TRABAJO FIN DE MÁSTER

Optimización de funciones DC. El algoritmo DCA

Presentado por:

Carlos Valverde Martín

Supervisado por:

DR. RAFAEL BLANQUERO BRAVO

DR. EMILIO CARRIZOSA PRIEGO

FACULTAD DE MATEMÁTICAS

Departamento de Estadística e Investigación Operativa

Sevilla, 21 de junio de 2017

Contents

Chapter 1

Introduction

In recent years there has been a very active research in nonconvex programming, because most real life optimization problems are nonconvex. DC programming constitutes the backbone of smooth/nonsmooth nonconvex programming and global optimization.

The focus of this work is on DC Algorithm, which was first introduced by Pham Dinh Tao in 1985 as an extension of his subgradients algorithms to DC programming.

This work is structured in four chapters apart from this introduction. In Chapter 2, we provide the necessary theory about convex analysis. In Chapter 3, properties of DC sets and DC functions are introduced. Then, in Chapter 4, DC Programming and DCA are explained in detail. Finally, in Chapter 5, some applications of DCA are shown, using Python to compare results with other algorithms.

Chapter 2

Preliminaries of Convex Analysis

In this chapter we state some properties of convex functions which will be needed along this work. We can find all these properties in [7].

Definition 2.0.1. *A* ***set*** $C \subset \mathbb{R}^n$ *is* ***convex*** *if* $\forall x_1, x_2 \in C$ *and* $\lambda \in [0,1]$ *we have*

$$(1-\lambda)x_1 + \lambda x_2 \in C.$$

Definition 2.0.2. *A* ***polyhedral convex set*** $C \subset \mathbb{R}^n$ *is a set which can be expressed as the intersection of some finite collection of closed half-spaces, i.e., if* C *is of the form:*

$$C = \{x \in \mathbb{R}^n : \langle b_i, x\rangle \leq \beta_i, \quad i = 1, \dots, m\}.$$

Definition 2.0.3. *Let* $M = \{a_1, \dots, a_m\} \subset \mathbb{R}^n$ *be a finite point set. We define the* ***convex hull*** *of* M*, denoted by* $co(M)$*, as the smallest convex set that contains* M*. Expressing as a single formula, the convex hull is the set:*

$$co(M) = \left\{ \sum_{i=1}^{m} \lambda_i a_i : \lambda_i \geq 0, i = 1, \dots, m, \sum_{i=1}^{m} \lambda_i = 1 \right\}.$$

Definition 2.0.4. *Let* $f : C \subset \mathbb{R}^n \to \mathbb{R}$ *and* C *be a convex set. The function* f *is* ***convex*** *if* $\forall x_1, x_2 \in C$ *and* $\lambda \in [0,1]$*:*

$$f((1-\lambda)x_1 + \lambda x_2) \leq (1-\lambda)f(x_1) + \lambda f(x_2).$$

Definition 2.0.5. *A function* $f : C \subset \mathbb{R}^n \to \mathbb{R}$*, where* C *is a convex set, is said to be* ***strictly convex*** *if*

$$f((1-\lambda)x_1 + \lambda x_2) < (1-\lambda)f(x_1) + \lambda f(x_2), \quad \forall x_1, x_2 \in C, \quad 0 < \lambda < 1. \quad (2.1)$$

Definition 2.0.6. *Let* $\rho \geq 0$ *and* $C \subset \mathbb{R}^n$ *be a convex set. A function* $f : C \subset \mathbb{R}^n \to \mathbb{R}$ *is* ***ρ-convex*** *if*

$$f((1-\lambda)x_1 + \lambda x_2) \leq (1-\lambda)f(x_1) + \lambda f(x_2) - \frac{\lambda(1-\lambda)}{2}\rho\|x_1 - x_2\|^2,$$

for every $x_1, x_2 \in C, \quad 0 < \lambda < 1$. We also define the ***modulus of strong convexity*** *of f on C, denoted by $\rho(f, C)$ or $\rho(f)$ if $C = X$ as*

$$\rho(f, C) = \sup_{\rho \geq 0} \left\{ f - \frac{\rho}{2} \| \cdot \|^2 \text{ is convex on } C \right\}.$$

Proposition 2.0.1. *Let $C \subset \mathbb{R}^n$ be an open convex set. A twice differentiable function $f : C \to \mathbb{R}$ is convex if and only if for every $x \in C$, its Hessian matrix, $\nabla^2 f(x)$, is positive semi-definite, i.e. $u' \nabla^2 f(x) u \geq 0, \forall u \in \mathbb{R}^n$.*

Definition 2.0.7. *A convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ is* ***proper*** *if $f(x_0) < +\infty$ for at least one $x_0 \in \mathbb{R}^n$ and $f(x) > -\infty$ for every $x \in \mathbb{R}^n$.*

Definition 2.0.8. *A norm $\| \cdot \|$ in $\mathbb{R}^n$ is said to be* ***monotonic*** *in $\mathbb{R}^n_+$ if*

$$\|x\| \leq \|y\| \quad \forall x = (x_1, \ldots, x_n), \quad \forall y = (y_1, \ldots, y_n), \quad 0 \leq x_i \leq y_i, \quad i = 1, \ldots, n.$$

Definition 2.0.9. *Given a norm $\| \cdot \|$, its* ***dual norm****, $\| \cdot \|_*$, is defined as:*

$$\|x^*\|_* = \max_{x \in B} \{\langle x^*, x \rangle\}$$

where B denotes the unit ball of the norm $\| \cdot \|$.

Definition 2.0.10. *A function $\gamma : \mathbb{R}^n \to \mathbb{R}$ is a* ***gauge*** *if there exists a closed convex set B, called the unit ball of γ, with the origin in its interior such that*

$$\gamma(x) = \inf\{t > 0 : x \in tB\}, \quad x \in \mathbb{R}^n.$$

Remark 2.0.1. *In particular, norms are those gauges with compact unit ball symmetric with respect to the origin.*

Definition 2.0.11. *Given a non-empty convex set C, we define its* ***polar set****, C° as:*

$$C^\circ = \{x \in \mathbb{R}^n : \langle u, x \rangle \leq 1, \forall u \in C\}$$

Definition 2.0.12. *Given a non-empty convex set C, we define its* ***normal cone*** *at $x \in C$, $N_C(x)$ as:*

$$N_C(x) = \{x^* \in \mathbb{R}^n : \langle x^*, x \rangle \geq \langle x^*, u \rangle, \forall u \in C\}$$

Definition 2.0.13. *Let $g : \mathbb{R}^n \to \mathbb{R}$ and $x \in \mathbb{R}^n$:*

1. *For given $g \geq 0$, we define the set*

   $$\partial_\varepsilon g(x) = \{x^* \in \mathbb{R}^n : g(y) \geq g(x) + \langle x^*, y - x \rangle - \varepsilon, \quad \forall y \in \mathbb{R}^n\}$$

   to be the ***ε-subdifferential of*** *g* ***at*** *x.*
   The elements of the ε-subdifferential are called ***ε-subgradients****.*

2. *For $\varepsilon = 0$, $\partial_0 g(x)$ is called* ***subdifferential of*** g ***at*** x *, and denoted $\partial g(x)$. If $\partial g(x)$ is not empty, g is said to be* ***subdifferentiable*** *at x. Besides, we will define*

$$dom\ \partial g = \{x \in \mathbb{R}^n : \partial g(x) \neq \emptyset\}$$

and

$$range\ \partial g = \bigcup_{x \in dom\ \partial g} \partial g(x).$$

3. *Let $\mathcal{C}^0$ denote the space of continuous real-valued functions on $\mathbb{R}^n$. We call*

$$\partial^\gamma g(x) = \{\psi \in \mathcal{C}^0 : g(y) \geq g(x) + \psi(y) - \psi(x), \quad \forall y \in \mathbb{R}^n\}$$

the γ***-subdifferential of*** g ***at*** x.

Definition 2.0.14. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be an arbitrary function. The set*

$$\text{epi } f = \{(x,t) \in \mathbb{R}^n \times \mathbb{R} : f(x) - t \leq 0\} \tag{2.2}$$

is called the ***epigraph of*** f.

