January 2021

Solving And Applications Of Multi-Facility Location Problems

Anuj Bajaj
Wayne State University

SOLVING AND APPLICATIONS OF MULTI-FACILITY LOCATION PROBLEMS

by

ANUJ BAJAJ

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2021

MAJOR: APPLIED MATHEMATICS

Approved By:

____

Advisor   Date

____

____

____

DEDICATION

To my parents and sister for their constant love and support.

"It's all about loving your family"

ACKNOWLEDGEMENTS

I take this opportunity to acknowledge and express my gratitude towards my supervisor Dr. Boris Mordukhovich for entrusting upon me confidence and providing me with such a wonderful opportunity.

I am thankful for his constant encouragement and support throughout the course of the thesis, especially for the expertise, patience and useful suggestions given during the course of the research. It added considerably to my graduate experience.

I would like to thank my committee members Dr. Boris Mordukhovich, Dr. Zhimin Zhang, Dr. Fatih Celiker, Dr. Rohini Kumar, and Dr. Alper Murat for the assistance they provided at all levels of the research, it was truly overwhelming.

I am also thankful to Wayne State University, USA National Science Foundation (Directorate for Mathematical and Physical Sciences), USA Air Force Office of Scientific Research, and Australian Research Council for providing financial assistance, without which this research would seem like a distant dream.

Finally, I take this opportunity to extend my deep appreciation to my family, collaborators, and friends for all that they meant to me during the crucial times of the completion of my research.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1 INTRODUCTION AND PROBLEM FORMULATION

In the seventeenth century, Pierre de Fermat postulated a problem of finding a point that minimizes the sum of its Euclidean distances to three given points in the plane. The problem was soon solved by Evangelista Torricelli, and is now famously known as the Fermat–Torricelli problem. This problem and its extended version that constitutes a finite number of points in higher dimensions are examples of continuous single facility location problems. Over the years several generalized models of the Fermat–Torricelli type have been introduced and studied in the literature with some practical applications to facility location decisions which can be found in [1–6] and the references therein. It is to be noted that an important feature of single facility location problems and the problems studied in the aforementioned references is that only one *center/server* has to be found to serve a finitely many *demand points/customers.*

However, numerous practical applications lead to formulations of facility location problems in which more than one center must be found to serve a finite number of demand points. Such problems are referred to as *multi-facility location problems* (MFLPs). Given a finite number of demand points $a_1, \dots, a_n$ in $\mathbb{R}^d$, we consider in this thesis the facility location in which k centers $v_1, \dots, v_k$ $(1 \le k \le n)$ in $\mathbb{R}^d$ need to be found to serve these demand points by assigning each of them to its nearest center and minimizing the total distances from the centers to the assigned demand points. In the case where $k = 1$, this problem reduces to the generalized Fermat-Torricelli problem of finding a point that minimizes the sum of the distances to a finite number of given points in $\mathbb{R}^d$.

We formulate the problem under consideration in this thesis as the following problem of *mixed integer programming* with *nonsmooth* objective functions. For convenience we will use a variable $k \times d$-matrix $\mathbf{V}$ with v_i as its ith row to store the centers to be found. We will also use another variable $k \times n$-matrix $\mathbf{U} = [u_{i,j}]$ with $u_{i,j} \in \{0, 1\}$ and $\sum_{i=1}^{k} u_{i,j} = 1$ for

$j = 1, \dots, n$ to assign demand points to the centers. The set of all such matrices shall be denoted by $\mathcal{U}$. Note that $u_{i,j} = 1$ if the center v_i is assigned to the demand point a_j while $\sum_{i=1}^k u_{i,j} = 1$ means that the demand point a_j is assigned to only one center. Our goal is to solve the constrained optimization problem formulated as follows:

$$\begin{aligned} &\text{minimize } \mathcal{F}(\mathbf{U}, \mathbf{V}) := \textstyle\sum_{i=1}^k \sum_{j=1}^n u_{i,j}^2 \|a_j - v_i\| \\ &\text{subject to } \mathbf{U} \in \mathcal{U} \ \text{ and } \ \mathbf{V} \in \mathbb{R}^{k \times d}. \end{aligned} \tag{1.1}$$

Taking into account that $u_{i,j} \in \{0, 1\}$, it is convenient to use $u_{i,j}^2$ instead of $u_{i,j}$ in the definition of the objective function $\mathcal{F}$ as seen in Section 4.2.

It is noteworthy that a similarly looking problem was considered by An, Minh and Tao [7] for different purposes. However, the main difference between our problem (1.1) and the one from [7] is that in [7] the squared Euclidean norm is used instead of the Euclidean norm in our formulation. From practical standpoint this difference is significant; namely, using the Euclidean norm allows us to model the total distance in *supply delivery*, while using the squared Euclidean norm is meaningful in *clustering*. Mathematically, these two problems are essentially different as well. In addition to the challenging discrete nature and nonconvexity that both problems share, the objective function of our multi-facility location problem (1.1) is *nondifferentiable* in contrast to [7]. This is a yet another serious challenge from both theoretical and algorithmic viewpoints. It is also to be observed that for $k = 1$ our problem becomes the aforementioned generalized Fermat-Torricelli problem that does not have a closed-form solution, while the problem considered in [7] reduces to the standard problem of minimizing the sum of squares of the Euclidean distances to the demand points. The latter has a simple closed form solution given by the aggregate of the data points.

In this thesis, we develop the following algorithmic procedure to solve the formulated nonsmooth problem (1.1) of mixed integer programming:

(i) Employ *Nesterov's smoothing* to approximate the nonsmooth objective function in

(1.1) by a family of *smooth* functions, which are represented as *differences of convex (DC)* functions.

(ii) Enclose the obtained smooth discrete problems into constrained problems of *continuous DC optimization* and then approximate them by unconstrained ones using a *penalty function.*

(iii) Solve the latter class of problems by developing an suitable modification of the algorithm for minimizing differences of convex functions known as the *DCA*.

As a consequence of all the three steps mentioned above, we propose a *new algorithm* for solving the class of multi-facility local problems of type (1.1), verify its efficacy and numerical implementation on both artificial and real data sets.

The early developments on the DCA trace back to the work by Tao in 1986 with more recent results presented in [7–10,14,15], and the bibliographies therein. Moreover, Nesterov's smoothing technique was introduced in his seminal paper [11] and was further developed and applied in many great publications [12, 13]. The combination of these two important tools provides an effective way to deal with nonconvexity and nondifferentiability in many optimization problems encountered in facility location, machine learning, compressed sensing, and imaging. It is demonstrated in this thesis in solving multi-facility location problems of type (1.1).

The present thesis is structured as follows. Chapter 2 contains the basic definitions and some preliminaries, required for understanding of the notations used. In Chapter 3 we briefly overview two versions of the DCA, discuss their convergence, and present two examples that illustrate their performances.

Chapter 4 is devoted to applying Nesterov's smoothing technique to the objective function of the multi-facility location problem (1.1) and constructing in this way a smooth approximation of the original problem by a family of DC ones. Further, we reduce the latter smooth DC problems of discrete constrained optimization to unconstrained problems by using an appropriate penalty function method. Finally, the obtained discrete optimization problems are enclosed into the DC framework of unconstrained continuous optimization.

In Chapter 5 we propose a new algorithm to solve the multi-facility location problem (1.1) by applying the updated version of the DCA taken from Chapter 3 to the smooth DC problems of continuous optimization constructed in Chapter 4. Additionally, the proposed algorithm is implemented to solve several multi-facility problems arising in practical modeling. Finally, Chapter 6 summarizes the obtained results and discusses some directions for future work.

Chapter 2 BACKGROUND

2.1 BASIC NOTATION AND DEFINITIONS

In this chapter, we define some basic terms that are required to understand this thesis. We also provide a previously known result concerning Fenchel conjugate of a convex function.

Definition 2.1. A function $\varphi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is said to be *proper* if $\operatorname{dom}(\varphi) \neq \emptyset$ and $\varphi(x) \neq -\infty$ for all $x \in \mathbb{R}^d$.

Definition 2.2. A function $\varphi : \mathbb{R}^d \to \mathbb{R} \cup \{-\infty, +\infty\}$ is said to be *lower semicontinuous* (l.s.c.) if,

$$\liminf_{x \to \bar{x}} \varphi(x) \geq \varphi(\bar{x}) \quad \text{for all} \quad \bar{x} \in \mathbb{R}^d.$$

Definition 2.3. A function $\varphi \colon \mathbb{R}^d \to \overline{\mathbb{R}} := (-\infty, \infty]$ is *coercive* if

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

Definition 2.4. A set $\Omega \subset \mathbb{R}^d$ is said to be a *convex set* if

$$\lambda x + (1 - \lambda) y \in \Omega$$

for all $x \neq y \in \Omega$, $\lambda \in [0, 1]$.

Definition 2.5. A function $\varphi : \mathbb{R}^d \to \mathbb{R} \cup \{-\infty, +\infty\}$ is said to be a *convex function* if $\operatorname{dom}(\varphi)$ is a convex set and

$$\varphi(\lambda x + (1 - \lambda) y) \leq \lambda \varphi(x) + (1 - \lambda) \varphi(y)$$

for all $x \neq y \in \operatorname{dom}(\varphi)$, $\lambda \in [0, 1]$.

Definition 2.6. A function $\varphi \colon \mathbb{R}^d \to (-\infty, \infty]$ is said to be $\gamma-$*convex* with a given modulus $\gamma \geq 0$ if the function

$$\varphi(x) - \frac{\gamma}{2} \|x\|^2 \quad \text{for all} \quad x \in \mathbb{R}^d$$

is convex.

Definition 2.7. A function $\varphi\colon \mathbb{R}^d \to (-\infty, \infty]$ is said to be *strongly convex* on $\mathbb{R}^d$, if there exists $\gamma > 0$ such that φ is $\gamma-$convex.

