assertion, it suffices to construct a suitable example, where the computation by Procedure 4.5 does not terminate after finitely many steps. Choose $n = 2$, $m = 3$, $k = 2$, $A = \{a^1, a^2, a^3\}$, where $a^1 = (0, 0)$, $a^2 = (1, 0)$, $a^3 = (0, 1)$. The barycenter of A is $a^0 = (\frac{1}{3}, \frac{1}{3})$. Let $\ell = 1$ and $\bar{x}^1 = a^0$. To solve (4.4) by Procedure 4.5, we select $x^1 = (0, \frac{5}{4})$ and set $p = 1$. From (4.9), (4.31) and (4.32) it follows that $A_1(x^1) = \{a^3\}$, $A_3(x^1) = \{a^1, a^2\}$, and $A_4(x^1) = \emptyset$. By induction, from (4.36) one deduces that $A_1(x^p) = \{a^3\}$, $A_3(x^p) = \{a^1, a^2\}$, $A_4(x^p) = \emptyset$ for every $p \geq 1$, and

$$\begin{cases} x_1^{p+1} = \frac{2}{3} x_1^p \quad \forall p \geq 1 \\ x_2^{p+1} = \frac{2}{3} x_2^p + \frac{1}{3} \quad \forall p \geq 1, \end{cases} \tag{4.37}$$

where $x^{p+1} = (x_1^{p+1}, x_2^{p+1})$. In accordance with (4.37), if $x_1^p = 0$, then $x_1^{p+1} = 0$; and if $x_2^p > 1$, then $x_2^{p+1} > 1$. Hence, the DCA sequence $\{x^p\}$ generated by Procedure 4.5 converges to $\bar{x} = (0, 1)$. However, the computation does not terminate at any step p, because the stopping criterion in Step 5 (which requires that x^p is the barycenter of $\Omega_p = A_1(x^p) \cup A_4(x^p) = \{a^3\}$) is not satisfied.

(ii) To show that the computation by Procedure 4.6 with $\varepsilon = 0$ may not terminate after finitely many steps, we consider the above two-dimensional clustering problem. Choose $\ell = 1$ and $\bar{x}^1 = a^0$. To solve (4.4) by Procedure 4.6, again we select $x^1 = (0, \frac{5}{4})$ and set $p = 1$. Clearly, from (4.36)

one obtains $A_1(x^p) = \{a^3\}$, $A_3(x^p) = \{a^1, a^2\}$, $A_4(x^p) = \emptyset$ for every $p \geq 1$, and the iteration formula (4.37). Thus, the DCA sequence $\{x^p\}$ generated by Procedure 4.6 converges to $\bar{x} = (0, 1)$. But, the computation does not terminate at any step p as the stopping criterion in Step 5 (which requires that $\|x^{p+1} - x^p\| \leq \varepsilon = 0$) is not satisfied.

(iii) Fix any $\varepsilon > 0$. Let $\{x^k\}$ be a sequence generated by Procedure 4.6. If the sequence $\{x^k\}$ is finite, then we are done. Suppose that the sequence $\{x^k\}$ is infinite. To obtain a contradiction, consider the auxiliary problem (4.4) with $g = g^1 - g^2$, where g^1 and g^2 are given respectively by (4.5) and (4.6). By the Weierstrass theorem, the problem of minimizing $g(y)$ on the topological closure of Y_1, where the latter is defined by (4.7), has a solution $\bar{y}$. Since $Y_1 \neq \emptyset$, $g(y) < \dfrac{1}{m}\sum\limits_{i=1}^{m} d_\ell(a^i)$ for all $y \in Y_1$, $g(y) = \dfrac{1}{m}\sum\limits_{i=1}^{m} d_\ell(a^i)$ for all $y \in Y_2$, $\bar{y} \in Y_1$ and $\bar{y}$ is a global solution of (4.4). Thus, $\alpha := \min\{g(y) \mid y \in \mathbb{R}^n\}$ is well defined. By (4.3) one has $\alpha \geq 0$. Denote by $\rho(g^i)$ the *modulus of strong convexity* [77, p. 8] of g^i on $\mathbb{R}^n$ for $i = 1, 2$. By (4.5) one has $\rho(g^1) > 0$, i.e., g^1 is strongly convex on $\mathbb{R}^n$. So, $\rho(g^1) + \rho(g^2) > 0$. Therefore, invoking the assertion (iii) of Theorem 3 in [77] (see also [79, Theorem 3.7]), we obtain $\lim\limits_{p\to\infty}(x^{p+1} - x^p) = 0$. In particular, there exists $p \in \mathbb{N}$ such that $\|x^{p+1} - x^p\| \leq \varepsilon$. This means that the computation by Procedure 4.6 cannot continue after step p. We have thus arrived at a contradiction.

(iv) We put $\Omega_p = A_1(x^p) \cup A_4(x^p)$. Suppose that the sequence $\{x^k\}$ is finite, i.e., the computation terminates at a step $p \in \mathbb{N}$. Since x^{p+1} is computed via x^p by (4.36) and $x^{p+1} = x^p$, we have $(m - |A_3(x^p)|)x^{p+1} = \sum\limits_{a^i \in \Omega_p} a^i$. As $m - |A_3(x^p)| = |\Omega_p|$, the last equality implies that x^{p+1} is the barycenter of Ω_p. This justifies our claim.

(v) Let Ω_p be as above. Suppose that the sequence $\{x^p\}$ is infinite. It follows from (4.36) that

$$x^{p+1} = \frac{1}{m}\Big(|A \setminus \Omega_p| x^p + \sum_{a^i \in \Omega_p} a^i\Big). \tag{4.38}$$

Hence, $x^{p+1} \in \mathrm{co}\big(A \cup \{x^p\}\big)$ for all $p \in \mathbb{N}$. Therefore, by induction one obtains $x^p \in \mathrm{co}\big(A \cup \{x^1\}\big)$ for all $p \in \mathbb{N}$. In particular, *the sequence* $\{x^p\}$ *is bounded*. So, there exists subsequence $\{x^{p'}\}$ of $\{x^p\}$, which converges to a point $\bar{x} \in \mathbb{R}^n$.

We have $\bar{x} \in \mathcal{B}$. Indeed, by the Dirichlet principle we can extract a subsequence $\{x^{p''}\}$ of $\{x^{p'}\}$ such that the sets $A_1(x^{p''})$, $A_3(x^{p''})$, and $A_4(x^{p''})$ are

stable in the sense that there exist disjoint subsets A_1, A_3, and A_4 of A satisfying $A_1(x^{p''}) = A_1$, $A_3(x^{p''}) = A_3$, and $A_4(x^{p''}) = A_4$ for each index p''. Let $\Omega := A_1 \cup A_4$. By (4.38), one has

$$x^{p''+1} = \frac{1}{m}\Big(|A_3|x^{p''} + \sum_{a^i \in \Omega} a^i\Big). \tag{4.39}$$

Since $\lim\limits_{p' \to \infty} x^{p'} = \bar{x}$, passing (4.39) to the limit as $p'' \to \infty$ yields

$$\bar{x} = \frac{1}{m}(|A_3|\bar{x} + \sum_{a^i \in \Omega} a^i).$$

It follows that $\bar{x} = b_\Omega$. We have thus proved that $\bar{x} \in \mathcal{B}$.

To complete the proof, it suffices to show that $\lim\limits_{p \to \infty} x^p = \bar{x}$. Let $\varepsilon_0 > 0$ be the minimum of the set consisting of the numbers $\|b_{\Omega_1} - b_{\Omega_2}\|$, where Ω_1 and Ω_2 are nonempty subsets of A with $b_{\Omega_1} \neq b_{\Omega_2}$. Then, for any nonempty subsets Ω_1 and Ω_2 of A with $b_{\Omega_1} \neq b_{\Omega_2}$, one has $B(b_{\Omega_1}, \frac{1}{4}\varepsilon_0) \cap B(b_{\Omega_2}, \frac{1}{4}\varepsilon_0) = \emptyset$. Put $V = \bigcup\limits_{b \in \mathcal{B}} B(b, \frac{1}{4}\varepsilon_0)$ and observe that $\mathbb{R}^n \setminus V$ is closed. One must have $x^p \in V$ for all p large enough. Indeed, if this is not the case then, by the boundedness of $\{x^p\}$, one can find a subsequence $\{x^{p_j}\}$ of $\{x^p\}$ such that $\{x^{p_j}\} \subset \mathbb{R}^n \setminus V$ and $x^{p_j} \to \bar{b}$ as $p_j \to \infty$. Repeating the arguments which have been applied to the above subsequence $\{x^{p'}\}$ of $\{x^p\}$, we can show that $\bar{b} \in \mathcal{B}$. Then, on one hand we have $\bar{b} \in V$. On the other hand, as $\{x^{p_j}\} \subset \mathbb{R}^n \setminus V$, the inclusion $\bar{b} \in \mathbb{R}^n \setminus V$ is valid. We have arrived at a contradiction.

Let $\bar{p} \in \mathbb{N}$ be such that one has $x^p \in V$ for all $p \geq \bar{p}$. By the equality $\lim\limits_{p \to \infty}(x^{p+1} - x^p) = 0$, which has been established in the proof of the assertion (iii), there is $\hat{p} \geq \bar{p}$ such that

$$\|x^{p+1} - x^p\| \leq \frac{1}{4}\varepsilon_0 \qquad \forall p \geq \hat{p}. \tag{4.40}$$

As $\lim\limits_{p' \to \infty} x^{p'} = \bar{x}$, there exists $p' \geq \hat{p}$ such that

$$x^{p'} \in B(\bar{x}, \frac{1}{4}\varepsilon_0). \tag{4.41}$$

By (4.40), one has $\|x^{p'+1} - x^{p'}\| \leq \frac{1}{4}\varepsilon_0$. Since $x^{p'+1} \in V$, there exits $b \in \mathcal{B}$ such that $x^{p'+1} \in B(b, \frac{1}{4}\varepsilon_0)$. If $b \neq \bar{x}$, then the definition of ε_0 implies that

$$\|b - \bar{x}\| \geq \varepsilon_0. \tag{4.42}$$

Thanks to (4.40) and (4.41), we have

$$\|b - \bar{x}\| \leq \|b - x^{p'+1}\| + \|x^{p'+1} - x^{p'}\| + \|x^{p'} - \bar{x}\| \leq \frac{3}{4}\varepsilon_0.$$

This contradicts (4.42). Thus, $b = \bar{x}$. It follows that $x^{p'+1} \in B(\bar{x}, \frac{1}{4}\varepsilon_0)$.

Letting $x^{p'+1}$ play the role of $x^{p'}$ in the inclusion (4.41), by the above argument we obtain $x^{p'+2} \in B(\bar{x}, \frac{1}{4}\varepsilon_0)$, and so on. Therefore, $\{x^p\} \subset B(\bar{x}, \frac{1}{4}\varepsilon_0)$ for all $p \geq p'$. Hence, any cluster point of $\{x^p\}$ must belong to both sets $\bar{B}(\bar{x}, \frac{1}{4}\varepsilon_0)$ and $\mathcal{B}$. Since $\mathcal{B} \cap \bar{B}(\bar{x}, \frac{1}{4}\varepsilon_0) = \bar{x}$, we conclude that $\lim\limits_{p\to\infty} x^p = \bar{x}$. □

Concerning the property (iv) in Theorem 4.3, we want to know at which *convergence rate* the DCA sequence, provided that it is infinite, converges to the limit point. Recall that the definitions of two types of linear convergence of vectors sequences were given in Definitions 1.9 and 1.10 in Section 1.4.