Definition 2.0.15. *Let $f : \mathbb{R}^n \to \mathbb{R}$ a convex function. The* ***effective domain*** *of f, which we denote by* dom f, *is the set*

$$\text{dom } f = \{x \in \mathbb{R}^n : f(x) < +\infty\}. \tag{2.3}$$

Definition 2.0.16. *A* ***function*** $\varphi : \mathbb{R}^n \to \mathbb{R}$ *is* ***closed*** *if for any closed set $F \subset \mathbb{R}^n$ the image $\varphi(F)$ is closed in $\mathbb{R}$.*

Definition 2.0.17. *A* ***function*** $f : \mathbb{R}^n \to \mathbb{R}$ *is said to be* ***lower semi-continuous*** *if*

$$f(x) \leq \lim_{n \to +\infty} f(x_n), \tag{2.4}$$

for any sequence $\{x_n\}_{n \geq 1} \subseteq \mathbb{R}^n$ that converges to x.

Chapter 3

DC Functions and DC Sets. Properties

3.1 DC Functions and DC Sets

Definition 3.1.1. *Let $C \subset \mathbb{R}^n$ be a convex set. A **function** $f : C \to \mathbb{R}$ is called **DC in** C if there exist two convex functions $g, h : C \to \mathbb{R}$ such that f can be expressed in the form*

$$f(x) = g(x) - h(x) \tag{3.1}$$

*If $C = \mathbb{R}^n$, then f is called a **DC function**. Each representation of the form (3.1) is said to be a **DC decomposition** of f.*

*We call a **function** $f : \mathbb{R}^n \to \mathbb{R}$ **locally DC** if $\forall x_0 \in \mathbb{R}^n, \exists\, \varepsilon > 0$ such that f is DC in the ball $B(x_0, \varepsilon) = \{x \in \mathbb{R}^n : \ \|x - x_0\| < \varepsilon\}$.*

The class of DC functions is very broad, since it includes, in particular:

- Convex and concave functions.
- Quadratic functions of the form $f(x) = x'Qx$, where Q is a square matrix $n \times n$. For these functions there exist two square, positive semidefinite $n \times n$ matrices A, B such that $x'Qx = x'Ax - x'Bx$, where $g(x) = x'Ax$ and $h(x) = x'Bx$ are the convex functions that appear in the DC decomposition.
- Inner product: $\langle x, y\rangle = \dfrac{1}{4}(\|x + y\|^2 - \|x - y\|^2)$

Definition 3.1.2. *A set $N \subset \mathbb{R}^n$ is said to be a **DC set** if there exist convex functions $g, h : \mathbb{R}^n \to \mathbb{R}$ such that $N = \{x \in \mathbb{R}^n : g(x) \leq 0, h(x) \geq 0\}$.*

Remark 3.1.1.

- *Using the previous definition we can write* N *as follows* $N = A \backslash B$, *where* $A = \{x \in \mathbb{R}^n : g(x) \leq 0\}$ *and* $B = \{x \in \mathbb{R}^n : h(x) < 0\}$.
- *Since* $N = \{x \in \mathbb{R}^n : g(x) \leq 0, h(x) \geq 0\}$ *can be written as the set* $N = \{x \in \mathbb{R}^n : \max[g(x), -h(x)] \leq 0\}$ *one has that a DC set can also be defined by the following DC inequality* $N = \{x \in \mathbb{R}^n : g(x) - h(x) \leq 0\}$.

Surprisingly, DC sets are not so different from arbitrary closed sets ([1]), as shown in Theorem 3.1.1 below. First, some definitions and results are needed.

Definition 3.1.3. *Let* $h : \mathbb{R}^n \to \mathbb{R}$ *be a strictly convex function and* $F \subset \mathbb{R}^n$ *a nonempty closed set. We define*

$$d^2(x) = \inf_{y \in F, x^* \in \partial h(x)} [h(y) - h(x) - \langle x^*, y - x \rangle]$$

Lemma 3.1.1. *Given a nonempty closed set* $F \subset \mathbb{R}^n$. *We have:*

1. $d(x) = 0, \forall x \in F$.
2. $d(x) > 0, \forall x \notin F$.
3. *If* $x_k \to x$ *and* $d(x_k) \stackrel{k \to \infty}{\longrightarrow} 0$, *then* $x \in F$.

Definition 3.1.4. *Let* F *be as in Lemma 3.1.1, let* $\theta > 0$ *and* $r : \mathbb{R}^n \to \mathbb{R}^+$ *be any function such that:*

- $r(x) = 0, \forall x \in F$
- $0 < r(x) \leq \min\{\theta, d(x)\}, \forall x \notin F$

We define:

$$g_F(x) = \sup_{y \notin F, x^* \in \partial h(y)} \{h(y) + \langle x^*, x - y \rangle + r^2(y)\}$$

Proposition 3.1.1. *Let* $h : \mathbb{R}^n \to \mathbb{R}$ *be any given strictly convex function. Then for every closed set* $F \subset \mathbb{R}^n$ *the function* $g_F(x)$ *is closed, convex, finite everywhere and satisfies*

$$F = \{x \in \mathbb{R}^n : g_F(x) - h(x) \leq 0\} \tag{3.2}$$

Proof:

For the proof, see [6].

□

Theorem 3.1.1. *Any closed set in $\mathbb{R}^n$ is the projection on $\mathbb{R}^n$ of a DC set in $\mathbb{R}^{n+1}$.*

Proof:

From (3.2) it follows

$$F = \{x \in \mathbb{R}^n : g_F(x) - h(x) \leq 0\} = \{x : \exists\,(x,t) \in \mathbb{R}^{n+1},\ g_F(x) \leq t,\ h(x) \geq t\}.$$

Therefore, we conclude that F is the projection of $A \backslash B \subset \mathbb{R}^{n+1}$, where $A = \{(x,t) \in \mathbb{R}^{n+1} : g_F(x) \leq t\}$ and $B = \{(x,t) \in \mathbb{R}^{n+1} : h(x) < t\}$.

□

As a consequence, we will now show that every optimization problem dealing with DC functions, i.e. every DC optimization problem can be written as an optimization problem which has a linear objective function and only two constraints, one of them being a convex inequality, the other being reverse convex, that is to say, it is expressed in the form $h(x) \geq 0$, h being convex. This type of problem is called canonical DC program. Obviously, both constraints can also be written as a DC set.

Let us study these results in more detail.

Definition 3.1.5. *A **canonical DC program** is an optimization problem of the form*

$$\begin{cases} \min & c'x \\ s.t. & g(x) \leq 0 \\ & h(x) \geq 0 \end{cases}$$

where $c \in \mathbb{R}^n$; $g, h : \mathbb{R}^n \to \mathbb{R}$ are convex.

Theorem 3.1.2. *Every DC programming problem of the form*

$$\begin{cases} \min & f_0(x) \\ s.t. & x \in R \\ & f_i(x) \leq 0 \quad i = 1, \dots, m \end{cases} \tag{3.3}$$

where R is defined by a finite system of convex inequalities $\varphi_j(x) \leq 0$, for $j \in J \subset \mathbb{N}$ and f_i, $i = 0, \dots, m$ are DC functions with DC decompositions $f_i = g_i - h_i$, $i = 0, \dots, m$, can be converted into an equivalent canonical DC program.

Proof:

By introducing an additional variable t, we see that the previous problem is equivalent to the following one:

$$\begin{cases} \min & t \\ s.t. & \varphi_j(x) \leq 0, \qquad j \in J \\ & f_i(x) \leq 0, \qquad i = 1, \ldots, m \\ & f_0(x) - t \leq 0 \end{cases}$$

Therefore, we have obtained a linear objective function. Furthermore the DC inequalities $f_0(x) - t \leq 0$ and $f_i(x) \leq 0, i = 1, \ldots, m$ can be replaced by a single one:

$$r(x,t) = \max_{i=1,\ldots,m} (f_0(x) - t, f_i(x)) \leq 0$$

We note that $r(x,t)$ is a DC function, since we can write $r(x,t)$ as

$$r(x,t) = \max\left\{g_0(x) + \sum_{i=1}^{m} h_i(x) - t, g_i(x) + \sum_{j \neq i} h_j(x), i = 1, \ldots, m\right\} - \sum_{i=0}^{m} h_i(x)$$

and thus,

$$r(x,t) = p(x,t) - q(x,t) \leq 0 \tag{3.4}$$

where $p(x,t) = \max\left\{g_0(x) + \sum_{i=1}^{m} h_i(x) - t, g_i(x) + \sum_{j \neq i} h_j(x), i = 1, \ldots, m\right\}$ and $q(x,t) = \sum_{i=0}^{m} h_i(x)$ are convex.