Definition 2.8. Consider the difference of two convex functions $g-h$ on a finite-dimensional space and assume that $g\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ is extended-real-valued while $h\colon \mathbb{R}^d \to \mathbb{R}$ is real-valued on $\mathbb{R}^d$. Then a general problem of *DC optimization* is defined by:

$$\text{minimize } f(x) := g(x) - h(x), \quad x \in \mathbb{R}^d. \tag{2.1}$$

Note that problem (2.1) is written in the unconstrained format, but—due to the allowed infinite value for g—it actually contains the domain constraint $x \in \text{dom}\,(g) := \{u \in \mathbb{R}^d \mid g(u) < \infty\}$. Furthermore, the explicit constraints of the type $x \in \Omega$ given by a nonempty convex set $\Omega \subset \mathbb{R}^d$ can be incorporated into the format of (2.1) via the indicator function $\delta_\Omega(x)$ of Ω, which equals 0 for $x \in \Omega$ and ∞ otherwise. The representation $f = g - h$ is called a *DC decomposition* of f. Note that the class of DC functions is fairly large and include many nonconvex functions important in optimization. We refer the reader to the recent book [17] with the commentaries and bibliographies therein for various classes of nonconvex optimization problems that can be represented in the DC framework (2.1).

Definition 2.9. For a nonempty set $\Omega \subset \mathbb{R}^d$ and a point $x \in \mathbb{R}^d$, The *Euclidean distance* of x onto Ω, is the set, denoted by $d(x;\Omega)$, defined as

$$d(x;\Omega) := \inf\big\{\|x-w\| \bigm| w \in \Omega\big\}. \tag{2.2}$$

Definition 2.10. For a nonempty set $\Omega \subset \mathbb{R}^d$ and a point $x \in \mathbb{R}^d$, The *Euclidean projection* of x onto Ω, is the set, denoted by $P(x;\Omega)$, defined as

$$P(x;\Omega) := \text{argmin}\{\|x-w\| : w \in \Omega\}, \tag{2.3}$$

i.e., it is the set of all those points in Ω that are closest to x in terms of the Euclidean distance.

Note that $P(x;\Omega) \neq \emptyset$ for closed sets Ω while being always a singleton if the set Ω is convex.

Definition 2.11. Given a function φ (not necessarily convex), the *convex conjugate* (commonly known as the Fenchel Conjugate) of φ, denoted by φ^*, is defined as

$$\varphi^*(v) = \sup_{x\in\mathbb{R}^d}\{\langle v, x\rangle - \varphi(x)\}.$$

Noe that if φ is *proper*, its Fenchel conjugate $\varphi^*\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ is automatically convex.

Definition 2.12. Given a (convex) function φ and a point $\bar{x} \in \operatorname{dom}(\varphi)$. The *subdifferential* of $\varphi\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ at $\bar{x}$, denoted by $\partial\varphi(\bar{x})$, is the set of subgradients given by

$$\partial\varphi(\bar{x}) := \big\{v \in \mathbb{R}^d \;\big|\; \langle v, x-\bar{x}\rangle \le \varphi(x) - \varphi(\bar{x}) \text{ for all } x \in \mathbb{R}^d\big\}. \tag{2.4}$$

Noe that if $\bar{x} \notin \operatorname{dom}(\varphi)$, we let $\partial\varphi(\bar{x}) := \emptyset$. Recall that for functions φ that are differentiable at $\bar{x}$ with the gradient given by $\nabla\varphi(\bar{x})$ we have $\partial\varphi(\bar{x}) = \{\nabla\varphi(\bar{x})\}$.

Definition 2.13. A vector $\bar{x} \in \mathbb{R}^d$ is a *stationary point* of the DC function f from (2.1) if

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

Definition 2.14. Given variable matrix $\mathbf{U} \in \mathbb{R}^{k\times n}$ as in the optimization problem (1.1), the *Frobenius norm* on $\mathbb{R}^{k\times n}$ is defined by

$$\|\mathbf{U}\|_F := \sqrt{\sum_{i=1}^{k}\sum_{j=1}^{n}|u_{i,j}|^2}. \tag{2.5}$$

Fact 2.15. *[20, Proposition 2.35, Page 56] For a convex function $\varphi : \mathbb{R}^n \to \mathbb{R}$, the following are equivalent:*

(i) φ is minimized at x over $\mathbb{R}^n$, i.e., $\varphi(y) \ge \varphi(x)$ for all $y \in \mathbb{R}^n$.

(ii) $0 \in \partial\varphi(x)$.

Note that the condition $0 \in \partial\varphi(x)$ is the generalization of the usual stationary condition $\nabla\varphi(x) = 0$ of the smooth case.

Finally in this chapter, we present the following proposition that gives a two-sided relationship between the Fenchel conjugates and subgradients of convex functions. Proposition 2.16 appears in [16, Corollary 1.4.4, Page 221], we recreate (and slightly expand) the proof for completeness.

Proposition 2.16. *Let $\varphi\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ be a proper, l.s.c, and convex function. Then $v \in \partial\varphi^*(y)$ if and only if*

$$v \in \operatorname{argmin}\big\{\varphi(x) - \langle y, x\rangle \;\big|\; x \in \mathbb{R}^d\big\}. \tag{2.6}$$

Furthermore, we have that $w \in \partial\varphi(x)$ if and only if

$$w \in \operatorname{argmin}\big\{\varphi^*(y) - \langle x, y\rangle \;\big|\; y \in \mathbb{R}^d\big\}. \tag{2.7}$$

Proof. To justify the first assertion, suppose that (2.6) is satisfied which yields $0 \in \partial\psi(v)$, where $\psi(x) := \varphi(x) - \langle y, x\rangle$ as $x \in \mathbb{R}^d$. It tells us that

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

and hence $y \in \partial\varphi(v)$, which is equivalent to $v \in \partial\varphi^*(y)$ due to the biconjugate relationship $\varphi^{**} = \varphi$ valued under the assumptions made.

Conversely, assuming $v \in \partial\varphi^*(y)$ gives us by the proof above that $0 \in \partial\psi(v)$, which clearly yields (2.6) and thus justifies the first assertion.

Next we justify the second assertion, suppose that (2.7) holds which gives $0 \in \partial\psi(w)$, where $\psi(y) := \varphi^*(y) - \langle x, y\rangle$ as $y \in \mathbb{R}^d$. This clearly implies that

$$0 \in \partial\varphi^*(w) - x,$$

and hence $x \in \partial\varphi^*(w)$, which is equivalent to $w \in \partial\varphi(x)$ due to the biconjugate relationship. The proof of the reverse implication in (2.7) is similar to the one given above. □

Chapter 3 DCA OVERVIEW AND SOME EXAMPLES

In this chapter we present two algorithms of the DCA type to solve DC problems (2.1) while referring the reader to [14, 15] for more details and further developments. We also present some convergence results and provide numerical examples illustrating both algorithms.

3.1 THE GENERIC DCA

In this section, we present a generic DCA which is a simple yet an effective optimization scheme for minimizing differences of convex functions. The algorithm is summarized as follows, as applied to (2.1).

Algorithm 1: DCA-1.

```
INPUT: $x_0 \in \mathbb{R}^d$, $N \in \mathbb{N}$.
for $l = 1, \ldots, N$ do
    Find $y_{l-1} \in \partial h(x_{l-1})$.
    Find $x_l \in \partial g^*(y_{l-1})$.
end for
OUTPUT: $x_N$.
```

Recall, the convex function $h\colon \mathbb{R}^d \to \mathbb{R}$ in (2.1) is real-valued on the whole space $\mathbb{R}^d$, we always have $\partial h(x) \neq \emptyset$ for all $x \in \mathbb{R}^d$. Simultaneously, the other convex function $g\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ in (2.1) is generally extended-real-valued, and so the subdifferential of its conjugate g^* may be empty. The following proposition excludes such a possibility.

Proposition 3.1 appears in [18, Proposition 1.3.8, Page 46], we recreate (and slightly expand) the proof for completeness.

Proposition 3.1. *Let $g\colon \mathbb{R}^d \to \overline{\mathbb{R}}$ be a proper, l.s.c, and convex function. If in addition g is coercive, then $\partial g^*(v) \neq \emptyset$ for all $v \in \mathbb{R}^d$.*

Proof. Since g is proper, the conjugate function g^* takes values in $(-\infty, \infty]$ being convex on $\mathbb{R}^d$. Taking into account that g is also lower semicontinuous and invoking the aforementioned biconjugate relationship, we find $w \in \mathbb{R}^d$ and $c \in \mathbb{R}$ such that

$$c + \langle w, x \rangle \leq g(x) \ \text{ for all } \ x \in \mathbb{R}^d. \tag{3.1}$$

The coercivity property of g ensures the existence of $\eta > 0$ for which

$$\|x\|\big(\|w\| + 1\big) \leq g(x) \ \text{ whenever } \ \|x\| \geq \eta.$$

It follows furthermore that

$$\sup\big\{\langle v, x \rangle - g(x) \ \big| \ \|x\| \geq \eta\big\} \leq -\|x\| \ \text{ for any } \ v \in \mathbb{R}^d.$$

By using (3.1), we arrive at the estimates

$$\sup\big\{\langle v, x \rangle - g(x) \ \big| \ \|x\| \leq \eta\big\} \leq \sup\big\{\langle v, x \rangle - \langle w, x \rangle - c \ \big| \ \|x\| \leq \eta\big\} < \infty, \quad v \in \mathbb{R}^d.$$

This tells us that $g^*(v) < \infty$, and therefore $\operatorname{dom}(g^*) = \mathbb{R}^d$. Since g^* is a convex function with finite values, it is continuous on $\mathbb{R}^d$ and hence $\partial g^*(v) \neq \emptyset$ for all $v \in \mathbb{R}^d$. □

Given a DCA, a natural question to ask is whether it has good convergence. The following result, which can be derived from [14,15], summarizes some convergence results of the DCA. Furthermore, deeper studies about the convergence of this algorithm and its generalizations involving the Kurdyka-Lojasiewicz (KL) inequality are discussed in [8,10].