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*

$$\|x^{p+1} - \bar{x}\| \leq \frac{m-1}{m}\|x^p - \bar{x}\| \tag{4.43}$$

for all p *sufficiently large.*

Proof. By our assumption and by assertion (iv) of Theorem 4.3, $\{x^p\}$ converges to a point $\bar{x} \in \mathcal{B}$. Suppose that $\{x^{p'}\}$ is *any* subsequence of $\{x^p\}$ such that the sets $A_1(x^{p'})$, $A_3(x^{p'})$, and $A_4(x^{p'})$ are stable, i.e., there exist disjoint subsets $\widetilde{A}_1$, $\widetilde{A}_3$, and $\widetilde{A}_4$ of A satisfying $A_1(x^{p'}) = \widetilde{A}_1$, $A_3(x^{p'}) = \widetilde{A}_3$, and $A_4(x^{p'}) = \widetilde{A}_4$ for every index p'. Let $\widetilde{\Omega} := \widetilde{A}_1 \cup \widetilde{A}_4$. By (4.38), one has

$$x^{p'+1} = \frac{1}{m}\Big(|\widetilde{A}_3|x^{p'} + \sum_{a^i \in \widetilde{\Omega}} a^i\Big). \tag{4.44}$$

If $|\widetilde{A}_3| = m$, then $\widetilde{\Omega} = \emptyset$. So, from (4.44) it follows that $x^{p'+1} = x^{p'}$; then the computation by Procedure 4.6 stops at step p'. This contradicts our assumption that the latter yields the infinite sequence $\{x^p\}$. Thus, setting $\bar{m} = |\widetilde{A}_3|$, one must have $\bar{m} \leq m - 1$. From (4.44) one can deduce that

$$mx^{p'+1} - m\bar{x} = \bar{m}x^{p'} - \bar{m}\bar{x} + \Big(\sum_{a^i \in \widetilde{\Omega}} a^i - |\widetilde{\Omega}|\bar{x}\Big). \tag{4.45}$$

Since $\lim\limits_{p\to\infty} x^p = \bar{x}$ and $\lim\limits_{p\to\infty}(x^{p+1} - x^p) = 0$, passing (4.44) to the limit as $p' \to \infty$, we get

$$\bar{x} = \frac{1}{m}(|\widetilde{A}_3|\bar{x} + \sum_{a^i \in \widetilde{\Omega}} a^i),$$

which implies that $|\widetilde{\Omega}|\bar{x} = \sum_{a^i \in \widetilde{\Omega}} a^i$. Obviously, this equality and (4.45) yield

$$\|x^{p'+1} - \bar{x}\| = \frac{\bar{m}}{m}\|x^{p'} - \bar{x}\|.$$

So, the inequality

$$\|x^{p'+1} - \bar{x}\| \leq \frac{m-1}{m}\|x^{p'} - \bar{x}\| \tag{4.46}$$

holds for every p'.

If (4.43) does not hold for all p sufficiently large, then there exists a subsequence of $\{x^p\}$ such that the inequality in (4.43) is violated for every member of that subsequence. Then, we can extract from the latter a subsequence, which is denoted by $\{x^{p'}\}$, such that the sets $A_1(x^{p'})$, $A_3(x^{p'})$, and $A_4(x^{p'})$ are stable. On one hand, the inequality (4.46) holds for every p' by the result of the first part of this proof. On the other hand, by the choice of this subsequence $\{x^{p'}\}$, we have $\|x^{p'+1} - \bar{x}\| > \frac{m-1}{m}\|x^{p'} - \bar{x}\|$. Thus, we have arrived at a contradiction. □

Remark 4.3 Select a constant C such that $\dfrac{m-1}{m} < C < 1$. By Theorem 4.4, if the computation is terminated at step p, provided that p is sufficiently large, then one has $\|x^p - \bar{x}\| \leq C\|x^{p-1} - \bar{x}\|$. Hence, the *computation error* between the obtained approximate solution x^p and the exact limit point $\bar{x}$ of the sequence $\{x^p\}$ is smaller than the number $C\|x^{p-1} - \bar{x}\|$. Since $\{x^p\}$ converges to $\bar{x}$, one sees that the *computation error bound* $C\|x^{p-1} - \bar{x}\|$ tends to 0 as $p \to \infty$.

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

$$f^1(x) := \frac{1}{m}\sum_{i \in I}\Big(\sum_{j \in J}\|a^i - x^j\|^2\Big) \tag{4.47}$$

and

$$f^2(x) := \frac{1}{m}\sum_{i \in I}\Big(\max_{j \in J}\sum_{q \in J\setminus\{j\}}\|a^i - x^q\|^2\Big). \tag{4.48}$$

By (4.47), one has $\partial f^1(x) = \{\nabla f^1(x)\} = \{2(x^1 - a^0, \ldots, x^k - a^0)\}$, where $a^0 = b_A$ is the barycenter of the system $\{a^1, \ldots, a^m\}$ (see [71] and Chapter 3).

Set $\varphi_i(x) = \max\limits_{j\in J} h_{i,j}(x)$ with $h_{i,j}(x) := \sum\limits_{q\in J\setminus\{j\}} \|a^i - x^q\|^2$ and $J_i(x)$ is given by (3.16). From (4.48) it follows that

$$\partial f^2(x) = \frac{1}{m}\sum_{i\in I} \partial\varphi_i(x) \tag{4.49}$$

with $\partial\varphi_i(x)$ being computed (see [71] and and Chapter 3) by the formula

$$\partial\varphi_i(x) = \mathrm{co}\Big\{\nabla h_{i,j}(x)\,|\,j\in J_i(x)\Big\} = \mathrm{co}\Big\{2\Big(\widetilde{x}^j - \widetilde{a}^{i,j}\Big)\;\big|\;j\in J_i(x)\Big\},$$

where $\widetilde{x}^j = (x^1,\ldots,x^{j-1},0_{\mathbb{R}^n},x^{j+1},\ldots,x^k)$ and

$$\widetilde{a}^{i,j} = \Big(a^i,\ldots,a^i,\underbrace{0_{\mathbb{R}^n}}_{j-\text{th position}},a^i,\ldots,a^i\Big). \tag{4.50}$$

For $\varphi := f$, $g := f^1$, and $h := f^2$, our clustering problem (3.2) has the form (4.28). Thus, both Procedures 4.4 and 4.6 can be used to solve (3.2). Let $x^p = (x^{p,1},...,x^{p,k}) \in \mathbb{R}^{nk}$ be the centroid system at an iteration $p\in\mathbb{N}$, $\{A^1,\ldots,A^k\}$ be the natural clustering associated with x^p. Clearly, the vector y^p in Step 2 of Procedure 4.4 satisfies the inclusion $y^p \in \partial f^2(x^p)$. By (4.49), one has

$$\partial f^2(x^p) = \frac{1}{m}\sum_{i\in I} \partial\varphi_i(x^p) \tag{4.51}$$

with $\partial\varphi_i(x^p)$ being computed by (4.50), i.e.,

$$\partial\varphi_i(x^p) = \mathrm{co}\,\{\nabla h_{i,j}(x^p)\,|\,j\in J_i(x^p)\} = \mathrm{co}\,\big\{2\,(\widetilde{x}^{p,j} - \widetilde{a}^{i,j})\;|\,j\in J_i(x^p)\big\} \tag{4.52}$$

with $\widetilde{x}^{p,j} = (x^{p,1},\ldots,x^{p,j-1},0_{\mathbb{R}^n},x^{p,j+1},\ldots,x^{p,k})$ for all $j\in J$. Note that the index sets $J_i(x^p)$, $i\in I$, in (4.52) are computed by formula (3.16) and the vectors $\widetilde{a}^{i,j}$, with $i\in I$ and $j\in J$, are given by (4.50). For every $i\in I$, if we assign the data point a^i to the centroid $x^{p,j}$ of the centroid system $\big\{x^{p,1},...,x^{p,k}\big\}$ with the smallest index j, denoted by $j(i)$, such that one has $\|a^i - x^{p,j(i)}\|^2 = \min\limits_{q\in J}\|a^i - x^{p,q}\|^2$. Since

$$J_i(x) = \Big\{j\in J\;\big|\;\|a^i - x^j\|^2 = \min_{q\in J}\|a^i - x^q\|^2\Big\},$$

one has $j(i)\in J_i(x^p)$. So, $2\,(\widetilde{x}^{p,j(i)} - \widetilde{a}^{i,j(i)}) \in \mathrm{co}\,\{2\,(\widetilde{x}^{p,j} - \widetilde{a}^{i,j})\;|\;j\in J_i(x^p)\}$. Hence, by (4.51) and (4.52),

$$\frac{2}{m}\sum_{i\in I}(\widetilde{x}^{p,j(i)} - \widetilde{a}^{i,j(i)}) \in \partial f^2(x^p). \tag{4.53}$$

The above assignment of the data point a^i, $i \in I$, to the centroid $x^{p,j(i)}$ corresponds to the the natural clustering for A on the basis of the the centroid system $\{x^{p,1}, ..., x^{p,k}\}$.

Let $\{A^{p,1}, \ldots, A^{p,k}\}$ be the natural clustering associated with the centroid system $x^p = (x^{p,1}, ..., x^{p,k}) \in \mathbb{R}^{nk}$. Thanks to (4.53), to have a vector $y^p \in \partial\varphi_i(x^p)$ one can choose $y^p = \dfrac{2}{m}\sum_{i\in I}\left(\widetilde{x}^{p,j(i)} - \widetilde{a}^{i,j(i)}\right)$. As observed by Bagirov [7, p. 7],

$$\begin{aligned} y^p &= \frac{2}{m}\Big(\sum_{a\in A\setminus A^{p,1}}(x^{p,1}-a), ..., \sum_{a\in A\setminus A^{p,k}}(x^{p,k}-a)\Big) \\ &= \frac{2}{m}\Big((m-\beta_{p,1})x^{p,1} - (ma^0 - \beta_{p,1}a^{0,p,1}), ..., (m-\beta_{p,k})x^{p,k} \\ &\quad -(ma^0 - \beta_{p,k}a^{0,p,k})\Big), \end{aligned} \tag{4.54}$$

where a^0 is the barycenter of A, $a^{0,p,j}$ is the barycenter of $A^{p,j}$, and $\beta_{p,j}$ is the number of elements in $A^{p,j}$ for every $j \in J$. In Step 4 of Procedure 4.3 (resp., Step 3 of Procedure 4.4), one solves the differentiable convex program

$$\min\left\{\phi(x) := f^1(x) - \langle y^p, x\rangle \mid x \in \mathbb{R}^{nk}\right\}. \tag{4.55}$$

From (4.47) and (4.54), one gets $\phi(x) = \dfrac{1}{m}\sum_{i\in I}\Big(\sum_{j\in J}\|a^i - x^j\|^2\Big) - \langle y^p, x\rangle$. By the Fermat Rule, $x^{p+1} \in \mathbb{R}^{nk}$ solves (4.55) if and only if $\nabla\phi(x^{p+1}) = 0$. By (4.54), this is equivalent to saying that the following holds for every $j \in J$:

$$\begin{aligned} &\frac{2}{m}\sum_{i\in I}(x^{p+1,j} - a^i) - \frac{2}{m}\Big((m-\beta_{p,j})x^{p,j} - (ma^0 - \beta_{p,j}a^{0,p,j})\Big) = 0 \\ &\iff mx^{p+1,j} - ma^0 - (m-\beta_{p,j})x^{p,j} + (ma^0 - \beta_{p,j}a^{0,p,j}) = 0. \end{aligned}$$

Therefore, the unique solution $x^{p+1} = \left(x^{p+1,1}, \ldots, x^{p+1,k}\right)$ of (4.35) is defined by

$$x^{p+1,j} = \Big(1 - \frac{\beta_{p,j}}{m}\Big)x^{p,j} + \frac{\beta_{p,j}}{m}a^{0,p,j} \tag{4.56}$$

for all $j \in J$. If $A^{p,j} = \emptyset$, then $\beta_{p,j} = 0$. So, from (4.56) it follows that $x^{p+1,j} = x^{p,j}$ for any $j \in J$ with $A^{p,j} = \emptyset$.

Procedure 4.7 (A DCA scheme for solving (4.8)**; see [7, p. 7])**

Input: *An approximate solution* $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ *of* (4.1)*, an integer* $\ell \geq 1$*,*

and a subset $\bar{A}_4 = \{c^1, \ldots, c^r\}$ *of* $\mathbb{R}^n$.

Output: *A set* $\widehat{A}_5 \subset \mathbb{R}^{n(\ell+1)}$ *consisting of some approximate solutions* $x^{p+1} = (x^{p+1,1}, \ldots, x^{p+1,\ell+1})$ *of* (4.8).

Step 1. Set $\widehat{A}_5 = \emptyset$ *and* $s = 1$.

Step 2. If $s > r$, *then stop.*

Step 3. Put $y = c^s$ *and set* $p := 1$.

Step 4. Compute the clusters $\{A^{p,1}, \ldots, A^{p,\ell+1}\}$, *which form the natural clustering associated with* $x^p := (\bar{x}^1, \ldots, \bar{x}^\ell, y) \in \mathbb{R}^{n\times(\ell+1)}$. *Compute the values* $\beta_j = |A^{p,j}|$ *for* $j \in \{1, \ldots, \ell+1\}$.

Step 5. Compute the vectors $x^{p+1,j}$, $j \in \{1, \ldots, \ell+1\}$, *by formula* (4.56).