If we now introduce a new additional real variable z, we see that (3.4) is equivalent to the system:

$$p(x,t) - z \leq 0$$

$$q(x,t) - z \geq 0$$

The first inequality is convex and the second reverse convex.

Finally, setting the functions

$$g(x,t,z) := \max_{j \in J}(p(x,t) - z, \varphi_j(x)) \text{ and } h(x,t,z) := q(x,t) - z,$$

we see that Problem (3.3) is transformed into an equivalent canonical DC program.

□

Furthermore, as we have said before, a canonical problem can be written as an optimization problem of the form

$$\begin{cases} \min & c'x \\ s.t. & x \in N \end{cases}$$

where $N = \{x : g(x) \leq 0, h(x) \geq 0\}$ is a DC set.

3.2 Properties of DC Functions

Let us now deal with some basic properties of DC functions:

Proposition 3.2.1. *Let f and $f_i, i = 1, \ldots, m$ be DC functions. Then, the following functions are also DC:*

1. $\sum\limits_{i=1}^{m} \lambda_i f_i(x),\ \lambda_i \in \mathbb{R},\ i = 1, ..., m$
2. $\max\limits_{i=1,\ldots,m} f_i(x)$
3. $\min\limits_{i=1,\ldots,m} f_i(x)$
4. $|f(x)|$
5. $f^+(x) := \max\{0, f(x)\}$
6. $f^-(x) := \min\{0, f(x)\}$

Proof:

Note that the results above will be proved constructively. Let us assume that $f_i = g_i - h_i$ is a DC descomposition of f_i, $\forall i$.

1.

$$\begin{aligned} \sum_i \lambda_i f_i &= \left(\sum_{i:\lambda_i \geq 0} \lambda_i g_i + \sum_{i:\lambda_i < 0} (-\lambda_i) h_i \right) - \left(\sum_{i:\lambda_i \geq 0} \lambda_i h_i + \sum_{i:\lambda_i < 0} (-\lambda_i) g_i \right) \\ &= \tilde{g}(x) - \tilde{h}(x) \end{aligned}$$

with

$$\tilde{g}(x) = \sum_{i:\lambda_i \geq 0} \lambda_i g_i + \sum_{i:\lambda_i < 0} (-\lambda_i) h_i \quad \text{and} \quad \tilde{h}(x) = \sum_{i:\lambda_i \geq 0} (-\lambda_i) h_i + \sum_{i:\lambda_i < 0} \lambda_i g_i$$

convex functions, because they are a linear combination of convex functions.

2. We know that $f_i(x) = g_i(x) - h_i(x) = g_i(x) + \sum_{\substack{j=1 \\ j \neq i}}^{m} h_j(x) - \sum_{j=1}^{m} h_j(x)$.

   Since the last sum does not depend on i, one has the following expression:

$$\begin{aligned} \max_{i=1,\dots,m} f_i(x) &= \max_{i=1,\dots,m} (g_i(x) - h_i(x)) = \\ &= \max_{i=1,\dots,m} \Big[g_i(x) + \sum_{\substack{j=1 \\ j \neq i}}^{m} h_j(x) \Big] - \sum_{j=1}^{m} h_j(x) = \\ &= \tilde{g}(x) - \tilde{h}(x), \end{aligned}$$

   where $\tilde{g}(x) = \max_{i=1,\dots,m} \Big[g_i(x) + \sum_{\substack{j=1 \\ j \neq i}}^{m} h_j(x) \Big]$ and $\tilde{h}(x) = \sum_{j=1}^{m} h_j(x)$. This is a DC decomposition because the finite sum and the maximum of convex functions are convex functions.

3. Analogously,
   $\min_{i=1,\dots,m} f_i(x) = -\max_{i=1,\dots,m} -f_i(x) = -\max_{i=1,\dots,m} (h_i(x) - g_i(x))$, where g_i and h_i are convex functions.

   Hence, $-f_i$ is a DC function. Using item 2 we have $\max_{i=1\dots m} -f_i(x)$ is DC and finally we obtain $-\max_{i=1,\dots,m} -f_i(x) = \min_{i=1,\dots,m} f_i(x)$ is DC by item 1.

4. We study separately the cases $f(x) \geq 0$ and $f(x) < 0$:

   - Assume that $g(x) \geq h(x)$. Then

$$\begin{aligned} |f(x)| &= |g(x) - h(x)| = g(x) - h(x) = 2g(x) - (g(x) + h(x)) = \\ &= 2\max\{g(x), h(x)\} - (g(x) + h(x)). \end{aligned}$$

- In this case we have $g(x) < h(x)$ so,

$$\begin{aligned} |f(x)| &= |g(x) - h(x)| = h(x) - g(x) = 2h(x) - (g(x) + h(x)) = \\ &= 2\max\{g(x), h(x)\} - (g(x) + h(x)). \end{aligned}$$

Thus the DC decomposition of $|f|$ is given by the following expression $|f(x)| = 2\max\{g(x), h(x)\} - (g(x) + h(x))$.

5. Applying item 2 to the functions $f_1(x) = 0$ and $f_2(x) = f(x)$, we obtain

$$f^+(x) = \max\{0, f(x)\} = \max\{0, g(x) - h(x)\} = \max\{g, h\} - h.$$

Hence the DC decomposition for f^+ is

$$\max\{0, g - h\} = \max\{g, h\} - h \tag{3.5}$$

6. To prove that, we just use the same previous functions and item 3 to obtain

$$\min\{0, g - h\} = g - \max\{g, h\}$$

□

The next result states that any locally DC function is DC. To show this, a technical result is needed, Proposition 3.2.2.

Proposition 3.2.2. *Let $g : C \to \mathbb{R}$ be any function defined on a convex set C. If $\forall\, x \in C$, $\exists\, U$, neighbourhood of x such that g is convex on $U \cap C$, then g is convex on C.*

Proof:

For the proof, the reader is referred to [6].

□

Proposition 3.2.3. *Every locally DC function is DC.*

Proof:

In the general case, see [4].

□

Now, we give the proof for the particular case in which D is assumed to be compact.

Proposition 3.2.4. *A locally DC function on a compact convex set D is DC on D.*

Proof:

D is a compact set, so there exist $\{x_1, \ldots, x_k\} \subset D$ such that the neighbourhoods of x_i, U_i, cover $D, \forall i$. Thus, f is DC on each U_i and $f = (f + h_i) - h_i$, with $h_i : \mathbb{R} \to \mathbb{R}^n$ convex on U_i, hence $f + h_i|_{U_i}$ is convex for each i.

Let $h = \sum_{i=1}^{k} h_i$ be a convex function and consider the function $g = f + h$.

Then $g|_{U_i} = f + h|_{U_i} = f + \sum_{i=1}^{k} h_i \Big|_{U_i} = f + h_i|_{U_i} + \sum_{j \neq i} h_j \Big|_{U_i}$ is convex.

Therefore, by Proposition 3.2.2 one obtains that g is convex on D and implies that $f = g - h$ is DC.

□

Proposition 3.2.5. *Every function $f \in C^2$ is DC on any compact convex set D.*

Proof:

It suffices to show that for sufficiently large ρ the function

$$g(x) = f(x) + \frac{1}{2}\rho\|x\|^2$$

is convex on D.

Let ρ be so large that $-\min\{u'\nabla^2 f(x)u : x \in D, \|u\| = 1\} \leq \rho$.