Fact 3.2. *Let f be a DC function taken from (2.1), and let $\{x_l\}$ be an iterative sequence generated by Algorithm 1. The following assertions hold:*

1. *The sequence $\{f(x_l)\}$ is always monotone decreasing.*

2. *Suppose that f is bounded from below, that g is l.s.c and γ_1-convex, and that h is γ_2-convex with $\gamma_1 + \gamma_2 > 0$. If $\{x_l\}$ is bounded, then the limit of any convergent subsequence of $\{x_l\}$ is a stationary point of f.*

Having established the convergence of Algorithm 1. We now present an example to illustrate its performance and compare it to the classical gradient method [19, Definition 2.2.2, Page 58].

Example 3.3. Consider the function $f\colon \mathbb{R} \to \mathbb{R}$ given

$$f(x) := x^4 - 2x^2 + 2x - 3 \ \text{ for } \ x \in \mathbb{R}.$$

Note that f admits the DC representation given as follows

$$\begin{aligned} f &= g - h, \\ &= (x^4) - (2x^2 - 2x + 3), \end{aligned}$$

where $g(x) := x^4$ and $h(x) := 2x^2 - 2x + 3$ respectively.

To minimize f, we first apply the gradient method with constant stepsize $t > 0$. Calculating the derivative of f as

$$f'(x) = 4x^3 - 4x + 2$$

and picking an arbitrary starting point $x_0 \in \mathbb{R}$, we get the following sequence of iterates

$$x_{l+1} = x_l - t(4x_l^3 - 4x_l + 2) \ \text{ for } \ l = 0, 1, \ldots .$$

constructed by the gradient method.

Subsequently, the usage of the DC Algorithm 1 (DCA-1) gives us

$$\begin{aligned} y_l = \nabla h(x_l) &= 4x_l - 2, \\ g^*(x) &= 3(x/4)^{4/3}, \\ \nabla g^*(x) &= (x/4)^{1/3}. \end{aligned}$$

Hence the iterates of DCA-1 are given by

$$x_{l+1} = \nabla g^*(y_l) = \Big(\frac{y_l}{4}\Big)^{1/3} = \Big(\frac{4x_l - 2}{4}\Big)^{1/3} = \Big(\frac{2x_l - 1}{2}\Big)^{1/3}, \quad l = 0, 1, \ldots .$$

Figure 1 below provides the visualization and establishes that for $x_0 = 0$ and $t = 0.01$ the DCA-1 exhibits much faster convergence.

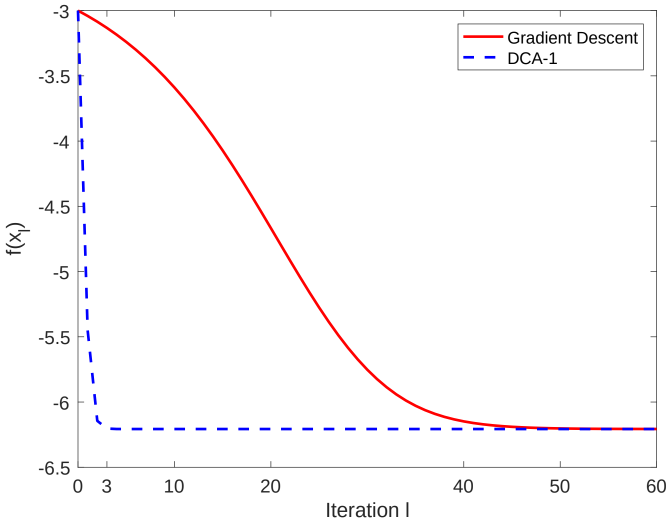


Figure 1: Convergence of the Gradient Method and DCA-1.

3.2 MODIFIED VERSION OF THE DCA

In many practical applications of Algorithm 1, for a given DC decomposition of f it is possible to find subgradient vectors from $\partial h(x_{l-1})$ based on available formulas and calculus rules of convex analysis. However, it may not be possible to explicitly compute an element of $\partial g^*(y_{l-1})$. Such a situation requires either constructing a more suitable DC decomposition of f, or finding $x_l \in \partial g^*(y_{l-1})$ by using the description of Proposition 2.16.

In this section we present the following modified version of DCA-1 and discuss its convergence.

Algorithm 2: DCA-2.

```
INPUT: $x_0 \in \mathbb{R}^d$, $N \in \mathbb{N}$
for $l = 1, \ldots, N$ do
    Find $y_{l-1} \in \partial h(x_{l-1})$
    Find $x_l$ by solving the problem:
    minimize $\varphi_l(x) := g(x) - \langle y_{l-1}, x \rangle,\ x \in \mathbb{R}^d$.
end for
OUTPUT: $x_N$
```

The following two-dimensional example illustrates the performance of the DCA-2.

Example 3.4. Consider the nonsmooth optimization problem defined by

$$\text{minimize} \;\; f(x_1, x_2) := x_1^4 + x_2^2 - 2x_1^2 - |x_2| \;\; \text{over} \;\; x = (x_1, x_2) \in \mathbb{R}^2.$$

as depicted in Figure 2 below. We observe that this function has four global minimizers, which are $(1, 0.5), (1, -0.5), (-1, 0.5)$, and $(-1, -0.5)$.

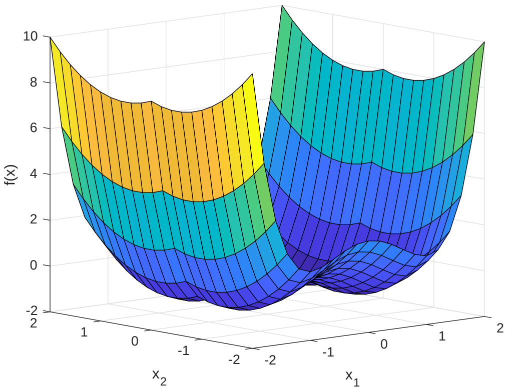


Figure 2: Graph of $f(x_1, x_2) := x_1^4 + x_2^2 - 2x_1^2 - |x_2|$.

It is easy to see that f admits a DC representation given by

$$\begin{aligned} f &= g - h, \\ &= (x_1^4 + x_2^2) - (2x_1^2 + |x_2|), \end{aligned}$$

where $g(x_1, x_2) := x_1^4 + x_2^2$ and $h(x_1, x_2) := 2x_1^2 + |x_2|$ respectively.

Moreover, the gradient $\nabla g(x)$ and the Hessian $\nabla^2 g(x)$ of $g(x)$ are given by

$$\nabla g(x) = [4x_1^3, 2x_2]^T,$$

and

$$\nabla^2 g(x) = \begin{bmatrix} 12{x_1}^2 & 0 \\ 0 & 2 \end{bmatrix}.$$

It is noteworthy that an explicit formula to calculate $\partial g^*(y_l)$ is not available. Thus, we apply the DCA-2 to solve this problem. The subdifferential of h is calculated as

$$\partial h(x) = \big[4x_1, \mathrm{sign}(x_2)\big]^T \text{ for any } x = (x_1, x_2) \in \mathbb{R}^2.$$

Having y_{l-1}, we proceed with solving the subproblem

$$\text{minimize } \varphi_l(x) := g(x) - \langle y_{l-1}, x\rangle \text{ over } x \in \mathbb{R}^2 \tag{3.2}$$

by the classical Newton method [19, Remark 2.3.5, Page 64] with $\nabla^2\varphi_l(x) = \nabla^2 g(x)$ and observe that the DCA-2 shows its superiority in convergence with different choices of initial points.

Figure 3 below presents the results of computation by using the DCA-2 with the starting point $x_0 = (-2, 2)$ and employing the Newton method with $\varepsilon = 10^{-8}$ to solve the subproblem (3.2).

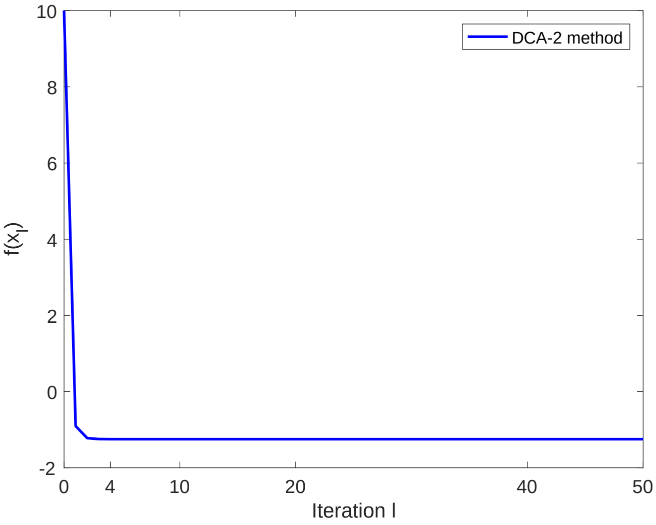


Figure 3: Convergence of the DCA-2.

Chapter 4 SMOOTH APPROXIMATION BY CONTINUOUS DC PROBLEMS

Notice that the main challenges for solving the multi-facility location problem (1.1) come from its intrinsic discrete, nonconvex, and nondifferentiable nature.

In this chapter we employ and further develop Nesterov's smoothing technique for the case of multi-facility location problem (1.1). We also enclose the family of DC mixed integer programs obtained in this way into a class of smooth DC problems of continuous optimization. The suggested procedures are efficiently justified by deriving numerical estimates expressed entirely via the given data of the original problem (1.1).

4.1 SMOOTH FORMULATION OF THE OBJECTIVE FUNCTION $\mathcal{F}$

We begin by considering the following result which is a direct consequence of Nesterov's smoothing technique given in [12] and [6, Proposition 3.1, Page 8].