Step 6. If $x^{p+1,j} = x^{p,j}$ *for* $j \in \{1, \ldots, \ell+1\}$, *then go to Step 8.*

Step 7. Set $p = p + 1$ *and go to Step 4.*

Step 8. Put $\widehat{A}_5 = \widehat{A}_5 \cup \{x^p\}$ *and* $s = s + 1$. *Go to Step 2.*

Combining Procedures 4.5 and 4.7, we have the DC incremental clustering algorithm of Bagirov [7] to solve (3.2).

Algorithm 4.3 (Bagirov's Algorithm [7, p. 8])

Input: *The parameters* n, m, k, *and the data set* $A = \{a^1, \ldots, a^m\}$.

Output: *A centroid system* $\{\bar{x}^1, \ldots, \bar{x}^k\}$ *and the corresponding clusters* $\{A^1, \ldots, A^k\}$.

Step 1. Compute $a^0 = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m} a^i$, *put* $\bar{x}^1 = a^0$, *and set* $\ell = 1$.

Step 2. If $\ell = k$, *then stop; the* k*-partition problem has been solved.*

Step 3. Select two control parameters: $\gamma_1 \in [0, 1]$, $\gamma_2 \in [0, 1]$.

Step 4. Compute $z^1_{\max}$ *by* (4.12) *and the set* $\bar{A}_1$ *by* (4.13).

Step 5. Compute the set $\bar{A}_2$ *by* (4.14), $z^2_{\max}$ *by* (4.15), *and the set* $\bar{A}_3$ *by* (4.16).

Step 6. Apply ***Procedure 4.5*** *to problem* (4.4) *with a starting point* $c \in \bar{A}_3$ *to find the set* $\bar{A}_4$.

Step 7. Apply ***Procedure 4.7*** *to* (4.8) *to obtain the set* $\widehat{A}_5$.

Step 8. Compute the value $f^{\min}_{\ell+1} = \min\left\{f_{\ell+1}(\hat{y}^1, ..., \hat{y}^{\ell+1}) \mid \forall(\hat{y}^1, ..., \hat{y}^{\ell+1}) \in \widehat{A}_5\right\}$ *and put* $\widehat{A}_6 = \left\{(\bar{y}^1, ..., \bar{y}^{\ell+1}) \mid f_{\ell+1}(\bar{y}^1, ..., \bar{y}^{\ell+1}) = f^{\min}_{\ell+1}\right\}$.

Step 9. Set $\bar{x}^j = \bar{y}^j$, $j = 1, ..., \ell+1$. *Put* $\ell = \ell + 1$, *and go to Step 2.*

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.

Procedure 4.8 (A modified version of Procedure 4.7)

Input: *An approximate solution* $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ *of* (4.1)*, an integer* $\ell \geq 1$*, and a subset* $\bar{A}_4 = \{c^1, \ldots, c^r\}$ *of* $\mathbb{R}^n$*.*
Output: *A set* $\widehat{A}_5 \subset \mathbb{R}^{n(\ell+1)}$ *of* r *vectors of the form*

$$x^{p+1} = (x^{p+1,1}, \ldots, x^{p+1,\ell+1}),$$

which are approximate solutions of (4.8)*.*
Step 1. Select a tolerance $\varepsilon > 0$*. Set* $\widehat{A}_5 = \emptyset$ *and* $s = 1$*.*
Step 2. If $s > r$*, then stop.*
Step 3. Put $y = c^s$ *and set* $p = 1$*.*
Step 4. Compute the clusters $\{A^{p,1}, \ldots, A^{p,\ell+1}\}$*, which form the natural clustering associated with* $x^p := (\bar{x}^1, \ldots, \bar{x}^\ell, y) \in \mathbb{R}^{n\times(\ell+1)}$*. Compute the values* $\beta_j = |A^{p,j}|$ *for* $j \in \{1, \ldots, \ell+1\}$*.*
Step 5. Compute the vectors $x^{p+1,j}$*,* $j \in \{1, \ldots, \ell+1\}$*, by formula* (4.56)*.*
Step 6. If $\|x^{p+1,j} - x^{p,j}\| \leq \varepsilon$ *for* $j \in \{1, \ldots, \ell+1\}$*, then go to Step 8.*
Step 7. Set $p = p + 1$ *and go to Step 4.*
Step 8. Put $\widehat{A}_5 = \widehat{A}_5 \cup \{x^{p+1}\}$ *and* $s = s + 1$*. Go to Step 2.*

Based on Procedures 4.6 and 4.8, we can propose the following improvement for Algorithm 4.3.

Algorithm 4.4 (A modified version of Algorithm 4.3)

Input*: The parameters* n, m, k*, and the data set* $A = \{a^1, \ldots, a^m\}$*.*
Output*: A centroid system* $\{\bar{x}^1, \ldots, \bar{x}^k\}$ *and the corresponding clusters* $\{A^1, \ldots, A^k\}$*.*
Step 1. Compute $a^0 = \dfrac{1}{m}\sum_{i=1}^{m} a^i$*, put* $\bar{x}^1 = a^0$*, and set* $\ell = 1$*.*
Step 2. If $\ell = k$*, then stop; the* k*-partition problem has been solved.*

Step 3. Select two control parameters: $\gamma_1 \in [0,1],\ \gamma_2 \in [0,1]$.
Step 4. Compute $z^1_{\max}$ *by* (4.12) *and the set* $\bar{A}_1$ *by* (4.13).
Step 5. Compute the set $\bar{A}_2$ *by* (4.14), $z^2_{\max}$ *by* (4.15), *and the set* $\bar{A}_3$ *by* (4.16).
Step 6. Apply ***Procedure 4.6*** *to problem* (4.4) *with a starting point* $c \in \bar{A}_3$ *to find the set* $\bar{A}_4$.
Step 7. Apply ***Procedure 4.8*** *to* (4.8) *to obtain the set* $\widehat{A}_5$.
Step 8. Compute the value

$$f^{\min}_{\ell+1} = \min\left\{f_{\ell+1}(\hat{y}^1, ..., \hat{y}^{\ell+1}) \mid \forall(\hat{y}^1, ..., \hat{y}^{\ell+1}) \in \widehat{A}_5\right\} \tag{4.57}$$

and put

$$\widehat{A}_6 = \left\{(\bar{y}^1, ..., \bar{y}^{\ell+1}) \mid f_{\ell+1}(\bar{y}^1, ..., \bar{y}^{\ell+1}) = f^{\min}_{\ell+1}\right\}. \tag{4.58}$$

Step 9. Select any element $(\bar{y}^1, ..., \bar{y}^{\ell+1})$ *from* $\widehat{A}_6$ *and set* $\bar{x}^j = \bar{y}^j$ *for all* $j = 1, ..., \ell+1$. *Put* $\ell = \ell + 1$, *and go to Step 2.*

We are interested to know how clustering problem in Example 4.1 can be solved by Algorithm 4.4.

Example 4.3 Let n, m, k, A be as in Example 4.1, i.e., $n = 2$, $m = 3$, $k = 2$, $A = \{a^1, a^2, a^3\}$, where $a^1 = (0,0)$, $a^2 = (1,0)$, $a^3 = (0,1)$. Let $\gamma_1 = \gamma_2 = 0.3$ and $\varepsilon = 10^{-3}$. To implement Algorithm 4.4, observe that the barycenter of A is $a^0 = (\frac{1}{3}, \frac{1}{3})$. We put $\bar{x}^1 = a^0$, and set $\ell = 1$. In Example 4.1 we have shown that $\bar{A}_3 = \bar{A}_2 = \bar{A}_1 = A$. For $x^p = a^1$, by (4.9), (4.31) and (4.32), we have $A_1(a^1) = \{a^1\}$, $A_3(a^1) = \{a^2, a^3\}$, and $A_4(a^1) = \emptyset$. By (4.36), $x^{p+1} = x^p = a^1$. Hence, the stopping criterion in Step 5 of Procedure 4.6 is satisfied. For $x^p = a^2$, by (4.9), (4.31) and (4.32), one has $A_1(a^2) = \{a^2\}$, $A_3(a^2) = \{a^1, a^3\}$, and $A_4(a^2) = \emptyset$. Using (4.36), one obtains $x^{p+1} = x^p = a^2$. For $x^p = a^3$, from (4.9), (4.31) and (4.32) it follows that $A_1(a^3) = \{a^3\}$, $A_3(a^3) = \{a^1, a^2\}$, and $A_4(a^3) = \emptyset$. By (4.36), one has $x^{p+1} = x^p = a^3$. Therefore, the realization of Step 6 of Algorithm 4.4 gives the set $\bar{A}_4 = \{a^1, a^2, a^3\}$. Now, to realize Step 7 of Algorithm 4.4, we apply Procedure 4.8 to solve (4.8).

For $s = 1$, we put $y = c^1 = a^1$ and set $p = 1$. Here, since one has $x^1 = (\bar{x}^1, a^1) = (a^0, a^1)$, the clusters $\{A^{1,1}, A^{1,2}\}$ in Step 4 of Procedure 4.8 are the following: $A^{1,1} = \{a^2, a^3\}$, $A^{1,2} = \{a^1\}$. So, $\gamma_1 = 2$ and $\gamma_2 = 1$. By (4.56), $x^{2,1} = \left(\frac{4}{9}, \frac{4}{9}\right)$ and $x^{2,2} = (0,0)$. It is not difficult to show that

$$x^{p+1,1} = \frac{1}{3}(x^p + (1,1)) \quad \forall p \geq 1$$

and $x^{p+1,2} = (0,0)$ for all $p \geq 1$. Since $x^{1,1} = (\frac{1}{3}, \frac{1}{3})$, from (4.3) we can deduce that $x^{p,1} = (\gamma_p, \gamma_p)$, where $\gamma_p > 0$ for all $p \geq 1$. Also by (4.3), $\gamma_{p+1} = \frac{1}{3}\gamma_p + \frac{1}{3}$, where $\gamma_1 = \frac{1}{3}$. Setting $u_p = \gamma_p - \frac{1}{2}$, one has $u_1 = -\frac{1}{6}$ and $u_{p+1} = \frac{1}{3}u_p$. So, $u_p = -\frac{1}{6}(\frac{1}{3})^p$ and $\gamma_p = \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^p$. Therefore, $\lim\limits_{p\to\infty} x^{p,1} = \lim\limits_{p\to\infty}(\gamma_p, \gamma_p) = (\dfrac{1}{2}, \dfrac{1}{2})$. Thus, the vector

$$x^p = (x^{p,1}, x^{p,2}) = ((\gamma_p, \gamma_p), (0,0))$$

converges to $((\frac{1}{2}, \frac{1}{2}), (0,0))$ as $p \to \infty$. The condition $\|x^{p+1,j} - x^{p,j}\| \leq \varepsilon$ for $j \in \{1, \ldots, \ell+1\}$ in Step 6 of Procedure 4.8 can be rewritten equivalently as $\sqrt{2}|\gamma_{p+1} - \gamma_p| \leq 10^{-3}$. As $\gamma_p = \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^p$, the smallest positive integer p satisfying this condition is $p = 5$. Hence, for $y = c^1 = a^1$, we get

$$\widehat{A}_5 = \emptyset \cup \{x^6\} = \Big\{\Big(\frac{1}{2} - \frac{1}{6}(\frac{1}{3})^6, \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^6\Big), (0,0)\Big\}.$$

Approximately, the first centroid in this system is $(0.49977138, 0.49977138)$.