Then:

$$\begin{aligned} u'\nabla^2 g(x)u &= u'\nabla^2(f(x) + \frac{1}{2}\rho\|x\|^2)u = u'\nabla^2 f(x)u + \rho\|u\|^2 = \\ &= u'\nabla^2 f(x)u + \rho \geq \min\{u'\nabla^2 f(x)u\} + \rho \geq -\rho + \rho = 0 \end{aligned}$$

Thus, $u'\nabla^2 g(x)u \geq 0, \forall\, u$. Hence, by Proposition 2.0.1, g is convex.

Therefore, $f(x) = g(x) - \frac{1}{2}\rho\|x\|^2$ is a DC function.

□

Proposition 3.2.6. *Every function $f \in C^2$ is DC.*

Proof:

This proposition is merely a consequence of Proposition 3.2.3.

Let us fix $x_0 \in \mathbb{R}^n$ and let consider the compact set

$$\overline{B}(x_0, \varepsilon) = \{x \in \mathbb{R}^n : \|x - x_0\| \leq \varepsilon\}.$$

Since the lowest eigenvalue is continuous, $\nabla^2 f(x)$ has the eigenvalues bounded in the compact $\overline{B}(x_0, \varepsilon)$.

Let be $\mu = -\min\{\lambda : \lambda$ is an eigenvalue of $\nabla^2 f(x)$ for some $x \in \overline{B}(x_0, \varepsilon)\}$.

As in the proof of Proposition 3.2.5 it then follows that $f(x) + \frac{1}{2}\mu\|x\|^2$ is convex on $\overline{B}(x_0, \varepsilon)$.

Hence, $f(x) = (f(x) + \frac{1}{2}\mu\|x\|^2) - \frac{1}{2}\mu\|x\|^2, \forall x \in \overline{B}(x_0, \varepsilon)$, so f is DC on $\overline{B}(x_0, \varepsilon), \forall x_0 \in \mathbb{R}^n$ which implies that f is DC on $\mathbb{R}^n$.

□

Proposition 3.2.7. *Let $D \subset \mathbb{R}^n$ be a compact convex set. Then any continuous function f on D is the limit of a sequence of DC functions on D which converges uniformly in D, in other words, for any continuous function $f : D \subset \mathbb{R}^n \to \mathbb{R}$ and for any $\varepsilon > 0$ there exists a DC function $\tilde{f} : D \subset \mathbb{R}^n \to \mathbb{R}$ such that $|f(x) - \tilde{f}(x)| \leq \varepsilon, \forall x \in D$.*

Proof:

This proposition is a consequence of the Stone-Weierstrass Theorem and the Weierstrass Approximation Theorem. Furthermore, polynomials in various variables in $\mathbb{R}^n$ are dense on continuous functions space. Therefore, we can approximate any continuous function $f : D \subset \mathbb{R}^n \to \mathbb{R}^n$ by a polynomial function which is obviously a $\mathcal{C}^2$ function and hence DC.

□

Proposition 3.2.8. *Let $D_1 \subset \mathbb{R}^n$ and $D_2 \subset \mathbb{R}^m$ be convex sets such that D_1 is open or closed and D_2 is open. If $F_1 : D_1 \to D_2$ and $F_2 : D_2 \to \mathbb{R}^k$ are DC functions, then $F_2 \circ F_1 : D_1 \to \mathbb{R}^k$ is also a DC function.*

Proof:

It suffices to show that if $F = (f_1, \dots, f_m) : D_1 \to D_2$ is DC, where $f_i : D_1 \subset \mathbb{R}^n \to \mathbb{R}$ are DC $\forall i = 1, \dots, m$ and $g : D_2 \to \mathbb{R}$ is convex then $g(f_1, \dots, f_m)$ is DC in D_1, because if we consider the DC function $F_2 = g^+ - g^-$, we have $g^+ \circ F$ and $g^- \circ F$ are DC, so

$$F_2 \circ \ F = (g^+ - g^-) \circ F = g^+ \circ F - g^- \circ F$$

is DC by Proposition 3.2.1.

Let $x \in D_1$ and $y = F(x) \in D_2$. It follows that g is a convex function, so $g(y)$ can be represented in a neighbourhood of y, U_2, as pointwise supremum of a family of affine functions:

$$g(y) = \sup_t \ell_t,$$

where $\ell_t = a_{0t} + a_{1t}y_1 + \ldots + a_{mt}y_m, y = (y_1, \dots, y_m)$ and $M = \sup\limits_{i,t} |a_{it}| < +\infty$.

Let $f_i(x) = f_i^+(x) - f_i^-(x)$ in a neighbourhood of x, U_1, satisfying that $F(U_1) \subset U_2$.

Then,

$$\begin{aligned}
\ell_t(f_1, \dots, f_m) &= a_{0t} + a_{1t}f_1 + \ldots + a_{mt}f_m = \\
&= a_{0t} + \sum_{i=1}^{m} a_{it}f_i = \\
&= a_{0t} + \sum_{i=1}^{m} a_{it}(f_i^+ - f_i^-) = \\
&= a_{0t} + \sum_{i=1}^{m} a_{it}f_i^+ - \sum_{i=1}^{m} a_{it}f_i^- = \\
&= \left[a_{0t} + \sum_{i=1}^{m}(M + a_{it})f_i^+ + \sum_{i=1}^{m}(M - a_{it})f_i^-\right] - M\sum_{i=1}^{m}(f_i^+ + f_i^-) = \\
&= p_t - q.
\end{aligned}$$

with p_t and q convex.

Therefore, $g(f_1, \ldots, f_m) = \sup_t \ell_t(f_1, \ldots, f_m) = \sup_t (p_t - q) = p - q$, i.e. $g(f_1, \ldots, f_m)$ is locally DC on D_1 and by Proposition 3.2.4, $g \circ F$ is DC on D_1.

□

Proposition 3.2.9. *Let f_i, $i = 1, ..., m$ be DC functions. Then $\prod_{i=1}^{m} f_i(x)$ is a DC function too.*

Proof:

It suffices to prove the result for two DC functions f_1 and f_2.

We consider the function

$$\begin{aligned} P \;:\; & \mathbb{R}^2 \to \mathbb{R} \\ & (s,t) \mapsto P(s,t) = st \end{aligned}$$

One has that P is a DC function with the following DC decomposition $P(s,t) = st = \frac{1}{4}(s+t)^2 - \frac{1}{4}(s-t)^2$. Then using Proposition 3.2.8 we have that $P(f_1, f_2)$ is a DC function.

□

We build now a DC decomposition for a polynomial function.

Proposition 3.2.10. *Let $f(x) = \prod_{i=1}^{m}(x - y_i)$ be, with $y_1 < \ldots < y_m$. Then f is DC on $[y_1, y_m]$ with DC decomposition*

$$f(x) = \Big(f(x) + \frac{1}{2}\mu x^2\Big) - \frac{1}{2}\mu x^2, \tag{3.6}$$

where $\mu = m(m-1)|y_m - y_1|^{m-2}$.

Proof:

For each i we have that $x - y_i$ is a affine function, so it is DC.

Furthermore using Proposition 3.2.9 we can affirm that $f(x) = \prod_{i=1}^{m}(x - y_i)$ is DC and Proposition 3.2.6 gives us the expression in (3.6) since f is a polynomial and hence a $\mathcal{C}^2$ function.