Fact 4.1. *Given any $a \in \mathbb{R}^d$ and $\mu > 0$, a Nesterov smoothing approximation of the function $f\colon \mathbb{R}^d \to \mathbb{R}$ defined by*

$$f(x) := \|x - a\|, \quad x \in \mathbb{R}^d,$$

admits the smooth DC representation

$$f_\mu(x) := \frac{1}{2\mu}\|x - a\|^2 - \frac{\mu}{2}\left[d\left(\frac{x-a}{\mu}; \mathcal{B}\right)\right]^2.$$

Furthermore, we have the relationships

$$\nabla f_\mu(x) = P\left(\frac{x-a}{\mu}; \mathbb{B}\right) \text{ and } f_\mu(x) \le f(x) \le f_\mu(x) + \frac{\mu}{2},$$

where $\mathbb{B} \subset \mathbb{R}^d$ is the closed unit ball, and where P stands for the Euclidean projection given by (2.3)*.*

Using Fact 4.1 we approximate the objective function $\mathcal{F}$ in (1.1) by a smooth DC function $\mathcal{F}_\mu$ as $\mu > 0$ defined as follows:

$$\begin{aligned}\mathcal{F}_\mu(\mathbf{U}, \mathbf{V}) &:= \frac{1}{2\mu}\sum_{i=1}^{k}\sum_{j=1}^{n} u_{i,j}^2\|a_j - v_i\|^2 - \frac{\mu}{2}\sum_{i=1}^{k}\sum_{j=1}^{n} u_{i,j}^2\left[d\left(\frac{a_j - v_i}{\mu}; \mathcal{B}\right)\right]^2, \\ &= \mathcal{G}_\mu(\mathbf{U}, \mathbf{V}) - \mathcal{H}_\mu(\mathbf{U}, \mathbf{V}),\end{aligned}$$

where $\mathcal{G}_\mu, \mathcal{H}_\mu\colon \mathbb{R}^{k\times n} \times \mathbb{R}^{k\times d} \to \mathbb{R}$ are given by

$$\begin{aligned}\mathcal{G}_\mu(\mathbf{U}, \mathbf{V}) &:= \frac{1}{2\mu}\sum_{i=1}^{k}\sum_{j=1}^{n} u_{i,j}^2\|a_j - v_i\|^2, \\ \mathcal{H}_\mu(\mathbf{U}, \mathbf{V}) &:= \frac{\mu}{2}\sum_{i=1}^{k}\sum_{j=1}^{n} u_{i,j}^2\left[d\left(\frac{a_j - v_i}{\mu}; \mathcal{B}\right)\right]^2.\end{aligned}$$

This leads us to the construction of the following family of smooth approximations of the main problem (1.1) defined as

$$\begin{aligned}&\text{minimize } \ \mathcal{F}_\mu(\mathbf{U}, \mathbf{V}) := \mathcal{G}_\mu(\mathbf{U}, \mathbf{V}) - \mathcal{H}_\mu(\mathbf{U}, \mathbf{V}) \ \text{ as } \ \mu > 0 \\ &\text{subject to } \ \mathbf{U} \in \mathcal{U} = \Delta^n \cap \{0,1\}^{k\times n} \ \text{ and } \ \mathbf{V} \in \mathbb{R}^{k\times d},\end{aligned} \tag{4.1}$$

where Δ^n is the the nth Cartesian degree of the $(k-1)$-simplex $\Delta := \{y \in [0,1]^k \mid \sum_{i=1}^k y_i = 1\}$, which is a subset of $\mathbb{R}^k$.

4.2 WELL-POSEDNESS AND CONTINUOUS FORMULATION

We observe that for each $\mu > 0$ problem (4.1) is of *discrete optimization*, while our intention is to convert it to a family of problems of *continuous optimization* for which we are going to develop and implement a DCA-based algorithm in Chapter 5.

In this section we derive two results, which justifies such a reduction. The following Theorem 4.2 allows us to verify the *existence* of optimal solutions to the constrained optimization problems that appear in this procedure. It is required for having well-posedness of the algorithm construction.

Theorem 4.2. *Let* $(\overline{\mathbf{U}}, \overline{\mathbf{V}})$ *be an optimal solution to problem* (4.1). *Then for any* $\mu > 0$ *we have* $\overline{\mathbf{V}} \in \mathcal{B}$*, where* $\mathcal{B} := \prod_{i=1}^k B_i$ *is the Cartesian product of the* k *Euclidean balls* B_i *centered at* $0 \in \mathbb{R}^d$ *with radius* $r := \sqrt{\sum_{j=1}^n \|a_j\|^2}$ *that contains the optimal centers* $\bar{v}_i$ *for each index* $i = 1, \ldots, k$.

Proof. We can clearly rewrite the objective function in (4.1) in the form

$$\mathcal{F}_\mu(\mathbf{U}, \mathbf{V}) = \frac{1}{2\mu} \sum_{i=1}^k \sum_{j=1}^n u_{i,j} \|a_j - v_i\|^2 - \frac{\mu}{2} \sum_{i=1}^k \sum_{j=1}^n u_{i,j} \left[d\left(\frac{a_j - v_i}{\mu}; \mathcal{B} \right) \right]^2 \tag{4.2}$$

due to interchangeability between $u_{i,j}^2$ and $u_{i,j}$.

Observe that $\mathcal{F}_\mu(\mathbf{U}, \mathbf{V})$ is differentiable on $R^{k\times n} \times \mathbb{R}^{k\times d}$. Employing Fact 2.15 with respect to V gives us $\nabla_{\mathbf{V}} \mathcal{F}_\mu(\overline{\mathbf{U}}, \overline{\mathbf{V}}) = 0$. To calculate this partial gradient, we need some clarification for the second term in (4.2), which is differentiable as a whole while containing the nonsmooth distance function (2.2).

The convexity of the distance function in the setting of (4.2) allows us to apply the subdifferential calculation of convex analysis [20, Theorem 2.39, Page 57] and to combine it with an appropriate chain rule to handle the composition in (4.2). Observe that the squared distance function in (4.2) is the composition of the nondecreasing convex function $\varphi(t) := t^2$ on $[0,\infty)$ and the distance function to the ball $\mathcal{B}$. Thus, the chain rule from [20, Corollary 2.62, Page 73] is applicable. Hence, we obtain $d^2(\cdot;\mathcal{B})$ is differentiable with

$$\nabla d^2(x;\mathcal{B}) = 2[x - P(x;\mathcal{B})] \text{ for } x \in \mathbb{R}^d. \tag{4.3}$$

Using (4.3), we consider the following two cases:

Case 1: $(a_j - \bar{v}_i)/\mu \in \mathcal{B}$ for the fixed indices $i \in \{1,\ldots,k\}$ and $j \in \{1,\ldots,n\}$. Then

$$\nabla d^2\Big(\frac{a_j - \bar{v}_i}{\mu};\mathcal{B}\Big) = \{0\},$$

which gives us

$$\frac{\partial \mathcal{F}_\mu}{\partial v_i}(\overline{\mathbf{U}},\overline{\mathbf{V}}) = \frac{1}{\mu}\sum_{j=1}^n \bar{u}_{i,j}(\bar{v}_i - a_j), \quad i = 1,\ldots,k,$$

for the corresponding partial derivatives of $\mathcal{F}_\mu$.

Case 2: $(a_j - \bar{v}_i)/\mu \notin \mathcal{B}$ for the fixed indices $i \in \{1,\ldots,k\}$ and $j \in \{1,\ldots,n\}$.

In this case we have

$$\begin{aligned}\frac{\partial \mathcal{F}_\mu}{\partial v_i}(\overline{\mathbf{U}},\overline{\mathbf{V}}) &= \frac{1}{2\mu}\sum_{j=1}^n \bar{u}_{i,j}2(\bar{v}_i - a_j) + \sum_{j=1}^n \bar{u}_{i,j}\Big[\frac{a_j - \bar{v}_i}{\mu} - P\Big(\frac{a_j - \bar{v}_i}{\mu};\mathcal{B}\Big)\Big]\\ &= \frac{1}{\mu}\sum_{j=1}^n \bar{u}_{i,j}(\bar{v}_i - a_j) + \sum_{j=1}^n \bar{u}_{i,j}\Big[\frac{a_j - \bar{v}_i}{\mu} - \Big(\frac{a_j - \bar{v}_i}{\|a_j - \bar{v}_i\|}\Big)\Big]\\ &= \frac{1}{\|a_j - \bar{v}_i\|}\sum_{j=1}^n \bar{u}_{i,j}(\bar{v}_i - a_j).\end{aligned}$$

Thus in both cases above it follows from the stationary condition $\nabla_{\mathbf{V}}\mathcal{F}_\mu(\overline{\mathbf{U}},\overline{\mathbf{V}}) = 0$ that

$$\bar{v}_i = \frac{\sum_{j=1}^n \bar{u}_{i,j}a_j}{\sum_{j=1}^n \bar{u}_{i,j}} \text{ for all } i = 1,\ldots,k,$$

since we have $\sum\limits_{j=1}^n \bar{u}_{i,j} > 0$ due to the nonemptiness of the clusters.

Finally, employing the classical Cauchy-Schwarz inequality leads us to the estimates

$$\|\bar{v}_i\|^2 \le \frac{\left(\sum_{j=1}^n \bar{u}_{i,j} a_j\right)^2}{\left(\sum_{j=1}^n \bar{u}_{i,j}\right)^2} \le \sum_{j=1}^n \|a_j\|^2 := r^2,$$

which therefore verify all the conclusions of the theorem. □

Our next step is to enclose each discrete optimization problem (4.1) into the corresponding one of *continuous optimization*. For the reader's convenience if no confusion arises, we keep the same notation $\mathbf{U}$ for *all* the $k \times n$-matrices without the discrete restrictions on their entries.