For $s = 2$, we put $y = c^2 = a^2$ and set $p = 1$. Since $x^1 = (\bar{x}^1, a^2) = (a^0, a^2)$, an analysis similar to the above shows that x^p converges to $\big((0, \frac{1}{2}), (1,0)\big)$ as $p \to \infty$. In addition, the computation by Procedure 4.8, which stops after 7 steps, gives us

$$\begin{aligned}\widehat{A}_5 &= \widehat{A}_5 \cup \{x^7\} \\ &= \Big\{\Big\{\Big(\frac{1}{2} - \frac{1}{6}(\frac{1}{3})^6, \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^6\Big), (0,0)\Big\}, \Big\{\Big((\frac{1}{3})^8, \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8\Big), (1,0)\Big\}\Big\}.\end{aligned}$$

The first element in the second centroid system is

$$\Big((\frac{1}{3})^8, \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8\Big) \approx (0.00015242, 0.49997460).$$

For $s = 3$, we put $y = c^3 = a^3$ and set $p = 1$. Since $x^1 = (\bar{x}^1, a^3) = (a^0, a^3)$, using the symmetry of the data set A, where the position of a^3 is similar to that of a^2, by the above result for $s = 2$ we can assert that x^p converges to $\big((\frac{1}{2}, 0), (0,1)\big)$ as $p \to \infty$. In addition, the computation stops after 7 steps and one has

$$\begin{aligned}\widehat{A}_5 &= \widehat{A}_5 \cup \{x^7\} \\ &= \Big\{\Big\{\Big(\frac{1}{2} - \frac{1}{6}(\frac{1}{3})^6, \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^6\Big), (0,0)\Big\}, \Big\{\Big((\frac{1}{3})^8, \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8\Big), (1,0)\Big\} \\ &\quad \bigcup \Big\{\Big(\frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8, (\frac{1}{3})^8\Big), (0,1)\Big\}\Big\}.\end{aligned}$$

By (4.57), one obtains $f_{\ell+1}^{\min} = f_2^{\min} \approx 0.16666667$. So, in accordance

with (4.58),

$$\widehat{A}_6 = \Big\{\Big((\frac{1}{3})^8, \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8), (1,0)\Big), \Big((\frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8, (\frac{1}{3})^8), (0,1)\Big)\Big\}.$$

Select any element $\bar{x} = (\bar{x}^1, \bar{x}^2)$ from $\widehat{A}_6$. Put $\ell = \ell + 1 = 2$. Since $\ell = k$, the computation terminates. So, we obtain two centroid systems: $\Big(\Big((\frac{1}{3})^8, \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8), (1,0)\Big)$ and $\Big((\frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8, (\frac{1}{3})^8), (0,1)\Big)$. It is worthy to stress that *they are good approximations of the global solutions* $\Big((0, \frac{1}{2}), (1,0)\Big)$ and $\Big((\frac{1}{2}, 0), (0,1)\Big)$ of (3.2).

Unlike Algorithms 4.1 and 4.2, both Algorithms 4.3 and 4.4 do not depend on the parameter γ_3. *The next example shows that Algorithms 4.4 can perform better than the incremental clustering Algorithms 4.1 and 4.2.*

Example 4.4 Consider the set $A = \{a^1, a^2, a^3, a^4\}$, where

$$a^1 = (0,0),\ a^2 = (1,0),\ a^3 = (0,5),\ a^4 = (0,10),$$

$k = 2$, $\gamma_1 = 0$, $\gamma_2 = 0$, and $\gamma_3 = 1.3$. To implement Algorithm 4.1, one computes the barycenter $a^0 = (\frac{1}{4}, \frac{15}{4})$ and puts $\bar{x}^1 = a^0$, $\ell = 1$. Applying Procedure 4.1, one gets $\bar{A}_5 = \{(0,10)\}$. Based on the set $\bar{A}_5$ and the k-means algorithm, $\bar{A}_6 = \Big\{\big((\frac{1}{3}, \frac{5}{3}), (0,10)\big)\Big\}$ (see Step 4 in Algorithm 4.1). Hence, the realization of Step 5 in Algorithm 4.1 gives the centroid system $\bar{x} = \big((\frac{1}{3}, \frac{5}{3}), (0,10)\big)$ and the value $f_{\ell+1}(\bar{x}) = \frac{13}{3}$. Observe that Algorithm 4.2 gives us the same $\hat{x}$ and the same value $f_{\ell+1}(\bar{x}) = \frac{13}{3}$. Thanks to Theorem 3.4, we know that $\bar{x}$ is a nontrivial local solution of (3.2). Observe that $\bar{x}$ is not a solution of the clustering problem in question. The natural clustering associated with this centroid system $\bar{x}$ has two clusters: $A^1 = \{a^1, a^2, a^3\}$ and $A^2 = \{a^4\}$.

Algorithm 4.4 gives better results than the previous two algorithms. Indeed, by (4.16) one has

$$\bar{A}_3 = \Big\{(\frac{1}{2}, 0),\ (\frac{1}{2}, 0),\ (0, \frac{15}{2}),\ (0,10)\Big\}. \tag{4.59}$$

Next, choosing $\varepsilon = 10^{-3}$, we apply Procedure 4.6 to problem (4.4) with initial points from $\bar{A}_3$ to find $\bar{A}_4$. For $x^p = c^1$, where $c^1 = (\frac{1}{2}, 0)$, using (4.9), (4.31) and (4.32), we have $A_1(c^1) = \{a^1, a^2\}$, $A_3(c^1) = \{a^3, a^4\}$, and $A_4(c^1) = \emptyset$. By (4.36), $x^{p+1} = x^p = c^1$. Hence, the stopping criterion in

Step 5 of Procedure 4.6 is satisfied. For $x^p = c^2$, where $c^2 = (\frac{1}{2}, 0)$, we get the same result. For $x^p = c^3$, where $c^3 = (0, \frac{15}{2})$, by (4.9), (4.31) and (4.32), one has $A_1(c^3) = \{a^3, a^4\}$, $A_3(c^3) = \{a^1, a^2\}$, and $A_4(a^2) = \emptyset$. From (4.36) it follows that $x^{p+1} = x^p = c^3$. For $x^p = c^4$, where $c^4 = (0, 10)$, from (4.9), (4.31) and (4.32) it follows that $A_1(c^4) = \{a^1, a^2, a^3\}$, $A_3(c^4) = \{a^4\}$, and $A_4(c^4) = \emptyset$. Using (4.36), one gets $x^{p+1} = x^p = c^4$. Hence, $\bar{A}_4 = \bar{A}_3$, where $\bar{A}_3$ is shown by (4.59).

Now, to realize Step 7 of Algorithm 4.4, we apply Procedure 4.8 to solve (4.8).

For $s = 1$, we put $y = c^1$, $c^1 = (\frac{1}{2}, 0)$, and set $p = 1$. Since one has $x^1 = (\bar{x}^1, c^1) = (a^0, c^1)$, the clusters $\{A^{1,1}, A^{1,2}\}$ in Step 4 of Procedure 4.8 are the following: $A^{1,1} = \{a^3, a^4\}$, $A^{1,2} = \{a^1, a^2\}$. Hence, $\gamma_1 = \gamma_2 = 2$. By (4.56), $x^{2,1} = \big(\frac{1}{8}, \frac{45}{8}\big)$ and $x^{2,2} = (\frac{1}{2}, 0)$. It is not difficult to show that

$$\begin{cases} x_1^{p+1,1} = \frac{1}{2} x_1^{p,1} \quad \forall p \geq 1 \\ x_2^{p+1,1} = \frac{1}{2} x_2^{p,1} + \frac{15}{4} \quad \forall p \geq 1, \end{cases} \tag{4.60}$$

where $x^{p+1,1} = (x_1^{p+1,1}, x_2^{p+1,1})$ and $x^{p+1,2} = (\frac{1}{2}, 0)$ for all $p \geq 1$. Noting that $x^{1,1} = (\frac{1}{4}, \frac{15}{4})$, by (4.60) we have $x^{p,1} = (\gamma_p, \beta_p)$ with $\gamma_p \geq 0$ and $\beta_p \geq 0$ for all $p \geq 1$. By (4.60), one has $\gamma_{p+1} = \frac{1}{2}\gamma_p$ for every $p \geq 1$. Since $\gamma_1 = \frac{1}{4}$, one gets $\gamma_p = (\frac{1}{2})^{p+2}$. Setting $u_p = \beta_p - \frac{15}{2}$, by (4.60) one has $u_{p+1} = \frac{1}{2} u_p$ for every $p \geq 1$. Since $u_1 = -\frac{15}{2}$, one gets $u_p = -15(\frac{1}{2})^p$ and $\beta_p = -15(\frac{1}{2})^p + \frac{15}{2}$. It follows that $\lim\limits_{p\to\infty} x^{p,1} = \lim\limits_{p\to\infty} (\gamma_p, \beta_p) = \big(0, \dfrac{15}{2}\big)$. Thus, the vector

$$x^p = (x^{p,1}, x^{p,2}) = ((\gamma_p, \beta_p), (\frac{1}{2}, 0))$$

converges to $((0, \frac{15}{2}), (\frac{1}{2}, 0))$ as $p \to \infty$. The condition $\|x^{p+1,j} - x^{p,j}\| \leq \varepsilon$ for every $j \in \{1, \ldots, \ell + 1\}$ in Step 6 of Procedure 4.8 can be rewritten equivalently as

$$(\gamma_{p+1} - \gamma_p)^2 + (\beta_{p+1} - \beta_p)^2 \leq 10^{-6}.$$

The smallest positive integer p satisfying this condition is $p = 13$. Hence, for $y = c^1$, we get

$$\widehat{A}_5 = \emptyset \cup \{x^{14}\} = \Big\{\Big((\frac{1}{2})^{14}, -15(\frac{1}{2})^{14} + \frac{15}{2}\Big), (\frac{1}{2}, 0)\Big\}. \tag{4.61}$$

Approximately, the first centroid in this system is $(0.00006104, 7.49816895)$.

For $s = 2$, we put $y = c^2$ and set $p = 1$. Since

$$x^1 = (\bar{x}^1, c^2) = (a^0, c^2) = (a^0, c^1),$$

we get the same centroid system x^{14} shown in (4.61). Hence, the set $\widehat{A}_5$ is updated as follows:

$$\begin{aligned}\widehat{A}_5 &= \widehat{A}_5 \cup \{x^{14}\}\\ &= \Big\{\Big\{\Big((\frac{1}{2})^{14}, -15(\frac{1}{2})^{14}+\frac{15}{2}\Big), (\tfrac{1}{2},0)\Big\}\\ &\quad \bigcup\Big\{\Big((\frac{1}{2})^{14}, -15(\frac{1}{2})^{14}+\frac{15}{2}\Big), (\tfrac{1}{2},0)\Big\}\Big\}.\end{aligned}$$

For $s=3$, we put $y=c^3$, $c^3=(0,\frac{15}{2})$ and set $p=1$. Since $x^1=(\bar{x}^1,c^3)$, an analysis similar to the above shows that x^p converges to $\big((0,\frac{1}{2}),(0,\frac{15}{2})\big)$ as $p\to\infty$. In addition, the computation by Procedure 4.8, which stops after 12 steps, gives us

$$\begin{aligned}\widehat{A}_5 &= \widehat{A}_5 \cup \{x^{13}\}\\ &= \Big\{\Big\{\Big((\frac{1}{2})^{14}, -15(\frac{1}{2})^{14}+\frac{15}{2}\Big), (\frac{1}{2},0)\Big\}, \Big\{\Big((\frac{1}{2})^{14}, -15(\frac{1}{2})^{14}+\frac{15}{2}\Big), (\frac{1}{2},0)\Big\}\\ &\quad \bigcup\Big\{\Big(-3(\frac{1}{2})^{13}+\frac{1}{2}, 15(\frac{1}{2})^{13}\Big), (0,\frac{15}{2})\Big\}\Big\}.\end{aligned}$$

The first element in the third centroid system is

$$\Big(-3(\frac{1}{2})^{13}+\frac{1}{2}, 15(\frac{1}{2})^{13}\Big) \approx (0.00183005, 0.499633789).$$

For $s=4$, we put $y=c^4$, $c^4=(0,10)$ and set $p=1$. Since $x^1=(\bar{x}^1,c^4)$, an analysis similar to the above shows that x^p converges to $\big((\frac{1}{3},\frac{5}{3}),(0,10)\big)$ as $p\to\infty$. In addition, the computation by Procedure 4.8, which stops after 7 steps, gives us

$$\begin{aligned}\widehat{A}_5 &= \widehat{A}_5 \cup \{x^{7}\}\\ &= \Big\{\Big\{\Big((\frac{1}{2})^{14}, -15(\frac{1}{2})^{14}+\frac{15}{2}\Big), (\frac{1}{2},0)\Big\}, \Big\{\Big((\frac{1}{2})^{14}, -15(\frac{1}{2})^{14}+\frac{15}{2}\Big), (\frac{1}{2},0)\Big\}\\ &\quad \bigcup\Big\{\Big(-3(\frac{1}{2})^{13}+\frac{1}{2}, 15(\frac{1}{2})^{13}\Big), (0,\frac{15}{2})\Big\}\\ &\quad \bigcup\Big\{\Big(-\frac{1}{12}(\frac{1}{4})^{8}+\frac{1}{3}, \frac{25}{12}(\frac{1}{4})^{8}+\frac{5}{3}\Big), (0,10)\Big\}\Big\}.\end{aligned}$$

The first element in the fourth centroid system is

$$\Big(-\frac{1}{12}(\frac{1}{4})^{8}+\frac{1}{3}, \frac{25}{12}(\frac{1}{4})^{8}+\frac{5}{3}\Big) \approx (0.33333206, 1.66669846).$$

By (4.57) and the current set $\widehat{A}_5$, one obtains $f_{\ell+1}^{\min}\approx 3.25$. Using (4.58), one gets

$$\widehat{A}_6 = \Big\{\Big((\frac{1}{2})^{14}, -15(\frac{1}{2})^{14}+\frac{15}{2}\Big), (\frac{1}{2},0), \Big(\big((\frac{1}{2})^{14}, -15(\frac{1}{2})^{14}+\frac{15}{2}\big), (\frac{1}{2},0)\Big)\Big\}.$$

Select any element $(\bar{y}^1, \bar{y}^2)$ from the set $\widehat{A}_6$ and set $\bar{x}^j = \bar{y}^j$, $j = 1, 2$. Put $\ell = \ell + 1 = 2$. Since $\ell = k$, the computation terminates. The centroid system $\bar{x} = (\bar{x}^1, \bar{x}^2)$ is *a global solution* of (3.2). The corresponding clusters $\{A^1, A^2\}$ are as follows: $A^1 = \{a^3, a^4\}$ and $A^2 = \{a^1, a^2\}$.