It just remains to calculate the value of μ that satisfies $f''(x)+\mu \geq 0$ (see proof of Proposition 3.2.6):

We consider $f(x)=\prod_{i=1}^{m}(x-y_i)$. Then,

$$f'(x)=\sum_{i=1}^{m}\prod_{j\neq i}(x-y_j)=\sum_{i=1}^{m}\frac{f(x)}{x-y_i}=f(x)\Big(\sum_{i=1}^{m}\frac{1}{x-y_i}\Big)$$

and

$$\begin{aligned}
f''(x) &= f(x)\Big(\sum_{i=1}^{m}\Big(\frac{-1}{(x-y_i)^2}\Big)\Big)+f'(x)\Big(\sum_{i=1}^{m}\frac{1}{x-y_i}\Big)\\
&= f(x)\Big(\sum_{i=1}^{m}\Big(\frac{-1}{(x-y_i)^2}\Big)\Big)+f(x)\Big(\sum_{i=1}^{m}\frac{1}{x-y_i}\Big)^2\\
&= f(x)\Big(-\sum_{i=1}^{m}\Big(\frac{1}{(x-y_i)^2}\Big)+\sum_{i=1}^{m}\Big(\frac{1}{(x-y_i)^2}\Big)+2\sum_{\substack{i,j=1\\ j\neq i}}^{m}\frac{1}{(x-y_i)(x-y_j)}\Big)\\
&= 2\sum_{\substack{i,j=1\\ j\neq i}}^{m}\frac{f(x)}{(x-y_i)(x-y_j)}\\
&= 2\sum_{\substack{i,j=1\\ j\neq i}}^{m}\prod_{k\neq i,j}(x-y_k)
\end{aligned}$$

Hence,

$$f''(x)=2\sum_{\substack{i,j=1\\ j\neq i}}^{m}\prod_{k\neq i,j}(x-y_k)\geq -2\sum_{\substack{i,j=1\\ j\neq i}}^{m}\prod_{k\neq i,j}|x-y_k|\geq -2\sum_{\substack{i,j=1\\ j\neq i}}^{m}\prod_{k\neq i,j}|y_m-y_1|=$$

$$-2\sum_{\substack{i,j=1\\ j\neq i}}^{m}|y_m-y_1|^{m-2}=-2\binom{m}{2}|y_m-y_1|^{m-2}=-m(m-1)|y_m-y_1|^{m-2}$$

Thus, $\mu=m(m-1)|y_m-y_1|^{m-2}$.

□

Corollary 3.2.1. *Let* $f(x) = -\prod_{i=1}^{m}(x - y_i)$ *be, with* $y_1 < \ldots < y_m$*. Then* f *is DC with DC decomposition*

$$f(x) = \frac{1}{2}\mu x^2 - \Big(f(x) + \frac{1}{2}\mu x^2\Big),$$

where $\mu = m(m-1)|y_m - y_1|^{m-2}$.

As application, we will show that the following statements can be rewritten as a DC set:

- $x_j \in \mathbb{Z}$
- $L_j \leq x_j \leq U_j$

$\forall j \in J$, where J is a set of subscripts.

These two statements are equivalents to:

$$(x_j - L_j)(x_j - (L_j+1))\ldots(x_j - (U_j - 1))(x_j - U_j) = 0 \Leftrightarrow$$

$$\Leftrightarrow \left\{ \begin{array}{lcl} (x_j - L_j)(x_j - (L_j+1))\ldots(x_j - (U_j - 1))(x_j - U_j) & \leq & 0 \\ (x_j - L_j)(x_j - (L_j+1))\ldots(x_j - (U_j - 1))(x_j - U_j) & \geq & 0 \end{array} \right. \Leftrightarrow$$

$$\Leftrightarrow \left\{ \begin{array}{lclcl} F_1(x_j) & := & (x_j - L_j)(x_j - (L_j+1))\ldots(x_j - (U_j - 1))(x_j - U_j) & \leq & 0 \\ F_2(x_j) & := & -(x_j - L_j)(x_j - (L_j+1))\ldots(x_j - (U_j - 1))(x_j - U_j) & \leq & 0 \end{array} \right.$$

$F_1(x_j)$ and $F_2(x_j)$ are the type of functions that appear in Proposition 3.2.10 and Corollary 3.2.1.

So, the DC set will be expressed as:

$$\left. \begin{array}{ll} x_j \in \mathbb{Z}, & \forall j \in J, \quad \text{where } J \text{ is a set of subscripts.} \\ L_j \leq x_j \leq U_j. & \end{array} \right\} \Leftrightarrow$$

$$\Leftrightarrow \Big\{x_j : \Big(F_1(x_j) + \frac{1}{2}\mu x_j^2\Big) - \frac{1}{2}\mu x_j^2 \leq 0\Big\} \bigcap \Big\{x_j : \frac{1}{2}\mu x_j^2 - \Big(F_2(x_j) + \frac{1}{2}\mu x_j^2\Big) \leq 0\Big\},$$

where μ has the following expression as we have stated in Proposition 3.2.10 and Corollary 3.2.1:

$$\mu = (U_j - L_j + 1)(U_j - L_j)|U_j - L_j|^{U_j - L_j - 1}.$$

3.3 Norm of a DC Function

Below, we will see some results that allow us to give a DC decomposition of the norm of f, $\|f\|$, where f is a DC function. These results are based on the Definitions 2.0.8, 2.0.9, 2.0.10 and 2.0.11 that appear in Chapter 2.

Lemma 3.3.1. *Let the norm $\|\cdot\|$ be monotonic in $\mathbb{R}^n_+$. Then*

$$\|z\| = \max_{u\in\mathbb{R}^n_+}\{\langle u, z\rangle : \|u\|_* \le 1\} \quad \forall z \in \mathbb{R}^n_+ \tag{3.7}$$

<u>Proof:</u>

First at all, observe that $(\|\cdot\|_*)_* = \|\cdot\|$, and hence, by definition we have

$$(\|z\|_*)_* = \|z\| = \max_{u\in\mathbb{R}^n}\{\langle u, z\rangle : \|u\|_* \le 1\} \quad \forall z \in \mathbb{R}^n,$$

the optimal value for the optimization problem being attained at any u, subgradient of $\|\cdot\|$ at z.

Thus, it suffices to prove that for any $z \in \mathbb{R}^n_+$, there exists some non-negative subgradient, i.e.

$$\partial\|z\| \cap \mathbb{R}^n_+ \neq \emptyset \quad \forall z \in \mathbb{R}^n_+ \tag{3.8}$$

Let $z \in \mathbb{R}^n_+$,

- Assume that z has all its components strictly positive. Let $u \in \partial\|z\|$. For any $i = 1, \ldots, n$, let e_i the unit vector with 1 in its i-th coordinate and zeroes everywhere else. For $\tau > 0$ sufficiently small, by the monotonicity of $\|\cdot\|$ and the definition of subgradient, one has

  $$0 \ge \|z - \tau e_i\| - \|z\| \ge -\tau u' e_i$$

  Thus, $u \ge 0$ and (3.8) holds.

- Now, for an arbitrary $z \in \mathbb{R}^n_+$, take a sequence $\{z_k\}$ of componentwise strictly positive vectors converging to z, and, for each k, an arbitrary $u_k \in \partial\|z_k\|$.

  Reasoning analogously as before, $u_k \ge 0$, $\forall\, k$. Moreover, the sequence $\{u_k\}$ is contained in a compact set, the dual unit ball, so it contains a subsequence converging to some u, which, by construction, is a non-negative subgradient of $\|\cdot\|$ at z.

  Hence, (3.8) holds.

□

Lemma 3.3.2. *Let $C \subset \mathbb{R}^n$ be a convex set. Let $\gamma : \mathbb{R}^m \to \mathbb{R}$ be a gauge in $\mathbb{R}^m$ with unit ball B, such that $B^\circ \subset \mathbb{R}^m_+$. Let $f = (f_1, \ldots, f_m) : C \to \mathbb{R}^m$ be a DC function with DC decomposition known $f_i = g_i - h_i$, with g_i, h_i convex. For any $i = 1, \ldots, m$, let $M_i \geq \gamma(e_i)$, where e_i is the i-th unit vector of $\mathbb{R}^m$. Then, $\gamma \circ f : C \to \mathbb{R}$ is a DC function and a DC decomposition for it is given by:*

$$\gamma \circ f = \Big(\gamma \circ f + \sum_{i=1}^{m} M_i h_i\Big) - \Big(\sum_{i=1}^{m} M_i h_i\Big)$$

Proof:

First, observe that a finite $M_i \geq \gamma(e_i)$ can be chosen, since the origin is an interior point of B and, hence, the polar set $B^\circ \subset \mathbb{R}^m_+$ is bounded.

By Theorem 14.5 of [7], every gauge γ in $\mathbb{R}^m$ can be written as a pointwise maximum of the affine functions φ_u,

$$\gamma(y) = \max_{u \in B^\circ} \varphi_u(y), \qquad \forall\, y \in \mathbb{R}^m_+$$

where $\varphi_u(y) = \langle u, y \rangle$.