We now define the function $\mathcal{P}\colon \mathbb{R}^{k\times n} \to \mathbb{R}$ by

$$\mathcal{P}(\mathbf{U}) := \sum_{i=1}^k \sum_{j=1}^n u_{i,j}(1-u_{i,j}) \text{ for all } \mathbf{U} \in \mathbb{R}^{k\times n},$$

and observe that this function is concave on $\mathbb{R}^{k\times n}$ with $\mathcal{P}(\mathbf{U}) \ge 0$ whenever $\mathbf{U} \in \Delta^n$. Furthermore, we have the representations

$$\mathcal{U} = \big\{\mathbf{U} \in \Delta^n \bigm| \mathcal{P}(\mathbf{U}) = 0\big\} = \big\{\mathbf{U} \in \Delta^n \bigm| \mathcal{P}(\mathbf{U}) \le 0\big\} \tag{4.4}$$

for the set of feasible $k \times n$-matrices $\mathcal{U}$ in the original problem (1.1). Employing further the standard *penalty function* method [7, Theorem 1, Page 392] allows us to eliminate the most involved constraint on $\mathbf{U}$ in (4.4) given by the function $\mathcal{P}$. Taking the penalty parameter $\alpha > 0$ sufficiently large and using the smoothing parameter $\mu > 0$ sufficiently small, we consider the following family of continuous optimization problems:

$$\begin{aligned} &\text{minimize } \mathcal{F}_\mu(\mathbf{U},\mathbf{V}) + \alpha\mathcal{P}(\mathbf{U}) = \mathcal{G}_\mu(\mathbf{U},\mathbf{V}) - \mathcal{H}_\mu(\mathbf{U},\mathbf{V}) + \alpha\mathcal{P}(\mathbf{U}) \\ &\text{subject to } \mathbf{U} \in \Delta^n \text{ and } \mathbf{V} \in \mathcal{B}. \end{aligned} \tag{4.5}$$

Observe that Theorem 4.2 ensures the existence of feasible solutions to problem (4.5) and hence optimal solutions to this problem by the classical Weierstrass theorem due to the continuity of the objective functions therein and the compactness of the constraints sets Δ^n

and $\mathcal{B}$.

We now introduce yet another parameter $\rho > 0$ ensuring a DC representation of the objective function in (4.5) as follows:

$$\begin{aligned}
\mathcal{F}_\mu(\mathbf{U},\mathbf{V}) + \alpha\mathcal{P}(\mathbf{U}) &= \frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2 - \left(\frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2 - \mathcal{F}_\mu(\mathbf{U},\mathbf{V}) - \alpha\mathcal{P}(\mathbf{U})\right)\\
&= \frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2 - \left(\frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2 - \mathcal{G}_\mu(\mathbf{U},\mathbf{V}) + \mathcal{H}_\mu(\mathbf{U},\mathbf{V}) - \alpha\mathcal{P}(\mathbf{U})\right)\\
&=: \mathcal{G}(\mathbf{U},\mathbf{V}) - \mathcal{H}(\mathbf{U},\mathbf{V}),
\end{aligned}$$

where the function $\mathcal{G}(\mathbf{U},\mathbf{V}) := \frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2$ is obviously convex, and

$$\mathcal{H}(\mathbf{U},\mathbf{V}) := \frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2 - \mathcal{G}_\mu(\mathbf{U},\mathbf{V}) + \mathcal{H}_\mu(\mathbf{U},\mathbf{V}) - \alpha\mathcal{P}(\mathbf{U}).$$

Since $\mathcal{H}_\mu(\mathbf{U},\mathbf{V}) - \alpha\mathcal{P}(\mathbf{U})$ is also convex as $\alpha > 0$, we are going to show that for any given number $\mu > 0$ it is possible to determine the values of the parameter $\rho > 0$ such that the function $\frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2 - \mathcal{G}_\mu(\mathbf{U},\mathbf{V})$ is convex under an appropriate choice of ρ. This would yield the convexity of $\mathcal{H}(\mathbf{U},\mathbf{V})$ and therefore would justify a desired representation of the objective function in (4.5). The following result gives us a precise meaning of this statement, which therefore verifies the required reduction of (4.5) to a *DC continuous optimization.*

Theorem 4.3. *The function*

$$\mathcal{G}_1(\mathbf{U},\mathbf{V}) := \frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2 - \mathcal{G}_\mu(\mathbf{U},\mathbf{V}) \tag{4.6}$$

is convex on $\Delta^n \times \mathcal{B}$ provided that

$$\rho \geq \frac{n}{2\mu}\left[\left(1 + \frac{1}{n}\xi^2\right) + \sqrt{\left(1 + \frac{1}{n}\xi^2\right)^2 + \frac{12}{n}\xi^2}\right], \tag{4.7}$$

where $\xi := r + \underset{1\leq j\leq n}{max}\|a_j\|$ and $r := \sqrt{\sum_{j=1}^n \|a_j\|^2}$.

Proof. Consider the function $\mathcal{G}_1(\mathbf{U},\mathbf{V})$ defined in (4.6) for all $(\mathbf{U},\mathbf{V}) \in \Delta^n \times \mathcal{B}$ and deduce by elementary transformations directly from its construction that

$$\begin{aligned}
\mathcal{G}_1(\mathbf{U},\mathbf{V}) &= \frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2 - \mathcal{G}_\mu(\mathbf{U},\mathbf{V}) \\
&= \frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2 - \frac{1}{2\mu}\sum_{i=1}^{k}\sum_{j=1}^{n} u_{i,j}^2\|a_j - v_i\|^2 \\
&= \frac{\rho}{2}\|\mathbf{U}\|^2 + \frac{\rho}{2}\|\mathbf{V}\|^2 - \frac{1}{2\mu}\sum_{i=1}^{k}\sum_{j=1}^{n} u_{i,j}^2\|a_j - v_i\|^2 \\
&= \frac{\rho}{2}\sum_{i=1}^{k}\sum_{j=1}^{n} u_{i,j}^2 + \frac{\rho}{2n}\sum_{i=1}^{k}\sum_{j=1}^{n}\|v_i\|^2 - \frac{1}{2\mu}\sum_{i=1}^{k}\sum_{j=1}^{n} u_{i,j}^2\|a_j - v_i\|^2 \\
&= \sum_{i=1}^{k}\sum_{j=1}^{n} \frac{\rho}{2}u_{i,j}^2 + \frac{\rho}{2n}\|a_j - v_i\|^2 + \frac{\rho}{n}\langle a_j, v_i\rangle - \frac{\rho}{2n}\|a_j\|^2 - \frac{1}{2\mu}u_{i,j}^2\|a_j - v_i\|^2.
\end{aligned}$$

Next we define the functions $\gamma_{i,j}\colon \mathbb{R}\times\mathbb{R}^d \to \mathbb{R}$ for all $i = 1,\ldots,k$ and $j = 1,\ldots,n$ by

$$\gamma_{i,j}(u_{i,j}, v_i) := \frac{\rho}{2}u_{i,j}^2 + \frac{\rho}{2n}\|a_j - v_i\|^2 - \frac{1}{2\mu}u_{i,j}^2\|a_j - v_i\|^2 \tag{4.8}$$

and show that each of these functions are convex on the set $\{u_{i,j} \in [0,1],\ \|v_i\| \le r\}$, where $r > 0$ is taken from Theorem 4.2.

To proceed, consider the Hessian matrix of each function in (4.8) given by

$$J_{\gamma_{i,j}}(u_{i,j}, v_i) := \begin{bmatrix} \rho - \frac{1}{\mu}\|a_j - v_i\|^2 & -\frac{2}{\mu}u_{i,j}(v_i - a_j) \\ -\frac{2}{\mu}u_{i,j}(v_i - a_j) & \frac{\rho}{n} - \frac{1}{\mu}u_{i,j}^2 \end{bmatrix}$$

and calculate its determinant $\det(J_{\gamma_{i,j}}(u_{i,j}, v_i))$ by

$$\begin{aligned}
\det(J_{\gamma_{i,j}}(u_{i,j}, v_i)) &:= \left(\rho - \frac{1}{\mu}\|a_j - v_i\|^2\right)\left(\frac{\rho}{n} - \frac{1}{\mu}u_{i,j}^2\right) - \frac{4}{\mu^2}u_{i,j}^2(v_i - a_j)^T(v_i - a_j) \\
&= \frac{\rho^2}{n} - \rho\left(\frac{u_{i,j}^2}{\mu} + \frac{1}{n\mu}\|v_i - a_j\|^2\right) - \frac{3u_{i,j}^2}{\mu^2}\|v_i - a_j\|^2.
\end{aligned}$$

It follows from the well-known second-order characterization of convexity [20, Theorem 2.42, Page 60] that the function $\gamma_{i,j}(u_{i,j}, v_i)$ is convex on $\{u_{i,j} \in [0,1],\ \|v_i\| \le r\}$ if $\det(J_{\gamma_{i,j}}(u_{i,j}, v_i)) \ge 0$. Using [9, Theorem 1] gives us the estimate

$$\det\left(J_{\gamma_{i,j}}(u_{i,j}, v_i)\right) \ \ge\ \frac{\rho^2}{n} - \rho\left(\frac{1}{\mu} + \frac{1}{n\mu}\|v_i - a_j\|^2\right) - \frac{3}{\mu^2}\|v_i - a_j\|^2.$$

Furthermore, from the construction of $\mathcal{B}$ in Theorem 4.2 we get that $0 < \|v_i - a_j\| \leq \|v_i\| + \|a_j\| \leq r + \max\limits_{1\leq j\leq n} \|a_j\| =: \xi$, and therefore

$$\det\left(J_{\gamma_{i,j}}(u_{i,j}, v_i)\right) \geq \frac{\rho^2}{n} - \frac{\rho}{\mu}\left(1 + \frac{1}{n}\xi^2\right) - \frac{3}{\mu^2}\xi^2, \tag{4.9}$$

which allows us to deduce from the aforementioned condition for convexity of $\gamma_{i,j}(u_{i,j}, v_i)$ that we do have convexity if ρ satisfies the estimate (4.7). □

Chapter 5 ALGORITHM DESIGN AND IMPLEMENTATION

Based on the developments presented in the previous sections and using the established smooth DC structure of problem (4.5) with the subsequent $\rho-$parameterization of the objective function therein as $\mathcal{G}(\mathbf{U},\mathbf{V})-\mathcal{H}(\mathbf{U},\mathbf{V})$, we are now ready to propose and implement a new algorithm for solving this problem involving both DCA-2 and Nesterov's smoothing.