Concerning Algorithms 4.3 and 4.4, one may ask the following questions:

(Q3) *Whether the computation in* Algorithm 4.3 (*resp., in* Algorithm 4.4) *terminates after finitely many steps?*

(Q4) *If the computation in* Algorithm 4.3 (*resp., in* Algorithm 4.4 *with* $\varepsilon = 0$) *does not terminate after finitely many steps, then the iteration sequence* $\{x^p\}$ *converges to a stationary point of* (3.2)*?*

Partial answers to (Q3) and (Q4) are given in the forthcoming statement, which is an analogue of Theorem 4.3.

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* $j \in \{1, \ldots, \ell + 1\}$, *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* $j \in \{1, \ldots, \ell + 1\}$, *the sequence* $\{x^{p,j}\}$ *converges to a point* $\bar{x}^j \in \mathcal{B}$.

Proof. (i) To show that the computation by Algorithm 4.3 may not terminate after finitely many steps, it suffices to construct a suitable example. Let n, m, k, A be as in Example 4.1 and let $\gamma_1 = \gamma_2 = 0.3$. The realization of Steps 1–6 in Algorithm 4.3 gives us the set $\bar{A}_4 = \{a^1, a^2, a^3\}$, the number $\ell = 1$, and the point $\bar{x}^1 = a^0 = (\frac{1}{3}, \frac{1}{3})$. In Step 7 of the algorithm, one applies Procedure 4.7 to (4.8) to obtain the set $\widehat{A}_5$. The analysis given in Example 4.3 shows that, the computation starting with $s = 1$ in Step 1 of

Procedure 4.7 does not terminate, because the stopping criterion $x^{p+1,j} = x^{p,j}$ for $j \in \{1, \ldots, \ell+1\}$ in Step 6 of that procedure is not satisfied for any $p \in \mathbb{N}$.

(ii) For $\varepsilon = 0$, since Algorithm 4.4 (resp., Procedure 4.8) coincides with Algorithm 4.3 (resp., Procedure 4.7), the just given example justifies our claim.

(iii) To obtain the result, one can argue similarly as in the proof of assertion (iii) in Theorem 4.3. This is possible because the iteration formula (4.56) can be rewritten equivalently as

$$x^{p+1,j} = \frac{1}{m}\Big((m - |A^{p,j}|)x^{p,j} + \sum_{a^i \in A^{p,j}} a^i\Big),$$

and the latter has the same structure as that of (4.36).

(iv) The proof is similar to that of assertion (iv) in Theorem 4.3.

(v) The proof is similar to that of assertion (v) in Theorem 4.3.

□

In analogy with Theorem 4.5, we have the following result.

Theorem 4.6 *If the computation by* Procedure 4.8 *with* $\varepsilon = 0$ *does not terminate after finitely many steps then, for every* $j \in \{1, \ldots, \ell+1\}$, *the sequence* $\{x^{p,j}\}$ *converges* $Q-$*linearly to a point* $\bar{x}^j \in \mathcal{B}$. *More precisely, one has*

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

for all p *sufficiently large.*

Proof. The proof is similar to that of Theorem 4.4. □

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 (Step 6) and apply the k-means algorithm in the outer loop (Step 7). First, using the DCA scheme in Procedure 4.6 instead of the k-means algorithm, we can modify Procedure 4.1 as follows.

Procedure 4.9 (Inner Loop with DCA)

Input: *An approximate solution* $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ *of problem* (4.1), $\ell \geq 1$.
Output: *A set* $\bar{A}_5$ *of starting points to solve problem* (4.8).
Step 1. Select three control parameters: $\gamma_1 \in [0,1],\ \gamma_2 \in [0,1],\ \gamma_3 \in [1,\infty)$.
Step 2. Compute $z^1_{\max}$ *by* (4.12) *and the set* $\bar{A}_1$ *by* (4.13).
Step 3. Compute the set $\bar{A}_2$ *by* (4.14), $z^2_{\max}$ *by* (4.15), *and the set* $\bar{A}_3$ *by* (4.16).
Step 4. For each $c \in \bar{A}_3$, *apply* ***Procedure 4.4*** *to problem* (4.4) *to find the set* $\bar{A}_4$.
Step 5. Compute the value $f^{\min}_{\ell+1}$ *by* (4.18).
Step 6. Form the set $\bar{A}_5$ *by* (4.19).

Now we are in a position to present the third DCA algorithm and consider an illustrative with a small-size data set.

Algorithm 4.5 (DCA in the Inner Loop and k-means Algorithm in the Outer Loop)

Input*: The parameters* n, m, k, *and the data set* $A = \{a^1, \ldots, a^m\}$.
Output*: A centroid system* $\{\bar{x}^1, \ldots, \bar{x}^k\}$ *and the corresponding clusters* $\{A^1, \ldots, A^k\}$.
Step 1. Compute $a^0 = \dfrac{1}{m}\sum_{i=1}^{m} a^i$, *put* $\bar{x}^1 = a^0$, *and set* $\ell = 1$.
Step 2. If $\ell = k$, *then stop. Problem* (3.2) *has been solved.*
Step 3. Apply ***Procedure 4.9*** *to find the set* $\bar{A}_5$ *of starting points.*
Step 4. For each point $\bar{y} \in \bar{A}_5$, *apply the k-means algorithm to problem* (4.8) *with the starting point* $(\bar{x}^1, ..., \bar{x}^\ell, \bar{y})$ *to find an approximate solution* $x = (x^1, \ldots, x^{\ell+1})$. *Denote by* $\bar{A}_6$ *the set of these solutions.*
Step 5. Select a point $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ *from* $\bar{A}_6$ *satisfying condition* (4.20). *Define* $\bar{x}^j = \hat{x}^j,\ j = 1, \ldots, \ell+1$. *Set* $\ell = \ell + 1$ *and go to Step 2.*

Example 4.5 Let n, m, k, A be as in Example 4.1, i.e., $n = 2$, $m = 3$, $k = 2$, $A = \{a^1, a^2, a^3\}$, where $a^1 = (0,0)$, $a^2 = (1,0)$, $a^3 = (0,1)$. Let $\gamma_1 = \gamma_2 = 0.3$ and $\gamma_3 = 3$. The barycenter of A is $a^0 = (\frac{1}{3}, \frac{1}{3})$. To implement Algorithm 4.5, put $\bar{x}^1 = a^0$ and set $\ell = 1$. Since $\ell < k$, we apply Procedure 4.9 to compute set

$\bar{A}_5$. The sets $\bar{A}_1, \bar{A}_2$ and $\bar{A}_3$ have been found in Example 4.1. Namely, we have $\bar{A}_3 = \bar{A}_2 = \bar{A}_1 = A = \{a^1, a^2, a^3\}$. Applying Procedure 4.6 to problem (4.4) with initial points from $\bar{A}_3$, we find $\bar{A}_4$. Since this computation of $\bar{A}_4$ is the same as that in Example 4.3, we have $\bar{A}_4 = \bar{A}_3 = A$. The calculations of $\bar{A}_5$ and $\bar{A}_6$ are as in Example 4.1. Thus, we get one of the two centroid systems, which is *a global solution* of (3.2). If $\bar{x} = \hat{x} = \Big((0, \frac{1}{2}), (1, 0)\Big)$, then $A^1 = \{a^1, a^3\}$ and $A^2 = \{a^2\}$. If $\bar{x} = \hat{x} = \Big((\frac{1}{2}, 0), (0, 1)\Big)$, then $A^1 = \{a^1, a^2\}$ and $A^2 = \{a^3\}$.

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 (4.8). If one applies the DCA instead, then one obtains an DC algorithm, which is based on the next procedure.

Procedure 4.10 (Solve (4.8) **by DCA)**

Input: *An approximate solution* $\bar{x} = (\bar{x}^1, ..., \bar{x}^\ell)$ *of problem* (4.1), $\ell \geq 1$.
Output: *An approximate solution* $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ *of problem* (4.8).
Step 1. Select three control parameters: $\gamma_1 \in [0, 1],\ \gamma_2 \in [0, 1],\ \gamma_3 \in [1, \infty)$.
Step 2. Compute $z^1_{\max}$ *by* (4.12) *and the set* $\bar{A}_1$ *by* (4.13).
Step 3. Compute the set $\bar{A}_2$ *by* (4.14), $z^2_{\max}$ *by* (4.15), *and the set* $\bar{A}_3$ *by* (4.16).
Step 4. Using (4.17), *form the set* Ω.
Step 5. Apply ***Procedure 4.8*** *to problem* (4.8) *for each initial vector centroid system* $(\bar{x}^1, ..., \bar{x}^\ell, c) \in \Omega$ *to get the set* $\widetilde{A}_4$ *of candidates for approximate solutions of* (4.8) *for* $k = \ell + 1$.
Step 6. Compute the value $\widetilde{f}^{\min}_{\ell+1}$ *by* (4.21) *and the set* $\widetilde{A}_5$ *by* (4.22).
Step 7. Pick a point $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ *from* $\widetilde{A}_5$.

Algorithm 4.6 (Solve (3.2) **by just one DCA procedure)**

Input*: The parameters* n, m, k, *and the data set* $A = \{a^1, \ldots, a^m\}$.
Output*: The set of* k *cluster centers* $\{\bar{x}^1, \ldots, \bar{x}^k\}$ *and the corresponding clus-*

ters $A^1, ..., A^k$.

Step 1. Compute $a^0 = \dfrac{1}{m}\sum_{i=1}^{m} a^i$, *put* $\bar{x}^1 = a^0$, *and set* $\ell = 1$.

Step 2. If $\ell = k$, *then go to Step 5.*

Step 3. Use ***Procedure 4.10*** *to find an approximate solution* $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ *of problem* (4.8).

Step 4. Put $\bar{x}^j = \hat{x}^j$, $j = 1, \ldots, \ell+1$. *Set* $\ell = \ell + 1$ *and go to Step 2.*

Step 5. Compute $\widetilde{A}_6$ *by* (4.23) *and select an element* $\bar{x} = (\bar{x}^1, \ldots, \bar{x}^k)$ *from* $\widetilde{A}_6$. *Using the centroid system* $\bar{x}$, *apply the natural clustering procedure to partition* A *into* k *clusters* $A^1, ..., A^k$. *Print* $\bar{x}$ *and* $A^1, ..., A^k$. *Stop.*

Example 4.6 Let n, m, k, A be as in Example 4.1, i.e., $n = 2$, $m = 3$, $k = 2$, $A = \{a^1, a^2, a^3\}$, where $a^1 = (0, 0)$, $a^2 = (1, 0)$, $a^3 = (0, 1)$. Let $\gamma_1 = 0.3$, $\gamma_2 = 0.3$ and $\gamma_3 = 3$. The implementation of Algorithm 4.6 begins with putting $\bar{x}^1 = a^0$ and setting $\ell = 1$. Since $\ell < k$, we apply Procedure 4.10 to find an approximate solution $\hat{x} = (\hat{x}^1, \ldots, \hat{x}^{\ell+1})$ of problem (4.8). By the results in Example 4.1, we have $\bar{A}_3 = \bar{A}_2 = \bar{A}_1 = A = \{a^1, a^2, a^3\}$. Next, we apply Procedure 4.10 to (4.8) with initial points from $\Omega = \{(\bar{x}^1, a^1), (\bar{x}^1, a^2), (\bar{x}^1, a^3)\}$ to find $\widetilde{A}_4$. Since the calculation of $\tilde{A}_4$ coincides with that of $\widehat{A}_5$ in Example 4.3, one gets

$$\begin{aligned}\widetilde{A}_4 &= \widehat{A}_5 \\ &= \Big\{\Big\{\Big(\frac{1}{2} - \frac{1}{6}(\frac{1}{3})^6, \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^6\Big), (0,0)\Big\}, \Big\{\Big((\frac{1}{3})^8, \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8\Big), (1,0)\Big\} \\ &\quad \bigcup \Big\{\Big(\frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8, (\frac{1}{3})^8\Big), (0,1)\Big\}\Big\}.\end{aligned}$$

By (4.21), we have

$$\widetilde{A}_5 = \Big\{\Big(\Big((\frac{1}{3})^8, \frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8\Big), (1,0)\Big), \Big(\Big(\frac{1}{2} - \frac{1}{6}(\frac{1}{3})^8, (\frac{1}{3})^8\Big), (0,1)\Big)\Big\}. \quad (4.62)$$

Put $\bar{x}^j = x^j$ for $j = 1, 2$. Set $\ell = 2$ and go to Step 2. Using (4.23), we get $\widetilde{A}_6 = \widetilde{A}_5$. Thus, we obtain one of the two centroid systems described in (4.62). If $\bar{x}$ happens to be the first centroid system, then $A^1 = \{a^1, a^3\}$ and $A^2 = \{a^2\}$. If the second centroid system is selected, then $A^1 = \{a^1, a^2\}$ and $A^2 = \{a^3\}$.