Then,

$$\begin{aligned} \varphi_u(f) &= \langle u, f \rangle = \sum_{i=1}^{m} u_i f_i = \sum_{i=1}^{m} u_i (g_i - h_i) = \sum_{i=1}^{m} u_i g_i - \sum_{i=1}^{m} u_i h_i = \\ &= \Big(\sum_{i=1}^{m} u_i g_i + \sum_{i=1}^{m} (M_i - u_i) h_i\Big) - \Big(\sum_{i=1}^{m} M_i h_i\Big) \end{aligned}$$

Since $M_i \geq \gamma(e_i) = \max_{u \in B^\circ} u_i$ it follows that $\varphi_u \circ f$ can be written as the difference of two convex functions, namely $\varphi_u \circ f = p_u - q$ where

$$p_u = \sum_{i=1}^{m} u_i g_i + \sum_{i=1}^{m} (M_i - u_i) h_i \text{ and}$$

$$q = \sum_{i=1}^{m} M_i h_i$$

Therefore,

$$\begin{aligned}\gamma \circ f &= \max_{u \in B^\circ} \varphi(f_1, \ldots, f_m) = \max_{u \in B^\circ}(p_u - q) = \Big(\max_{u \in B^\circ} p_u\Big) - q = \\ &= \Big(\max_{u \in B^\circ}(\varphi_u \circ f + q)\Big) - q = \Big(\max_{u \in B^\circ}(\varphi_u \circ f) + q\Big) - q = \\ &= (\gamma \circ f + q) - q\end{aligned}$$

and the result holds.

For more details of this proof, see [2].

□

Theorem 3.3.1. *Let the norm $\|\cdot\|$ be monotonic in $\mathbb{R}^m_+$, and let define $f : \mathbb{R}^n \to \mathbb{R}^m_+$ as a non-negative DC function on the convex set $C \subset \mathbb{R}^n$, with DC decomposition $f = g - h$. Then a DC decomposition on C for $\|f\|$ is given by*

$$\|f\| = \|g - h\| = \Big(\|g - h\| + \sum_{i=1}^{m} \|e_i\| h_i\Big) - \Big(\sum_{i=1}^{m} \|e_i\| h_i\Big),$$

where, for each $i = 1, \ldots, m$, e_i is the unit vector with 1 in its $i-$th coordinate and zeroes everywhere else.

Proof:

Let γ be the gauge in $\mathbb{R}^m$ defined as

$$\gamma(z) = \max_{u \in \mathbb{R}^m_+}\{\langle u, z\rangle : \|u\|_* \leq 1\}, \quad \forall z \in \mathbb{R}^m$$

Let $B_{\|\cdot\|}$ denote the unit ball of $\|\cdot\|$. Observe that

$$\begin{aligned}(B^\circ_{\|\cdot\|} \cap \mathbb{R}^m_+)^\circ &= \{x \in \mathbb{R}^m : \langle u, x\rangle \leq 1, \forall u \in \mathbb{R}^m_+ \cap B^\circ_{\|\cdot\|}\} = \\ &= \{x \in \mathbb{R}^m : 1 \geq \max_{u \in \mathbb{R}^m_+}\{\langle u, x\rangle : \|u\|_* \leq 1\}\} = \\ &= \{x \in \mathbb{R}^m : 1 \geq \gamma(x)\} = \\ &= B_\gamma\end{aligned}$$

Hence $B^\circ_\gamma = (B^\circ_{\|\cdot\|} \cap \mathbb{R}^m_+)^{\circ\circ} = B^\circ_{\|\cdot\|} \cap \mathbb{R}^m_+ \subset \mathbb{R}^m_+$, so Lemma 3.3.2 can be applied, taking $M_i = \gamma(e_i)$, $i = 1, \ldots, m$. We conclude that $\gamma \circ f$ is a DC function with the following DC decomposition:

$$\gamma \circ f = \Big(\gamma \circ (g - h) + \sum_{i=1}^{m} \gamma(e_i) h_i\Big) - \Big(\sum_{i=1}^{m} \gamma(e_i) h_i\Big)$$

By Lemma 3.3.1 $\|z\| = \gamma(z)$, $\forall z \geq 0$ and therefore, since f is non-negative on C, $\|f\| = \gamma(f) = \gamma \circ f$ on C.

Thus, $\|f\|$ is DC with this DC decomposition on C:

$$\|f\| = \|g - h\| = \Big(\|g - h\| + \sum_{i=1}^{m} \|e_i\| h_i\Big) - \Big(\sum_{i=1}^{m} \|e_i\| h_i\Big).$$

□

Theorem 3.3.1 is valid for non-negative functions f. In arbitrary functions we have the following result of the norm of f:

Proposition 3.3.1. *Let $\|\cdot\|$ be any norm in $\mathbb{R}^m$, let be C a convex set and let define $f : C \subset \mathbb{R}^n \to \mathbb{R}^m$ as a DC function with DC decomposition $f = g - h$. Then a DC decomposition on C for $\|f\|$ is given by:*

$$\|f\| = \|g - h\| = \Big(\|g - h\| + \sum_{i=1}^{m} M_i(g_i + h_i)\Big) - \Big(\sum_{i=1}^{m} M_i(g_i + h_i)\Big), \quad (3.9)$$

where, for each $i = 1, \ldots, m$, e_i is the unit vector with 1 in its $i-$th coordinate and zeroes everywhere else, and M_i is an arbitrary constant satisfying that $M_i \geq \|e_i\|$.

Proof:

For the proof of this result the reader is referred to [2].

□

3.4 Non-uniqueness of DC Decomposition

Proposition 3.4.1. *If f is a DC function, then there are infinitely many DC decompositions.*

Proof:

If we have a DC decomposition of f, $f = g - h$, with g and h convex, another DC decomposition is obtained just adding a convex term, ϕ, since the sum of convex functions is also convex. This way, $f = (g + \phi) - (h + \phi)$.

□

Chapter 4

DC Programming. The DCA Algorithm

In this chapter, we are going to study both unconstrained and constrained DC optimization problems. We are giving an algorithm called DCA to solve this problem. Before this, we need to introduce some concepts and prove some results to understand this algorithm. We consider the Hilbert space $X = \mathbb{R}^n$ with the inner product $\langle \cdot, \cdot \rangle$ and the corresponding Euclidan norm $\| \cdot \|$. Since X is a finite-dimensional vector space, its dual X^* can be identified with X itself.

4.1 Conjugate Functions and Properties

In this section, we are studying conjugate functions, which play an important role in duality.

Notation 4.1.1. *We denote $\Gamma_0(X)$ the set of all lower semicontinuous proper convex functions on X.*

Definition 4.1.1. *Let $g \in \Gamma_0(X)$, the **conjugate function** g^* is a function defined by*

$$g^*(x^*) = \sup_{x \in X} \{ \langle x^*, x \rangle - g(x) \}.$$

Examples 4.1.1. *Now, we show some examples of conjugate functions of familiar functions:*

1. *Let $g(x) = mx + n$, $x \in \mathbb{R}$. Its conjugate function is*

$$g^*(x^*) = \sup_{x \in \mathbb{R}} \{ \langle x^*, x \rangle - (mx + n) \}.$$

Let denote $h(x) = \langle x^*, x \rangle - (mx + n)$. Then, we have that h is differentiable. Computing the derivative, we obtain $h'(x) = x^* - m$. Therefore, we have that the *supremum is attained when $x^* = m$, i.e, $g^*(m) = -n$. If $x^* > m$, $h(x) \to +\infty$ when $x \to +\infty$. If $x^* < m$, $h(x) \to +\infty$ when $x \to -\infty$.*

In summary, we have:

$$g^*(x^*) = \begin{cases} -n, & \text{if } x^* = m, \\ +\infty, & \text{otherwise.} \end{cases}$$

2. *Let $g(x) = \frac{1}{2}x'x$, $x \in X$. Its conjugate function is*

$$g^*(x^*) = \sup_{x \in X}\{\langle x^*, x\rangle - \frac{1}{2}x'x\}.$$

If we denote $h(x) = \langle x^*, x\rangle - \frac{1}{2}x'x$, we have a differentiable function. Then, we have $\nabla h(x) = \frac{1}{2}(x^* - x) = 0$ that implies $\hat{x} = x^*$ is a maximum.