5.1 THE SOLUTION ALGORITHM

To proceed, let us present the problem under consideration in the equivalent *unconstrained* format by using the infinite penalty via the indicator function:

$$\begin{aligned} &\text{minimize} \quad \frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2-\mathcal{H}(\mathbf{U},\mathbf{V})+\delta_{\Delta\times\mathcal{B}}(\mathbf{U},\mathbf{V}) \\ &\text{subject to } \ (\mathbf{U},\mathbf{V})\in\mathbb{R}^{k\times n}\times\mathbb{R}^{k\times d}, \end{aligned} \tag{5.1}$$

where $\mathcal{B}$, Δ, and ρ are taken from Section Chapter 4.

We first explicitly compute the gradient of the convex function $\mathcal{H}(\mathbf{U},\mathbf{V})$ in (5.1). Denoting

$$\begin{aligned}[\mathcal{Y},\mathcal{Z}] \ &:= \nabla\mathcal{H}(\mathbf{U},\mathbf{V})=\nabla\bigg(\frac{\rho}{2}\|(\mathbf{U},\mathbf{V})\|^2-\frac{1}{2\mu}\sum_{i=1}^{k}\sum_{j=1}^{n}u_{i,j}^2\|a_j-v_i\|^2 \\ &+\frac{\mu}{2}\sum_{i=1}^{k}\sum_{j=1}^{n}u_{i,j}^2\left[d\bigg(\frac{a_j-v_i}{\mu};\mathcal{B}\bigg)\right]^2-\alpha\sum_{i=1}^{k}\sum_{j=1}^{n}u_{i,j}(1-u_{i,j})\bigg),\end{aligned}$$

we have

$$\begin{aligned} \mathcal{Y}&=\nabla\mathcal{H}_{\mathbf{U}}(\mathbf{U},\mathbf{V}), \\ \text{and} \quad \mathcal{Z}&=\nabla\mathcal{H}_{\mathbf{V}}(\mathbf{U},\mathbf{V}), \end{aligned}$$

respectively.

Thus for each $i=1,\ldots,k$ and $j=1,\ldots,n$ the $(j,i)-$entry of the matrix $\mathcal{Y}$ and the ith row of the matrix $\mathcal{Z}$ are given by

$$\begin{aligned}
\mathcal{Y}_{j,i} &:= \rho u_{i,j} - \frac{u_{i,j}}{\mu}\|a_j - v_i\|^2 + \mu u_{i,j}\left[d\left(\frac{a_j - v_i}{\mu}; \mathcal{B}\right)\right]^2 + 2\alpha u_{i,j} - \alpha,\\
\mathcal{Z}_i &:= \rho v_i - \frac{1}{\mu}\sum_{j=1}^{n} u_{i,j}^2(v_i - a_j) - \sum_{j=1}^{n} u_{i,j}^2\left[\frac{a_j - v_i}{\mu} - P\left(\frac{a_j - v_i}{\mu}; \mathcal{B}\right)\right],
\end{aligned}$$

respectively.

We now describe the proposed algorithm for solving the DC program (5.1) and hence the original problem (1.1) of multi-facility location. The symbols $\mathcal{Y}^{l-1}_{[j,:]}$ and $\mathcal{Z}^{l-1}_i$ in this description represents the jth row of the matrix $\mathcal{Y}$ and the ith row of the matrix $\mathcal{Z}$ at the lth iteration, respectively. Accordingly we use the symbols $\mathbf{U}^l_{[:,j]}$ and $\mathbf{V}^l_i$. We also recall that the Frobenius norm of the matrices in this algorithm is defined in (2.5).

Algorithm 3: Solving Multi-facility Location Problems.

```
INPUT: $\mathbf{X}$ (the dataset), $\mathbf{V}^0$ (initial centers), ClusterNum (number of clusters), $\mu > 0$,
$\beta$ (scaling parameter) $> 0$, $N \in \mathbb{N}$
INITIALIZATION: $\mathbf{U}^0$, $\varepsilon > 0$, $\mu_f$ (minimum threshold for $\mu$) $> 0$, $\alpha > 0$, $\rho > 0$,
tol (tolerance parameter) $= 1$
while tol $> \varepsilon$ and $\mu > \mu_f$
for $l = 1, 2, \ldots, N$
For $1 \le i \le k$ and $1 \le j \le n$ compute
        $\mathcal{Y}_{j,i}^{l-1} := \rho u_{i,j}^{l-1} - \frac{u_{i,j}^{l-1}}{\mu}\left\|a_j - v_i^{l-1}\right\|^2 + \mu u_{i,j}^{l-1}\left[d\left(\frac{a_j - v_i^{l-1}}{\mu};\mathcal{B}\right)\right]^2 + 2\alpha u_{i,j}^{l-1} - \alpha,$

        $\mathcal{Z}_i^{l-1} := \rho v_i^{l-1} - \frac{1}{\mu}\sum_{j=1}^{n}(u_{i,j}^{l-1})^2(v_i^{l-1} - a_j) - \sum_{j=1}^{n}(u_{i,j}^{l-1})^2\left[\frac{a_j - v_i^{l-1}}{\mu} - P\left(\frac{a_j - v_i^{l-1}}{\mu};\mathcal{B}\right)\right].$
For $1 \le i \le k$ and $1 \le j \le n$ compute
        $\mathbf{U}^l_{[:,j]} := P\left(\frac{\mathcal{Y}^{l-1}_{[j,:]}}{\rho};\Delta\right),$
        $\mathbf{V}^l_i := P\left(\frac{\mathcal{Z}_i^{l-1}}{\rho};B_i\right) = \begin{cases} \frac{\mathcal{Z}_i^{l-1}}{\rho} & \text{if } |\mathcal{Z}_i^{l-1}|| \le \rho r, \\ \frac{r\mathcal{Z}_i^{l-1}}{||\mathcal{Z}_i^{l-1}||} & \text{if } ||\mathcal{Z}_i^{l-1}|| > \rho r. \end{cases}$
end for
UPDATE:
        tol $:= \left\|[\mathbf{U}^l, \mathbf{V}^l] - [\mathbf{U}^{l-1}, \mathbf{V}^{l-1}]\right\|_F$
        $\mu := \beta\mu.$
end while
OUTPUT: $[\mathbf{U}^N, \mathbf{V}^N]$.
```

Next we employ Algorithm 3 to solve several multi-facility location problems of some practical meaning. By trial and error we verify that the values chosen for μ determine the performance of the algorithm for each data set. It can be seen that very small values of the smoothing parameter μ may prevent the algorithm from clustering, and thus we gradually decrease these values. This is done via multiplying μ by a constant $\beta \in (0,1)$ and stopping when $\mu < \mu_f$. Note also that in the implementation of our algorithm we use the standard approach of choosing $\mathbf{U}^0$ by computing the distance between the point in question and each group center $\mathbf{V}^0$ and then by classifying this point to be in the group whose center is the closest to it by assigning the value of 1, while otherwise we assign the value of 0.

We now present several numerical examples, where we find the optimal centers by using Algorithm 3 via MATLAB software. We fix in what follows the values of $\mu = 0.5$, $\beta = 0.85$, $\varepsilon = 10^{-6}$, $\mu_f = 10^{-6}$, $\alpha = 30$, and $\rho = 30$ unless otherwise stated. The objective function is the *total distance* from the centers to the assigned data points. Note that this choice of the objective function seems to be natural from practical aspects in, e.g., airline and other transportation industries, where the goal is to reach the destination via the best possible route available. This reflects minimizing the transportation cost.

5.2 NUMERICAL ILLUSTRATIONS

In the following examples we implement the standard k-means algorithm in MATLAB using the in-built function kmeans().

Example 5.1. Let us consider a data set with 14 entries in $\mathbb{R}^2$ given by

$$\mathbf{X} = \begin{bmatrix} 0 & 2 & 7 & 2 & 3 & 6 & 5 & 8 & 8 & 9 & 1 & 7 & 0 & 0 \\ 3 & 2 & 1 & 4 & 3 & 2 & 3 & 1 & 3 & 2 & 1 & 4 & 4 & 1 \end{bmatrix}^T$$

with the initial data defined by

$$\mathbf{V}^0 := \begin{bmatrix} 7.1429 & 2.2857 \\ 1.1429 & 2.5714 \end{bmatrix} \text{ is obtained from the } k\text{-means algorithm, and}$$

ClusterNum := 2.

Employing Algorithm 3, we obtain the optimal centers as depicted in Table 1 and Figure 4.

Table 1: Comparison between Algorithm 3 and k-means

Method	**Optimal Center ($\mathbf{V}^N$)**	**Cost Function**
k-means	$\begin{bmatrix} 7.1429 & 2.2857 \\ 1.1429 & 2.5714 \end{bmatrix}$	22.1637
Algorithm 3	$\begin{bmatrix} 7.2220 & 2.1802 \\ 1.1886 & 2.5069 \end{bmatrix}$	22.1352

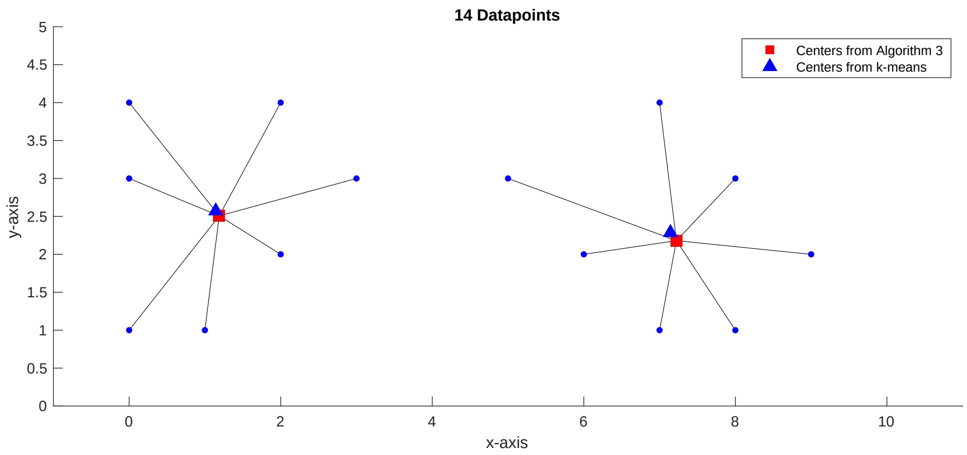


Figure 4: MFLP with 14 demand points and 2 centers.