4.4 Numerical Tests

Using several well-known real-world data sets, we have tested the efficiencies of the Algorithms 4.1, 4.2, 4.4, 4.5, and 4.6 above, and compared them with that of the k-means Algorithm, which has been denoted by KM. The six algorithms were implemented in the Visual C++ 2010 environment, and performed on a PC Intel Core^{TM} i7 (4 x 2.0 GHz) processor, 4GB RAM. 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. Brief descriptions of these data sets are given in Table 4.1 of the data sets. Their detailed descriptions can be found in [56].

Table 4.1: Brief descriptions of the data sets

Data sets	Number of instances	Number of attributes
Iris	150	4
Wine	178	13
Glass	214	9
Heart	270	13
Gene	384	17
Synthetic Control	600	60
Balance Scale	625	4
Stock Price	950	10

The computational results for the first 4 data sets, where $150 \leq m \leq 300$, are given in Table 4.2. In Table 4.3, we present the computational results for the last 4 data sets, where $300 < m < 1000$. In Tables 4.2 and 4.3, $k \in \{2, 3, 5, 7, 9, 10\}$ is the number of clusters; *fbest* is the best value of the cluster function $f(x)$ in (3.2) found by the algorithm, and CPU is the CPU time (in seconds). Since there are 8 data sets (see Table 4.1) and 6 possibilities for the number k of the data clusters (namely, $k \in \{2, 3, 5, 7, 9, 10\}$), one has **48 cases** in Tables 4.2 and 4.3.

- *Comparing Algorithm 4.2 with Algorithm 4.1*, we see that there are 9 cases where Alg. 2 performs better than Alg. 4.1 in term of the CPU time, while there are **37 cases** where Alg. 4.2 performs better than Alg. 4.1 in term of the best value of the cluster function.

- *Comparing Algorithm 4.5 with Algorithm 4.4*, we see that there are 14 cases where Alg. 4.5 performs better than Alg. 4.4 in term of the CPU time, while there are **48 cases** where Alg. 4.5 performs better than Alg. 4.4 in term

of the best value of the cluster function.

- *Comparing Algorithm 4.5 with Algorithm 4.6*, we see that there are 17 cases where Alg. 4.5 performs better than Alg. 4.6 in term of the CPU time, while there are **32 cases** where Alg. 4.5 performs better than Alg. 4.6 in term of the best value of the cluster function.

- *Comparing Algorithm 4.2 with KM*, we see that there are **39 cases** where Alg. 4.2 performs better than KM in term of the best value of the cluster function.

- *Comparing Algorithm 4.5 with KM*, we see that there are **45 cases** where Alg. 4.5 performs better than KM in term of the best value of the cluster function.

The above analysis of the computational results is summarized in Table 4.4. Clearly, 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. It is worthy to stress that the construction of the sets $A_i(y)$, $i = 1, \ldots, 4$, and the sets $\bar{A}_1$, $\bar{A}_2$, etc., as well as the choice of the control parameters $\gamma_1, \gamma_2, \gamma_3$ allow one to approach different parts of the given data set A. Thus, the computation made by each one of the Algorithms 4.1, 4.2, 4.4, 4.5, and 4.6, is more flexible than that of KM. This is the reason why the just mentioned incremental clustering algorithms usually yield better values of the cluster function than KM.

Table 4.2: Results for data sets with $150 \leq m \leq 300$

k	KM		Algorithm 4.1		Algorithm 4.2		Algorithm 4.4		Algorithm 4.5		Algorithm 4.6	
	CPU	fbest	CPU	fbest	CPU	fbest	CPU	fbest	CPU	fbest	CPU	fbest
Iris												
2	0.063	1.016	3.011	1.016	7.839	1.016	3.135	1.163	6.224	1.016	5.115	1.023
3	0.156	0.526	7.956	0.526	9.656	0.526	2.247	0.597	2.870	0.526	2.476	0.535
5	0.047	0.342	8.299	0.333	17.945	0.312	6.567	0.358	5.569	0.312	1.948	0.324
7	0.046	0.246	11.575	0.233	14.065	0.233	4.540	0.274	9.640	0.233	3.236	0.265
9	0.063	0.221	14.320	0.208	15.197	0.187	6.364	0.235	12.543	0.192	2.075	0.216
10	0.078	0.208	16.177	0.178	16.504	0.173	7.769	0.218	14.570	0.178	3.538	0.195
Wine												
2	0.031	549258.852	1.903	541862.384	6.169	534974.704	1.591	537561.829	1.248	534974.704	3.024	534974.704
3	0.047	514451.879	3.042	502433.589	3.027	487665.786	1.685	502246.122	2.901	488673.131	2.059	487665.786
5	0.062	452791.214	8.939	403343.625	7.138	400629.991	3.853	439653.844	4.165	403343.625	3.645	400844.924
7	0.078	378303.123	5.757	327987.888	10.509	320683.364	1.295	352424.277	5.554	323131.978	2.357	320217.370
9	0.078	290208.454	17.254	258785.447	13.640	241848.925	1.887	275219.324	7.161	248803.908	4.407	242410.937
10	0.047	276982.408	11.560	212663.744	16.809	204041.211	1.966	242756.566	6.817	208318.667	4.295	203434.844
Glass												
2	0.031	3564.149	2.324	3533.231	10.317	3533.231	1.701	3533.784	1.809	3533.231	2.707	3533.231
3	0.015	3069.947	2.745	3060.957	14.566	3020.039	1.310	3043.924	1.732	3024.761	2.795	3020.039
5	0.062	2159.891	3.417	2094.145	22.431	2052.915	2.043	2085.013	3.307	2008.142	3.776	2052.915
7	0.046	1364.233	3.619	1119.563	29.620	1108.555	1.747	1206.080	4.914	1078.542	2.456	1108.260
9	0.047	673.142	15.553	280.064	171.167	228.027	2.262	426.809	11.014	280.064	4.793	280.064
10	0.055	1145.738	8.674	9.433	33.331	9.433	3.494	57.236	8.159	9.433	5.675	9.433
Heart												
2	0.031	70328.745	3.634	70328.745	3.313	66330.308	1.579	72549.620	2.877	70186.358	2.557	66349.704
3	0.016	63254.621	2.262	63512.422	3.718	57475.501	1.535	65590.605	1.719	63806.208	3.450	58790.659
5	0.031	47662.713	3.479	45446.462	5.839	44880.560	2.113	52694.631	2.302	49256.104	3.455	42185.487
7	0.062	39488.646	4.212	38168.810	10.661	34548.338	1.330	41132.504	2.123	36499.093	2.756	33081.935
9	0.047	28085.005	3.566	23619.804	15.435	23844.306	2.792	31353.253	1.949	23619.804	3.293	21868.980
10	0.047	25413.423	3.837	22841.603	18.933	18483.967	1.796	27369.822	2.487	20036.279	3.878	17248.164

Table 4.3: Results for data sets with $300 < m < 1000$

	k	KM		Algorithm 4.1		Algorithm 4.2		Algorithm 4.4		Algorithm 4.5		Algorithm 4.6	
		CPU	fbest	CPU	fbest	CPU	fbest	CPU	fbest	CPU	fbest	CPU	fbest
Gene													
	2	0.100	19.763	5.787	19.605	7.366	19.597	2.504	21.280	2.747	19.605	3.054	19.985
	3	0.103	17.997	5.121	17.983	5.462	17.960	0.908	20.376	3.740	17.968	2.408	18.854
	5	0.093	16.586	4.745	16.462	8.555	16.494	0.981	19.342	3.549	16.459	4.041	16.856
	7	0.091	15.748	7.355	15.797	8.113	16.494	1.067	18.409	5.123	15.633	3.487	15.857
	9	0.081	14.998	7.955	14.971	11.503	15.625	1.629	17.598	5.606	14.871	4.570	15.015
	10	0.098	14.670	11.405	14.570	13.481	14.854	2.587	17.248	6.372	14.527	6.199	14.695
Synthtic Control													
	2	0.078	5420.831	1.607	6514.797	14.120	5374.252	4.665	6514.797	1.029	6514.797	5.516	5374.952
	3	0.062	4475.241	2.528	4052.743	14.312	4007.072	3.760	4787.951	1.669	4052.743	5.423	4475.409
	5	0.093	2762.076	8.221	2590.846	23.794	2590.846	3.448	3105.945	4.758	2590.846	10.538	2593.273
	7	0.140	3307.790	4.586	2182.728	34.652	2061.205	4.009	2760.146	4.243	2182.728	13.737	2214.051
	9	0.219	3026.856	6.038	1994.641	43.008	1787.536	4.072	2702.741	4.727	1993.433	14.829	1879.731
	10	0.202	3006.403	7.691	1964.729	48.403	1672.576	4.930	2670.838	7.238	1933.637	14.749	1754.639
Balance Scale													
	2	0.078	9.264	8.595	9.266	46.048	9.180	3.385	9.923	8.080	9.180	4.719	9.288
	3	0.109	7.554	14.274	7.633	53.883	7.512	4.181	8.590	12.417	7.488	5.437	7.768
	5	0.093	5.721	23.026	5.686	68.169	5.554	5.741	6.439	17.176	5.635	6.014	6.218
	7	0.140	4.615	31.528	4.539	79.125	4.521	7.831	5.307	27.768	4.521	9.078	4.869
	9	0.218	3.995	41.886	3.948	89.314	3.885	12.292	4.500	37.050	3.643	8.306	4.246
	10	0.141	3.677	47.174	3.647	103.406	3.582	13.353	4.270	30.217	3.599	10.344	4.022
Stock Price													
	2	0.124	284.176	9.664	284.176	10.220	394.485	2.450	358.554	6.427	284.176	4.776	307.215
	3	0.146	226.263	18.512	224.458	17.696	363.036	2.715	306.419	11.638	224.458	7.320	232.700
	5	0.210	128.097	57.946	123.361	21.466	222.894	3.494	243.327	18.315	125.067	9.735	131.842
	7	0.266	88.573	74.339	74.278	24.599	163.061	3.838	155.157	22.870	85.286	9.219	80.890
	9	0.103	73.693	83.350	55.036	31.585	142.396	4.414	149.415	52.299	52.670	17.649	60.695
	10	0.275	68.755	90.681	48.839	29.624	130.664	6.458	123.943	58.519	50.902	19.954	53.569

Table 4.4: The summary table

	CPU time	fbest
Algorithm 4.2 vs. Algorithm 4.1	9	37
Algorithm 4.5 vs. Algorithm 4.4	14	48
Algorithm 4.5 vs. Algorithm 4.6	17	32
Algorithm 4.2 vs. KM	0	39
Algorithm 4.5 vs. KM	0	45

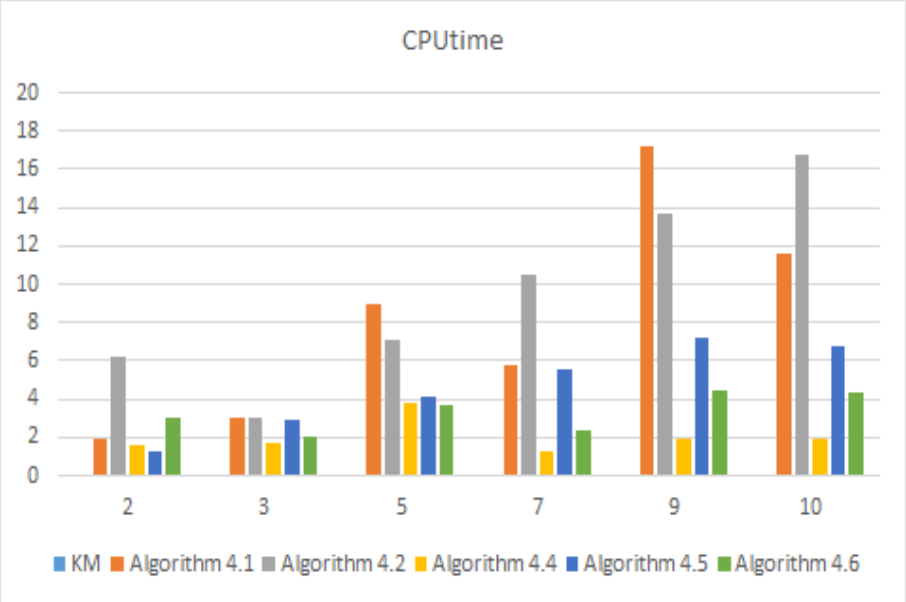

Figure 4.1: The CPU time of the algorithms for the Wine data set

4.5 Conclusions

We have presented the incremental DC clustering algorithm of Bagirov and proposed three modified versions Algorithms 4.4, 4.5, and 4.6 for this algorithm. By constructing some concrete MSSC problems with small data sets, we have shown how these algorithms work.