It follows that

$$g^*(x^*) = \sup_{x \in X} h(x) = h(\hat{x}) = \langle x^*, \hat{x}\rangle - \frac{1}{2}\hat{x}'\hat{x} = \langle x^*, x^*\rangle - \frac{{x^*}'x^*}{2} = \frac{{x^*}'x^*}{2}.$$

Then, we have $g^* \equiv g$ in X.

3. *Let $g(x) = \|x\|$, $x \in X$. We recall that*

$$g^*(y) = \sup_{x \in X}\{\langle x^*, x\rangle - g(x)\} = \sup_{x \in X}\{\langle x^*, x\rangle - \|x\|\}.$$

We have two cases:

- *If $\|x^*\|_* \leq 1$, then*

$$\langle x^*, x\rangle \leq \|x^*\|_*\|x\| \leq \|x\|, \quad \forall x \in X.$$

Therefore, $g^*(x^*) \leq 0, \forall x \in X$, in particular, for $x = 0$, we have the equality. Hence, $g^*(x^*) = 0$, $\forall x^*$ with $\|x^*\|_* \leq 1$.

- *If $\|x^*\|_* > 1$, there exists an x_0 with $\|x_0\| \leq 1$ and $\langle x^*, x\rangle > 1$. Then,*

$$\begin{aligned} g^*(x^*) &= \sup_{x \in X}\{\langle x^*, x\rangle - \|x\|\} \\ &\geq \sup_{\lambda > 0}\{\langle x^*, \lambda x_0\rangle - \|\lambda x_0\|\} \\ &= \sup_{\lambda > 0} \lambda(\langle x^*, x_0\rangle - \|x_0\|) = +\infty. \end{aligned}$$

In summary, we have:

$$\|x^*\|^* = \begin{cases} 0, & \text{if } \|x^*\|_* \leq 1, \\ +\infty, & \text{otherwise.} \end{cases}$$

Proposition 4.1.1. *The conjugate function g^* has the following properties:*

1. *(Young's inequality)* $g(x) + g^*(x^*) \geq \langle x^*, x \rangle, \quad \forall x, x^* \in X.$
2. $g^*(0) = -\inf_{x\in X} g(x).$
3. *If $f \leq g$ then $g^* \leq f^*$.*
4. $(\sup_{i\in I} g_i)^* \leq \inf_{i\in I} g_i^*$ *and* $(\inf_{i\in I} g_i)^* \geq \inf_{i\in I} g_i^*$.
5. $(\lambda g)^*(x^*) = \lambda g^*\left(\frac{1}{\lambda}x^*\right)$, $\forall \lambda > 0.$
6. *If we consider the translation function $g_\alpha(x) = g(x-\alpha)$, α, $x \in X$. We have $g_\alpha^*(x^*) = g^*(x^*) + \langle x^*, \alpha\rangle$.*

Proof:

1. $g^*(x^*) = \sup_{x\in X}\{\langle x^*, x\rangle - g(x)\} \geq \langle x^*, x\rangle - g(x)$, $\forall x, x^* \in X$. Hence, we have the Young's inequality.
2. $g^*(0) = \sup_{x\in X}\{\langle 0, x\rangle - g(x)\} = \sup_{x\in X}\{-g(x)\} = -\inf_{x\in X}\{g(x)\}.$
3. $g^*(x^*) = \sup_{x\in X}\{\langle x^*, x\rangle - g(x)\} \geq \sup_{x\in X}\{\langle x^*, x\rangle - f(x)\} = f^*(x^*).$
4. We have

$$\begin{aligned}(\sup_{i\in I} g_i)^*(x^*) &= \sup_{x\in X}\{\langle x^*, x\rangle - \sup_{i\in I} g_i(x)\}\\ &\leq \sup_{x\in X}\{\langle x^*, x\rangle - g_i(x)\} = g_i^*(x^*), \quad \forall i \in I,\, \forall x^* \in X.\end{aligned}$$

   Therefore, $(\sup_{i\in I} g_i)^* \leq \inf_{i\in I} g_i^*$. We can obtain analogously the other inequality.

5. If $\lambda > 0$, we have

$$(\lambda g)^*(x^*) = \sup_{x\in X}\{\langle x^*, x\rangle - \lambda g(x)\} = \lambda \sup_{x\in X}\left\{\left\langle \frac{x^*}{\lambda}, x\right\rangle - g(x)\right\} = \lambda g^*\left(\frac{x^*}{\lambda}\right).$$

6. Let $\alpha \in X$. Then, we obtain

$$\begin{aligned}g_\alpha^*(x^*) &= \sup_{x\in X}\{\langle x^*, x\rangle - g_\alpha(x)\}\\ &= \sup_{x\in X}\{\langle x^*, x-\alpha\rangle - g(x-\alpha)\} + \langle x^*, x\rangle = g^*(x^*) + \langle x^*, \alpha\rangle.\end{aligned}$$

□

Theorem 4.1.1 (Hahn-Banach separation theorem in $\mathbb{R}^n$)**.** *If $C \subset \mathbb{R}^n$ closed and convex and $x_0 \notin C$, then there exists $x^* \in \mathbb{X}$, $x^* \neq 0$ that strictly separates C and x_0, i.e. there is a $c \in \mathbb{R}$ such that $\langle x^*, x\rangle < c$, $\forall x \in W$ and $\langle x^*, x_0\rangle > c$.*

Now, we are going to use this main theorem to prove next result.

Theorem 4.1.2. *The conjugate function g^* of $g \in \Gamma_0(X)$ is in $\Gamma_0(X)$.*

Proof:

- $\boxed{g^* \text{ is convex}}$ Let $x_1^*,\ x_2^* \in X$ and $\lambda \in [0,1]$. We have

$$\begin{aligned} g^*(\lambda x_1^* + (1-\lambda x_2^*)) &= \sup_{x\in X}\{\langle \lambda x_1^* + (1-\lambda)x_2^*, x\rangle - g(x)\} \\ &\leq \lambda \sup_{x\in X}\{\langle x_1^*, x\rangle - g(x)\} + (1-\lambda)\sup_{x\in X}\{\langle x_2^*, x\rangle - g(x)\} \\ &\leq \lambda g^*(x_1^*) + (1-\lambda) g^*(x_2^*). \end{aligned}$$

- $\boxed{g^* \text{ is lower semicontinuous}}$ Let be $x_n^* \to x^*$. From the Young's inequality, we have:

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

  In particular,

$$g^*(x_n^*) \geq \langle x_n^*, x\rangle - g(x), \quad \forall x \in X.$$

  Then follows

$$\liminf_{n\to+\infty} g^*(x_n^*) \geq \liminf_{n\to+\infty} \langle x_n^*, x\rangle - g(x) = \langle x^*, x\rangle - g(x), \quad \forall x \in X.$$

  Hence

$$\liminf_{n\to+\infty} g^*(x_n^*) \geq \sup_{x\in X}\{\langle x^*, x\rangle - g(x)\} = g^*(x^*),$$

  so we have that g^* is lower semicontinuous.

- $\boxed{g^* \text{ is proper}}$ We have $g \in \Gamma_0(X)$, in particular, g is a convex proper function.

  1. $g^*(x^*) > -\infty$, $\forall x^* \in X$. Because of g is proper, there exists $x \in X$ such that $g(x) < \infty$. Therefore

$$g^*(x^*) = \sup_{x\in X}\{\langle x^*, x\rangle - g(x)\} \geq \langle x^*, y\rangle - g(y) > -\infty, \quad \forall x^* \in X.$$

2. $g^* \not\equiv +\infty$. Let $C = \text{epi } g$ which is a closed convex set. For any $c > 0$ and $y \in X$ such that $f(y) < \infty$ holds $(y, g(y) - c) \notin C$ because

$$g(y) - (g(y) - c) = c > 0.$$

Then, applying Separation Theorem 4.1.1 to C and $x_0 = (y, g(y) - c)$. We obtain some $(x^*, \alpha) \in X \times \mathbb{R}$ such that

$$\begin{aligned}\langle x^*, x\rangle + \alpha t &= \langle (x^*, \alpha), (x, t)\rangle \\ &< \langle (x^*, \alpha), x_0\rangle = \langle x^*, y\rangle + \alpha(g(y) - c), \quad \forall (x, t) \in C.\end{aligned}$$

Let see that $\alpha < 0$.