Table 1 shows that the proposed Algorithm 3 is marginally better for the given data in comparison to the classical k-means approach in terms of the objective function.

The following example shows that the DCA and the k-means may result in both different clusters and cluster centers.

Example 5.2. Let $\mathbf{X}$ be 10 data points in $\mathbb{R}^2$ given by

$$\begin{bmatrix} 1.90 & 1.76 & 2.32 & 2.31 & 1.14 & 5.02 & 5.74 & 2.25 & 4.71 & 3.17 \\ 0.97 & 0.84 & 1.63 & 2.09 & 2.11 & 3.02 & 3.84 & 3.47 & 3.60 & 4.96 \end{bmatrix}^T.$$

With the initial data defined by taking two random data points from the data set

$$\mathbf{V}^0 := \begin{bmatrix} 1.90 & 0.97 \\ 3.17 & 4.96 \end{bmatrix} \text{chosen randomly, and}$$

$$\text{ClusterNum} := 2.$$

We obtain the optimal centers as outlined in Table 2.

Table 2: Comparison between Algorithm 3 and k-means

Method	**Optimal Center ($\mathbf{V}^N$)**	**Cost Function**
k-means	$\begin{bmatrix} 3.3943 & 2.2843 \\ 2.1867 & 3.5133 \end{bmatrix}$	16.5669
Algorithm 3	$\begin{bmatrix} 1.9995 & 1.4757 \\ 4.7185 & 3.5838 \end{bmatrix}$	9.0994

Observe from Table 2 that the proposed Algorithm 3 is better for the given data in comparison to the standard k-means algorithm. In addition, our approach gives a better approximation for the optimal center as shown in Figure 5, which coincides with the results of the Fermat-Torricelli problem [21]. It also yields a different set of clusters compared to the standard k-means algorithm as shown in Figures 6 and 7, respectively.

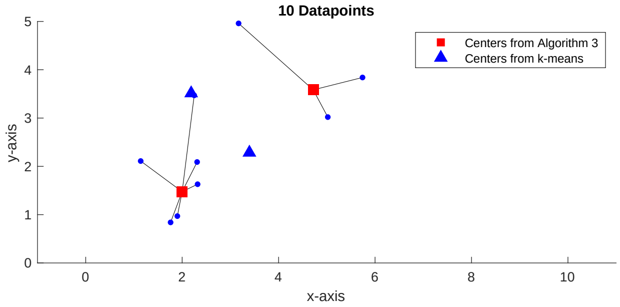
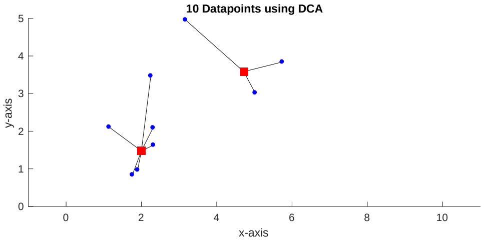


Figure 5: MFLP with 10 demand points and 2 centers.



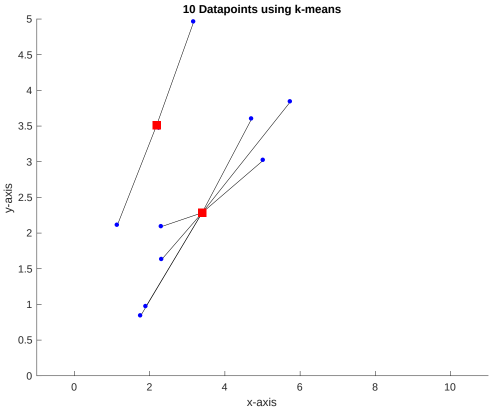


Figure 6: Clusters from Algorithm 3

Figure 7: Clusters from k-means

Note that a drawback in employing the random approach to choose the initial cluster $\mathbf{V}^0$ in Example 5.2 is the need of having prior knowledge about the data. Typically, it may not be plausible to extract such an information from large unpredictable real life datasets.

In the next example we choose the initial cluster by the process of *random selection* and see its effect on the optimal centers. Then the results obtained in this way by Algorithm 3 are compared with those computed by the k-means approach.

Example 5.3. Let $\mathbf{X}$ be 200 standard normally distributed random datapoints in $\mathbb{R}^2$, and let the initial data be given by

$$\mathbf{V}^0 := \text{randomly permuting and selecting 2 rows of } \mathbf{X}\text{, and}$$

$$\text{ClusterNum} := 2.$$

We obtain the optimal centers as outlined in Table 3.

Table 3: Comparison between Algorithm 3 and k-means

Method	**Optimal Center ($\mathbf{V}^N$)**	**Cost Function**
k-means	$\begin{bmatrix} 2.1016 & 1.2320 \\ -1.3060 & -1.0047 \end{bmatrix}$	403.3966
Algorithm 3	$\begin{bmatrix} 1.4902 & 0.7406 \\ -1.3464 & -1.0716 \end{bmatrix}$	401.7506

Observe from Table 3 that the proposed Algorithm 3 is better for the given data in comparison to the standard k-means approach. In addition, our approach gives a better approximation for the optimal center as shown in Figure 8.

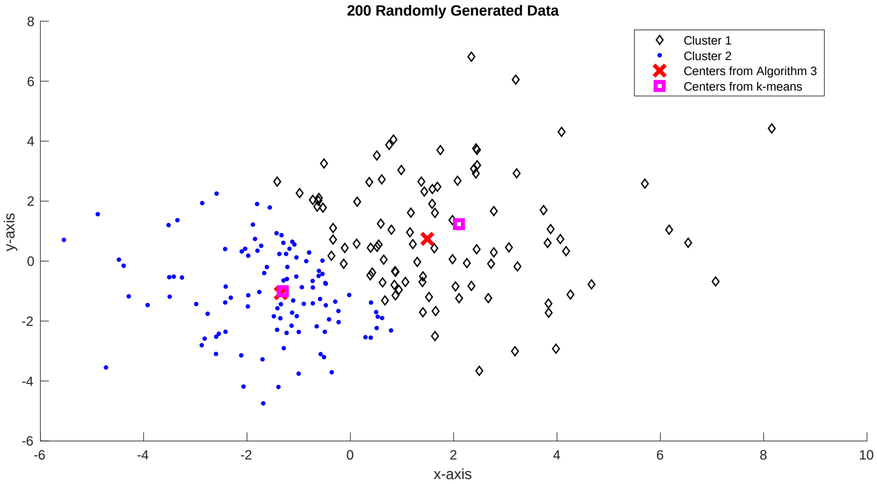


Figure 8: MFLP with 200 demand points and 2 centers.

Although, it is noteworthy that a real-life data may not be as efficiently clustered as in Example 5.3. Thus a suitable selection of the initial cluster $\mathbf{V}^0$ is vital for the convergence of the DCA based algorithms. In the next Example 5.4 we select $\mathbf{V}^0$ in Algorithm 3 by using the standard k-means method. The results achieved by our Algorithm 3 are then compared with those obtained by using the k-means approach.

Example 5.4. Consider the dataset $\mathbf{X}$ consisting of the latitudes and longitudes of 50 most populous cities in the USA[i] with

$$\mathbf{V}^0 := \begin{bmatrix} -80.9222 & 37.9882 \\ -97.8273 & 35.3241 \\ -118.3121 & 36.9535 \end{bmatrix} \text{ is obtained from the } k\text{-means algorithm, and}$$

$\text{ClusterNum} := 3.$

[i] Available at https://en.wikipedia.org/wiki/List of United States cities by population

By using Algorithm 3 we obtain the following optimal centers as given in Table 4.

Table 4: Comparison between Algorithm 3 (combined with k-means) and standard k-means

Method	**Optimal Center** $(\mathbf{V}^N)$	**Cost Function**
k-means	$\begin{bmatrix} -80.9222 & 37.9882 \\ -97.8273 & 35.3241 \\ -118.3121 & 36.9535 \end{bmatrix}$	288.8348
Algorithm 3 (combined with k-means)	$\begin{bmatrix} -81.0970 & 38.3092 \\ -97.4138 & 35.3383 \\ -119.3112 & 36.5410 \end{bmatrix}$	286.6523

We see that Algorithm 3 (combined with k-means) in which the initial cluster $\mathbf{V}^0$ is selected by using k-means method performs better in comparison to the standard k-means approach. Moreover, it gives us optimal centers as depicted in Figure 9.

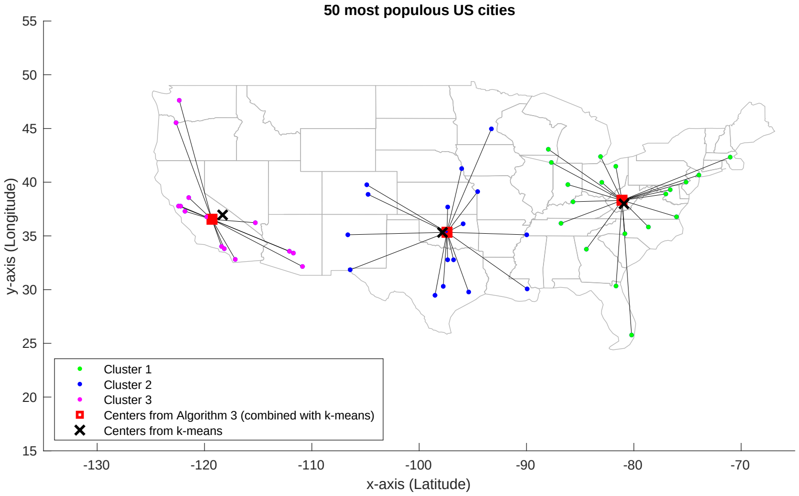


Figure 9: MFLP with 50 demand points and 3 centers.

In the next example we efficiently solve yet another multi-facility location problem by using Algorithm 3.