Two convergence theorems and two theorems about the $Q-$linear convergence rate of the first modified version of Bagirov's algorithm have been obtained by some delicate arguments.

Numerical tests of the above-mentioned algorithms on some real-world databases have shown the effectiveness of the proposed algorithms.

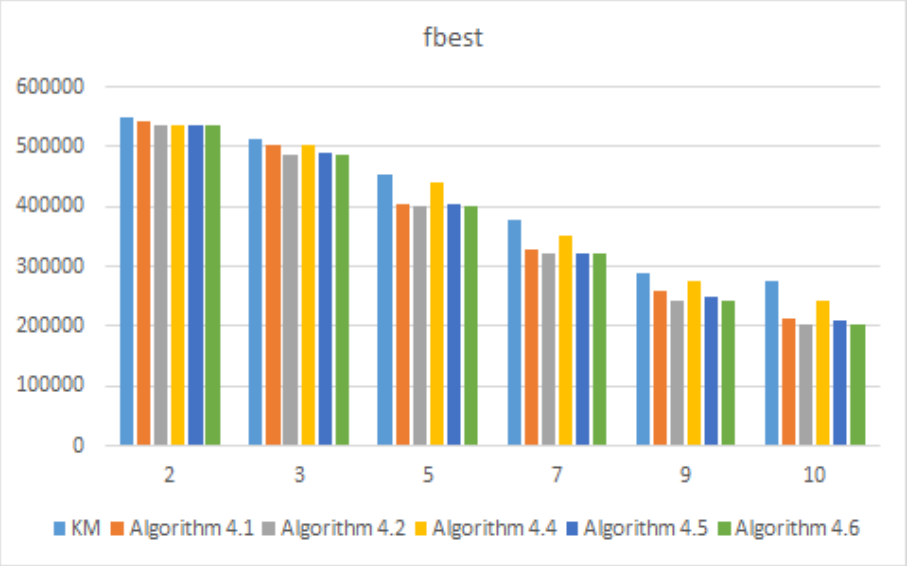

Figure 4.2: The value of objective function of the algorithms for the Stock Wine data set

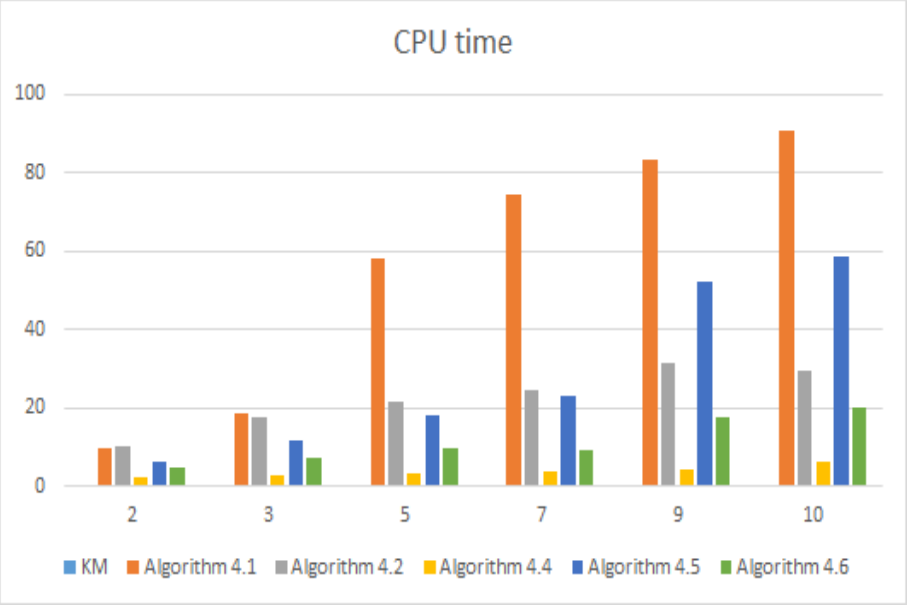

Figure 4.3: The CPU time of the algorithms for the Stock Price data set

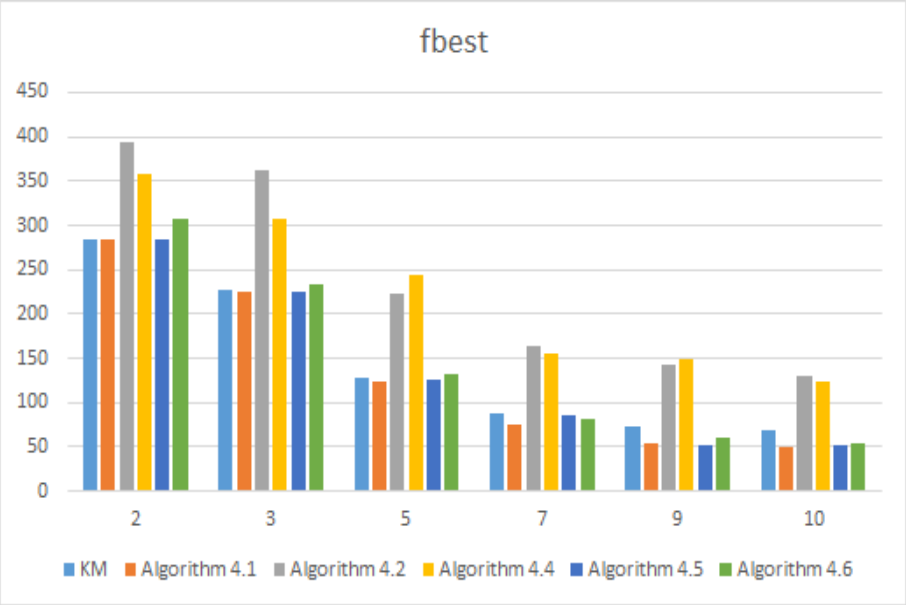

Figure 4.4: The value of objective function of the algorithms for the Stock Price data set

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 convex analysis, set-valued analysis, and optimization theory to study qualitative properties (solution existence, finiteness of the global solution set, 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 (i.e., DCAs with a additional line search procedure at each iteration step; see [5]) to increase the computation speed;

- Qualitative properties and solution methods for constrained clustering problems (see [14, 24, 73, 74] 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 1.52 (2019), Q1 in Control and Optimization, H-index 39; MCQ of 2019: 0.76)

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; IF 0.71, Q2 in Applied Mathematics, H-index 23; 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; IF 0.71, Q2 in Applied Mathematics, H-index 23; MCQ of 2019: 0.56)

References

[1] C. C. Aggarwal, C. K. Reddy: *Data Clustering Algorithms and Applications*, Chapman & Hall/CRC Press, Boca Raton, Florida, 2014.

[2] F. B. Akoa, *Combining DC Algorithms (DCAs) and decomposition techniques for the training of nonpositive–semidefinite kernels*, IEEE Trans. Neur. Networks **19** (2008), 1854–1872.

[3] D. Aloise, A. Deshpande, P. Hansen, P. Popat, *NP-hardness of Euclidean sum-of-squares clustering*, Mach. Learn. **75** (2009), 245–248.

[4] N. T. An, N. M. Nam, *Convergence analysis of a proximal point algorithm for minimizing differences of functions*, Optimization **66** (2017), 129–147.

[5] F. J. Aragón Artacho, R. M. T. Fleming, P. T. Vuong, *Accelerating the DC algorithm for smooth functions*, Math. Program. **169** (2018), 95–118.

[6] A. M. Bagirov, *Modified global k-means algorithm for minimum sum-of-squares clustering problems*, Pattern Recognit. **41** (2008), 3192–3199.

[7] A. M. Bagirov, *An incremental DC algorithm for the minimum sum-of-squares clustering*, Iranian J. Oper. Res. **5** (2014), 1–14.

[8] A. M. Bagirov, E. Mohebi, *An algorithm for clustering using L_1-norm based on hyperbolic smoothing technique*, Comput. Intell. **32** (2016), 439–457.

[9] A. M. Bagirov, A. M. Rubinov, N. V. Soukhoroukova, J. Yearwood, *Unsupervised and supervised data classification via nonsmooth and global optimization*, TOP **11** (2003), 1–93.

[10] A. M. Bagirov, S. Taher, *A DC optimization algorithm for clustering problems with L_1-norm*, Iranian J. Oper. Res. **2** (2017), 2–24.

[11] A. M. Bagirov, J. Ugon, *Nonsmooth DC programming approach to clusterwise linear regression: optimality conditions and algorithms*, Optim. Methods Softw. **33** (2018), 194–219.

[12] A. M. Bagirov, J. Ugon, D. Webb, *Fast modified global k-means algorithm for incremental cluster construction*, Pattern Recognit. **44** (2011), 866–876.

[13] A. M. Bagirov, J. Yearwood, *A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems*, European J. Oper. Res. **170** (2006), 578–596.

[14] S. Basu , I. Davidson, K. L. Wagstaff, *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, CRC Press, New York, 2009.

[15] H. H. Bock, *Clustering and neural networks*. In *"Advances in Data Science and Classification"*, Springer, Berlin (1998), pp. 265–277.

[16] I. M. Bomze, *On standard quadratic optimization problems*, J. Global Optim. **13** (1998), 369–387.

[17] I. M. Bomze, G. Danninger, *A finite algorithm for solving general quadratic problems*, J. Global Optim. **4** (1994), 1–16.

[18] M. J. Brusco, *A repetitive branch-and-bound procedure for minimum within-cluster sum of squares partitioning*, Psychometrika, **71** (2006), 347–363.

[19] R. Cambini, C. Sodini, *Decomposition methods for solving nonconvex quadratic programs via Branch and Bound*, J. Global Optim. **33** (2005), 313–336.

[20] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Second edition, SIAM, Philadelphia, 1990.

[21] G. Cornuéjols, J. Peña, R. Tütüncü, *Optimization Methods in Finance*, Second edition, Cambridge University Press, Cambridge, 2018.

[22] L. R. Costa, D. Aloise, N. Mladenović, *Less is more: basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering*, Inform. Sci. **415/416** (2017), 247–253.

[23] T. F. Covões, E. R. Hruschka, J. Ghosh, *A study of k-means-based algorithms for constrained clustering*, Intelligent Data Analysis **17** (2013), 485–505.

[24] I. Davidson, S. S. Ravi, *Clustering with constraints: Feasibility issues and the k-means algorithm*, In: Proceedings of the 5th SIAM Data Mining Conference, 2005.

[25] V. F. Dem'yanov, A. M. Rubinov, *Constructive Nonsmooth Analysis*, Peter Lang Verlag, Frankfurt am Main, 1995.

[26] V. F. Dem'yanov, L. V. Vasil'ev, *Nondifferentiable Optimization*, Translated from the Russian by T. Sasagawa, Optimization Software Inc., New York, 1985.

[27] G. Diehr, *Evaluation of a branch and bound algorithm for clustering*, SIAM J. Sci. Stat. Comput. **6** (1985), 268–284.

[28] O. Du Merle, P. Hansen, B. Jaumard, N. Mladenović, *An interior point algorithm for minimum sum of squares clustering*, SIAM J. Sci. Comput. **21** (2000), 1485–1505.

[29] N. I. M. Gould, Ph. L. Toint, *A Quadratic Programming Page*, http://www.numerical.rl.ac.uk/people/nimg/qp/qp.html.