Example 5.5. Consider the dataset $\mathbf{X}$ in $\mathbb{R}^2$ that consists of the latitudes and longitudes of 988 US cities [22] with

$$\mathbf{V}^0 := \begin{bmatrix} -89.6747 & 41.1726 \\ -88.4834 & 30.2475 \\ -118.4471 & 35.2843 \\ -75.7890 & 40.0329 \\ -114.1897 & 43.0798 \end{bmatrix}$$ is obtained from the k-means algorithm, and

ClusterNum := 5.

The optimal centers illustrated in Figure 10 are given by

$$\mathbf{V}^N := \begin{bmatrix} -88.2248 & 40.7241 \\ -88.0154 & 30.7041 \\ -120.0735 & 33.1941 \\ -74.5328 & 38.6996 \\ -113.2341 & 42.3969 \end{bmatrix}.$$

The total transportation cost in this problem is 5089.5150.

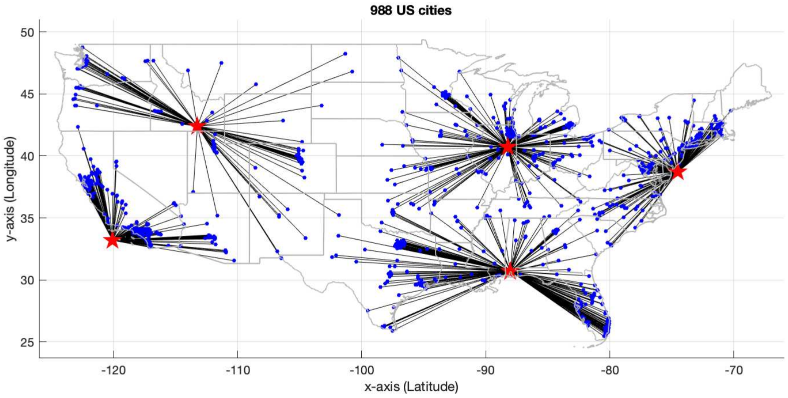


Figure 10: MFLP with 988 demand points and 5 centers.

In the last example presented in this section we efficiently solve a higher dimensional multi-facility location problem by using Algorithm 3 and compare its value of the cost function with the standard k-means algorithm.

Example 5.6. Let $\mathbf{X}$ in $\mathbb{R}^{13}$ be the *wine dataset* from the UCI Machine Learning Repositiory [23] consisting of 178 demand points. We apply Algorithm 3 with

$$\mathbf{V}^0 \text{ is obtained from the } k\text{-means algorithm, and}$$

$$\text{ClusterNum} := 3.$$

The respective total costs using Algorithm 3 and the k-means algorithm are obtained in Table 5 showing that the former algorithm performs better than the latter.

Table 5: Cost comparison between Algorithm 3 (combined with k-means) and k-means

Method	**Cost Function**
k-means	16556
Algorithm 3 (combined with k-means)	16460

Chapter 6 CONCLUSION AND FUTURE WORK

This chapter summarizes the work we have done by highlighting the research conducted on the topic: Solving and Applications of Multi-Facility Location Problems.

As detailed in Chapter 5, one of the most rewarding outcomes of the research was to be able to present a novel general algorithm to solve a class of multi-facility location problems. In particular, its implementation exhibits a better approximation compared to the classical k-means approach. This is demonstrated by a series of examples dealing with two-dimensional problems with non-negative weights, serving to cogently address the concept. This might seem a bit restrictive and leads us to a natural desire to extend the present algorithm to higher dimensions. Thus, the verification and implementation of the proposed algorithm for real-life multi-facility location problems in higher dimensions with arbitrary weights is a central direction of our future work.

Moreover, initialization and stopping criteria are fundamental aspects to any numerical simulation. Therefore, developing approaches for a suitable selection of the initial cluster and the stopping criterion different from the ones presented is a good practice and an important area to explore.

REFERENCES

[1] T. Jahn, Y.S. Kupitz, H. Martini, and C. Richter, *Minsum location extended to gauges and to convex sets*, J. Optim. Theory Appl. 166 (2015), pp. 711–746.

[2] H. Martini, K.J. Swanepoel, and G. Weiss, *The fermat-torricelli problem in normed planes and spaces*, J. Optim. Theory Appl. 115 (2002), pp. 283–314.

[3] B.S. Mordukhovich and N.M. Nam, *Applications of variational analysis to a generalized fermat-torricelli problem*, J. Optim. Theory Appl. 148 (2011), pp. 431–454.

[4] N.M. Nam and N. Hoang, *A generalized sylvester problem and a generalized fermat-torricelli problem*, J. Convex Anal. 20 (2013), pp. 669–687.

[5] N.M. Nam, N.T. An, R.B. Rector, and J. Sun, *Nonsmooth algorithms and nesterov's smoothing technique for generalized fermat-torricelli problems*, SIAM J. Optim. 24 (2014), pp. 1815–1839.

[6] N.M. Nam, R.B. Rector, and D. Giles, *Minimizing differences of convex functions with applications to facility location and clustering*, J. Optim. Theory Appl. 173 (2017), pp. 255–278.

[7] L.T.H. An, L.H. Minh, and P.D. Tao, *New and efficient DCA based algorithms for minimum sum-of-squares clustering*, Pattern Recogn. 47 (2014), pp. 388–401.

[8] N.T. An and N.M. Nam, *Convergence analysis of a proximal point algorithm for minimizing differences of functions*, Optim. 66 (2017), pp. 129–147.

[9] L.T.H. An and P.D. Tao, *Minimum sum-of-squares clustering by DC programming and DCA*, Int. Conf. Intell. Comp. (2009), pp. 327–340.

[10] L.T.H. An, H.V. Ngai, and P.D. Tao, *Convergence analysis of difference-of-convex algorithm with subanalytic data*, J. Optim. Theory Appl. 179 (2018), pp. 103–126.

[11] Yu. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$*, Soviet Math. Dokl. 269 (1983), pp. 543–547.

[12] Yu. Nesterov, *Smooth minimization of nonsmooth functions*, Math. Program. 103 (2005), pp. 127–152.

[13] Yu. Nesterov, *Lectures on Convex Optimization*, 2nd edition, Springer, Cham, Switzerland, 2018.

[14] P.D. Tao and L.T.H. An, *Convex analysis approach to D.C. programming: theory, algorithms and applications*, Acta Math. Vietnam. 22 (1997), pp. 289–355.

[15] P.D. Tao and L.T.H. An, *A D.C. optimization algorithm for solving the trust-region subproblem*, *SIAM J. Optim.* 8 (1998), pp. 476–505.

[16] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*, Springer, Berlin, 2001.

[17] B.S. Mordukhovich, *Variational Analysis and Applications*, Springer, Cham, Switzerland, 2018.

[18] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*, Springer-Verlag, New York, 1993.

[19] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms I: Fundamentals*, Springer-Verlag, New York, 1993.

[20] B.S. Mordukhovich and N.M. Nam, *An Easy Path to Convex Analysis and Applications*, Morgan & Claypool Publishers, San Rafael, 2014.

[21] B.S. Mordukhovich and N.M. Nam, *The fermat-torricelli problem and weiszfeld's algorithm in the light of convex analysis*, J. Appl. Numer. Optim. 1 (2019), 205–215.

[22] United States Cities Database, *Simple maps: geographic data products*, 2017. `http://simplemaps.com/data/us-cities`.

[23] UCI Machine Learning Repository, `https://archive.ics.uci.edu/ml/datasets/wine`

ABSTRACT

SOLVING AND APPLICATIONS OF MULTI-FACILITY LOCATION PROBLEMS

by

ANUJ BAJAJ

August 2021

Advisor:	Dr. Boris Mordukhovich
Major:	Applied Mathematics
Degree:	Doctor of Philosophy

This thesis is devoted towards the study and solving of a new class of multi-facility location problems. This class is of a great theoretical interest both in variational analysis and optimization while being of high importance to a variety of practical applications. Optimization problems of this type cannot be reduced to convex programming like, the much more investigated facility location problems with only one center. In contrast, such classes of multi-facility location problems can be described by using DC (difference of convex) programming, which are significantly more involved from both theoretical and numerical viewpoints.

In this thesis, we present a new approach to solve multi-facility location problems, which is based on mixed integer programming and algorithms for minimizing differences of convex (DC) functions. We then computationally implement the proposed algorithm on both artificial and real data sets and provide many numerical examples. Finally, some directions and insights for future work are detailed.

AUTOBIOGRAPHICAL STATEMENT

Anuj Bajaj

Education

- *Ph.D. in Applied Mathematics*, Wayne State University, USA, 2021.
- *M.A. in Mathematical Statisics*, Wayne State University, USA, 2018.
- *M.Sc in Mathematics*, The University of British Columbia, Canada, 2016.
- *M.Sc in Mathematics*, University of Delhi, India, 2014.
- *B.Sc (Hons.) in Mathematics*, University of Delhi, India, 2012.

Awards & Honors

- Nominated (by Department of Mathematics) for General Education Teaching Award, Wayne State University, 2021.
- Robert and Nancy Irvan Endowed Fellowship in Mathematics, Wayne State University, 2020.
- Nominated (by Department of Mathematics) for Garrett T. Heberlein Excellence in Teaching Awards for Graduate Students, Wayne State University, 2019.
- M.F. Janowitz Endowed Mathematics Scholarship, Wayne State University, 2019.
- University Graduate Fellowship, The University of British Columbia, 2015–16.
- Graduate Dean's Entrance Scholarship, The University of British Columbia, 2014–15.

Publications and Preprints

1. (with Boris Mordukhovich, Mau Nam Nguyen, and Tuyen Tran) *Solving a continuous multifacility location problem by DC algorithms*, Optim. Method Softw., **0** (2020), pp. 1–23.
2. (with Warren Hare, and Yves Lucet) *Visualization of the ε-subdifferential of piecewise linear-quadratic functions*, Comput. Optim. Appl., **67** (2017), pp. 421–442.
3. (with Boris Mordukhovich, Mau Nam Nguyen, and Tuyen Tran) *Solving a continuous max-min problem by DC algorithms* (*In Preparation*).
4. (with Boris Mordukhovich, and Alain Zemkoho) *DC programming algorithm for fully convex bilevel optimization* (*In Preparation*).