[30] O. K. Gupta, *Applications of quadratic programming*, J. Inf. Optim. Sci. **16** (1995), 177–194.

[31] N. T. V. Hang, N. D. Yen, *On the problem of minimizing a difference of polyhedral convex functions under linear constraints*, J. Optim. Theory Appl. **171** (2016), 617–642.

[32] J. Han, M. Kamber, J. Pei, *Data Mining: Concepts and Techniques*, Third edition, Morgan Kaufmann, New York, 2012.

[33] P. Hansen, E. Ngai, B. K. Cheung, N. Mladenovic, *Analysis of global k-means, an incremental heuristic for minimum sum-of-squares clustering*, J. Classif. **22** (2005), 287–310.

[34] P. Hansen, N. Mladenović, *Variable neighborhood decomposition search*, J. Heuristics **7** (2001), 335–350.

[35] P. Hansen, N. Mladenović, *J-means: a new heuristic for minimum sum-of-squares clustering*, Pattern Recognit. **4** (2001), 405–413.

[36] P. T. Hoai, *Some Nonconvex Optimization Problems: Algorithms and Applications*, Ph.D. Dissertation, Hanoi University of Science and Technology, Hanoi, 2019.

[37] R. Horst, H. Tuy, *Global Optimization, Deterministic Approaches*, Second edition, Springer-Verlag, Berlin, 1993.

[38] A. D. Ioffe, V. M. Tihomirov, *Theory of Extremal Problems*, North-Holland Publishing Company, Amsterdam, 1979.

[39] A. K. Jain, *Data clustering: 50 years beyond K-means*, Pattern Recognit. Lett. **31** (2010), 651–666.

[40] N. Jain, V. Srivastava, *Data mining techniques: A survey paper*, Inter. J. Res. Engineering Tech. **2** (2010), no. 11, 116–119.

[41] T.-C. Jen, S.-J. Wang, *Image enhancement based on quadratic programming*, Proceedings of the 15th IEEE International Conference on Image Processing, pp. 3164–3167, 2008.

[42] K. Joki, A. M. Bagirov, N. Karmitsa, M.M. Mäkelä, S. Taheri, *Cluster-wise support vector linear regression*, European J. Oper. Res. **287** (2020), 19–35.

[43] M. Kantardzic, *Data Mining Concepts, Models, Methods, and Algorithms*, Second edition, John Wiley & Sons, Hoboken, New Jersey, 2011.

[44] N. Karmitsa, A. M. Bagirov, S. Taheri, *New diagonal bundle method for clustering problems in large data sets*, European J. Oper. Res. **263** (2017), 367–379.

[45] D. Kinderlehrer, G. Stampacchia, *An Introduction to Variational Inequalities and Their Applications*, Academic Press, Inc., New York-London, 1980.

[46] H. Konno, P. T. Thach, H. Tuy, *Optimization on Low Rank Nonconvex Structures*, Kluwer Academic Publishers, Dordrecht, 1997.

[47] W. L. G. Koontz, P. M. Narendra, K. Fukunaga, *A branch and bound clustering algorithm*, IEEE Trans. Comput. **24** (1975), 908–915.

[48] K. M. Kumar, A. R. M. Reddy, *An efficient k-means clustering filtering algorithm using density based initial cluster centers*, Inform. Sci. **418/419** (2017), 286–301.

[49] J. Z. C. Lai, T.-J. Huang, *Fast global k-means clustering using cluster membership and inequality*, Pattern Recognit. **43** (2010), 731–737.

[50] G. M. Lee, N. N. Tam, N. D. Yen, *Quadratic Programming and Affine Variational Inequalities: A Qualitative Study*, Springer–Verlag, New York, 2005.

[51] H. A. Le Thi, M. T. Belghiti, T. Pham Dinh, *A new efficient algorithm based on DC programming and DCA for clustering*, J. Global Optim. **37** (2007), 593–608.

[52] H. A. Le Thi, M. Le Hoai, T. Pham Dinh, *New and efficient DCA based algorithms for minimum sum-of-squares clustering*, Pattern Recognition **47** (2014), 388–401.

[53] H. A. Le Thi, , V. N. Huynh, T. Pham Dinh, *Convergence analysis of DCA with subanalytic data*, J. Optim. Theory Appl. **179** (2018), 103–126.

[54] H. A. Le Thi, T. Pham Dinh, *The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems*, Ann. Oper. Res. **133** (2005), 23–46.

[55] H. A. Le Thi, T. Pham Dinh, *DC programming and DCA: thirty years of developments*, Math. Program. **169** (2018), Ser. B, 5–68.

[56] M. Lichman, *UCI machine learning repository*, University of California, Irvine, School of Information and Computer Sciences, 2013; http://archive.ics.uci.edu/ml.

[57] H. A. Le Thi, T. Pham Dinh, N. D. Yen, *Properties of two DC algorithms in quadratic programming*, J. Global Optim. **49** (2011), 481–495.

[58] H. A. Le Thi, T. Pham Dinh, N. D. Yen, *Behavior of DCA sequences for solving the trust-region subproblem*, J. Global Optim. **53** (2012), 317–329.

[59] W. J. Leong, B. S. Goh, *Convergence and stability of line search methods for unconstrained optimization*, Acta Appl. Math. **127** (2013), 155–167.

[60] H. A. Le Thi, T. Pham Dinh, *Minimum sum-of-squares clustering by DC programming and DCA*, ICIC 2009, LNAI 5755 (2009), 327–340.

[61] A. Likas, N. Vlassis, J. J. Verbeek, *The global k-means clustering algorithm*, Pattern Recognit. **36** (2003), 451–461.

[62] F. Liu, X. Huang, J. Yang, *Indefinite kernel logistic regression*, Preprint [arXiv:1707.01826v1], 2017.

[63] F. Liu, X. Huang, C. Peng, J. Yang, N. Kasabov, *Robust kernel approximation for classification*, Proceedings of the 24th International Conference "Neural Information Processing", ICONIP 2017, Guangzhou, China, November 14–18, 2017, Proceedings, Part I, pp. 289–296, 2017.

[64] Z.-Q. Luo, *New error bounds and their applications to convergence analysis of iterative algorithms*, Math. Program. **88** (2000), 341–355.

[65] Z.-Q. Luo, P. Tseng, *Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem*, SIAM J. Optim. **2** (1992), 43–54.

[66] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297, 1967.

[67] M. Mahajan, P. Nimbhorkar, K. Varadarajan, *The planar k-means problem is NP-hard*, Theoret. Comput. Sci. **442** (2012), 13–21.

[68] B. A. McCarl, H. Moskowitz, H. Furtan, *Quadratic programming applications*, Omega **5** (1977), 43–55.

[69] L. D. Muu, T. D. Quoc, *One step from DC optimization to DC mixed variational inequalities*, Optimization **59** (2010), 63–76.

[70] J. Nocedal, S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[71] B. Ordin, A. M. Bagirov, *A heuristic algorithm for solving the minimum sum-of-squares clustering problems*, J. Global Optim. **61** (2015), 341–361

[72] P. M. Pardalos, S. A. Vavasis, *Quadratic programming with one negative eigenvalue is NP-hard*, J. Global Optim. **1** (1991), 15–22.

[73] D. Pelleg, D. Baras, *K-Means with large and noisy constraint sets*, In "Machine Learning: ECML 2007" (J. N. Kok et al., Eds.), Series "Lecture Notes in Artificial Intelligence" **4701**, pp. 674–682, 2007.

[74] D. Pelleg, D. Baras, *K-Means with large and noisy constraint sets*, Technical Report H-0253, IBM, 2007.

[75] J. Peng, Y. Xia, *A cutting algorithm for the minimum sum-of-squared error clustering*, Proceedings of the SIAM International Data Mining Conference, 2005.

[76] T. Pereira, D. Aloise, B. Daniel, J. Brimberg, N. Mladenović, *Review of basic local searches for solving the minimum sum-of-squares clustering problem. Open problems in optimization and data analysis*, Springer Optim. Appl. **141**, pp. 249–270, Springer, Cham, 2018.

[77] T. Pham Dinh, H. A. Le Thi, *Convex analysis approach to d.c. programming: theory, algorithms and applications*, Acta Math. Vietnam. **22** (1997), 289–355.

[78] T. Pham Dinh, H. A. Le Thi, *Solving a class of linearly constrained indefinite quadratic programming problems by d.c. algorithms*, J. Global Optim. **11** (1997), 253–285.

[79] T. Pham Dinh, H. A. Le Thi, *A d.c. optimization algorithm for solving the trust-region subproblem*, SIAM J. Optim. **8** (1998), 476–505.

[80] T. Pham Dinh, H. A. Le Thi, *A branch and bound method via DC optimization algorithm and ellipsoidal techniques for box constrained nonconvex quadratic programming problems*, J. Global Optim. **13** (1998), 171–206.

[81] T. Pham Dinh, H. A. Le Thi, *DC (difference of convex functions) programming. Theory, algorithms, applications: The state of the art*, Proceedings of the First International Workshop on Global Constrained Optimization and Constraint Satisfaction (Cocos'02), Valbonne Sophia Antipolis, France, pp. 2–4, 2002.

[82] T. Pham Dinh, H. A. Le Thi, F. Akoa, *Combining DCA (DC Algorithms) and interior point techniques for large-scale nonconvex quadratic programming*, Optim. Methods Softw. **23** (2008), 609–629.

[83] E. Polak, *Optimization. Algorithms and Consistent Approximations*, Springer-Verlag, New York, 1997.

[84] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, 1970.

[85] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim. **14** (1976), 877–898.

[86] S. Z. Selim, M. A. Ismail, *K-means-type algorithms: A generalized convergence theorem and characterization of local optimality*, IEEE Trans. Pattern Anal. Mach. Intell. **6** (1984), 81–87.

[87] H. D. Sherali, J. Desai, *A global optimization RLT-based approach for solving the hard clustering problem*, J. Global Optim. **32** (2005), 281–306.

[88] J. Stoer, R. Burlisch, *Introduction to Numerical Analysis*, Third edition, Springer, New York, 2002.

[89] N. N. Tam, J.-C. Yao, N. D. Yen, *Solution methods for pseudomonotone variational inequalities*, J. Optim. Theory Appl. **138** (2008), 253–273.

[90] P. Tseng, *On linear convergence of iterative methods for the variational inequality problem*, J. Comput. Appl. Math. **60** (1995), 237–252.

[91] H. N. Tuan, *Boundedness of a type of iterative sequences in two-dimensional quadratic programming*, J. Optim. Theory Appl. **164** (2015), 234–245.

[92] H. N. Tuan, *Linear convergence of a type of DCA sequences in nonconvex quadratic programming*, J. Math. Anal. Appl. **423** (2015), 1311–1319.

[93] H. N. Tuan, *DC Algorithms and Applications in Nonconvex Quadratic Programing,* Ph.D. Dissertation, Institute of Mathematics, Vietnam Academy of Science and Technology, Hanoi, 2015.

[94] H. Tuy, *Convex Analysis and Global Optimization*, Second edition, Springer, 2016.

[95] H. Tuy, A. M. Bagirov, A. M. Rubinov, *Clustering via d.c. optimization*, In: "Advances in Convex Analysis and Global Optimization", pp. 221–234, Kluwer Academic Publishers, Dordrecht, 2001.

[96] R. Wiebking, *Selected applications of all-quadratic programming*, OR Spektrum **1** (1980), 243–249.

[97] J. Wu, *Advances in k-means Clustering: A Data Mining Thinking*, Springer-Verlag, Berlin-Heidelberg, 2012.

[98] J. Xie, S. Jiang, W. Xie, X. Gao, *An efficient global k-means clustering algorithm*, J. Comput. **6** (2011), 271–279.

[99] H.-M. Xu, H. Xue, X.-H. Chen, Y.-Y. Wang, *Solving indefinite kernel support vector machine with difference of convex functions programming*, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), Association for the Advancement of Artificial Intelligence, pp. 2782–2788, 2017.

[100] H. Xue, Y. Song, H.-M. Xu, *Multiple indefinite kernel learning for feature selection*, Knowledge-Based Systems **191** (2020), Article 105272 (12 pages).

[101] Y. Ye, *An extension of Karmarkar's algorithm and the trust region method for quadratic programming*, In "Progress in Mathematical Programming" (N. Megiddo, Ed.), pp. 49–63, Springer, New York, 1980.

[102] Y. Ye, *On affine scaling algorithms for nonconvex quadratic programming*, Math. Program. **56** (1992), 285–300.

[103] Y. Ye, *Interior Point Algorithms: Theory and Analysis*, Wiley, New York, 1997.