- If $\alpha > 0$, the left-hand side tends to $+\infty$ when $t \to +\infty$ that contradicts the relation.
- If $\alpha = 0$, we would have $\langle x^*, y\rangle < \langle x^*, x\rangle$, $\forall x \in \text{dom } g$. But it is also a contradiction because $y \in \text{dom } g$.

Dividing by $-\alpha$ and setting $x_1^* := \frac{x^*}{-\alpha}$ and $t := g(x)$, we obtain

$$\langle x_1^*, x\rangle - g(x) < \langle x_1^*, y\rangle - g(y) + c, \quad \forall x^* \in \text{dom } g.$$

Taking supremum in this expression, we obtain that

$$g^*(x_1^*) = \sup_{x \in X}\{\langle x_1^*, x\rangle - g(x)\} \leq \langle x_1^*, y\rangle - g(y) + c < +\infty.$$

Therefore, $g^* \in \Gamma_0(X)$ as we wanted to prove. □

Remark 4.1.1. *It is not necessary that $g \in \Gamma_0(X)$ to prove that g^* is convex and lower semicontinuous. However, g must be in $\Gamma_0(X)$ to see that it is proper.*

Now, we introduce the concept of biconjugate of a function which is the following

Definition 4.1.2. *Let $g^* \in \Gamma_0(X)$, the* ***biconjugate function*** *g^{**} is a function defined by*

$$g^{**}(x) = \sup_{x^* \in X}\{\langle x^*, x\rangle - g^*(x^*)\}.$$

Examples 4.1.2. *Now, we are going to compute the biconjugate of the functions from examples 4.1.1:*

1. *Let $g^*(x^*)$ defined by*

$$g^*(x^*) = \begin{cases} -n, & \text{if } x^* = m, \\ +\infty, & \text{otherwise.}\end{cases}$$

It follows $g^{**}(x) = \sup_{x^* \in X}\{\langle x^*, x\rangle - g^*(x^*)\} = mx + n$, $\forall x \in \mathbb{R}$. Therefore $g^{**} \equiv g$.

2. *From the second example, we obtained that* $g^* \equiv g$. *Then, it is obvious that* $g^{**} \equiv g^* \equiv g$.

3. *Let* $g^*(x^*)$ *from the third example, which is defined by*

$$\|x^*\|^* = \begin{cases} 0, & \text{if } \|x^*\|_* \leq 1, \\ +\infty, & \text{otherwise.} \end{cases}$$

Then, we have $g^{**}(x) = \sup_{x^* \in X}\{\langle x^*, x\rangle - g^*(x^*)\} = \max_{x^* \in X}\{\langle x^*, x\rangle \,,\, \|x^*\|_* \leq 1\}$. *Consequently, we conclude that* $g^{**} \equiv g$, *namely,* $\|\cdot\|^{**} \equiv \|\cdot\|$.

All of the examples above are functions in $\Gamma_0(X)$. All these functions have the property that their biconjugates are themselves. This result will be proved in the next theorem.

Theorem 4.1.3. *If* $g \in \Gamma_0(X)$, *then* $g^{**} \equiv g$.

Proof:

We must prove that $g^{**}(x) \leq g(x)$ and $g^{**}(x) \geq g(x)$, for every $x \in X$.

- $\boxed{g^{**} \leqq g}$ Using the Young's inequality it follows

$$g^{**}(x) = \sup_{x^* \in X}\{\langle x^*, x\rangle - g^*(x^*)\} \leq \sup_{x^* \in X}\{\langle x^*, x\rangle - \langle x^*, x\rangle + g(x)\} = g(x).$$

- $\boxed{g^{**} \geqq g}$ We have 2 cases:
  - Suppose that $y \notin \operatorname{dom} g^{**}$, i.e. $g^{**}(y) = \infty$. Since $g \in \Gamma_0(X)$, because of Theorem 4.1.2 $g^* \in \Gamma_0(X)$ and g^{**} too. In particular, g^{**} is a proper function. Therefore, we have $\infty = g^{**}(y) \geq g(y)$.
  - Now, let be $y \in \operatorname{dom} g^{**}$ and suppose that $g(y) > g^{**}(y)$. We define $c := \frac{1}{2}(g(y) - g^{**}(y)) > 0$ and $C = \operatorname{epi} g$. Then $(y, g(y) - c) \notin C$ because

$$g(y) - (g(y) - c) = c > 0.$$

Therefore, arguing as we did in Theorem 4.1.2, we can say that there exists $x_1^* \in X$ such that

$$g^*(x_1^*) \leq \langle x_1^*, y\rangle - g(y) + \frac{1}{2}(g(y) - g^{**}(y)).$$

Using Young's inequality with g^{**}, we obtain

$$\langle x_1^*, y\rangle - g^{**}(y) \leq g^*(x_1^*) \leq \langle x_1^*, y\rangle - \frac{1}{2}g(y) - \frac{1}{2}g^{**}(y).$$

Therefore, $g(y) \leq g^{**}(y)$, which is a contradiction.

Therefore, $g^{**} \equiv g$.

□

Now, we are going to get assertions which relates subdifferentials with conjugate functionals.

Theorem 4.1.4. *Let be a function g and its conjugate g^* in X. Then for every $x \in X$, $x^* \in \partial g(x)$ if and only if $g(x) + g^*(x^*) = \langle x^*, x \rangle$.*

Proof:

$\boxed{\Longrightarrow}$ Let $x^* \in \partial g(x)$. Then we have that

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

This expression is equivalent to

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

Taking supreme in the equation above, we obtain

$$\langle x^*, x \rangle - g(x) \geq \sup_{x \in X} \{\langle x^*, x \rangle - g(x)\} = g^*(x^*).$$

On the other hand, because of Young's we have $\langle x^*, x \rangle - g(x) \leq g^*(x^*)$. In summary,

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

$\boxed{\Longleftarrow}$ Suppose that $g(x) + g^*(x^*) = \langle x^*, x \rangle$ for some $x,\ x^* \in X$. Then

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

Hence

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

which is equivalent to say that $x^* \in \partial g(x)$.

□

Theorem 4.1.5. *Let be $g \in \Gamma_0(X)$. Then $x^* \in \partial g(x)$ if and only if $x \in \partial g^*(x^*)$.*

We recall that

$$\partial g^*_\epsilon(x^*) = \{x \in \mathbb{R}^n : g^*(y^*) \geq g^*(x^*) + \langle y^* - x^*, x \rangle - \epsilon, \quad \forall y^* \in \mathbb{R}^n\}.$$

Proof:

$\boxed{\Longrightarrow}$ Let be $x^* \in \partial g(x)$. Using Young's inequality we obtain that

$$g^*(y^*) \geq \langle y^*, x\rangle - g(x).$$

Besides, Theorem 4.1.4 yields that $g(x) = \langle x^*, x\rangle - g^*(x^*)$. Combining both equations, we have

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

i.e $x \in \partial g^*(x^*)$.

$\boxed{\Longleftarrow}$ Let be $x \in \partial g^*(x^*)$. Arguing as we did above, we obtain that $x^* \in \partial g^{**}(x)$. Since $g \in \Gamma_0(X)$, Theorem 4.1.3 yields $g^{**} \equiv g$. Therefore, we have $x^* \in \partial g(x)$.

□

4.2 Duality in DC Programming

Let be (P) the following primal problem:

$$(\mathrm{P}) \qquad \inf_{x \in X} \{g(x) - h(x)\}.$$

Remark 4.2.1. *If the optimal solution of* (P) *is finite, then dom* $g \subset$ *dom* h.

In this section, we will give the dual problem (D) of the primal problem (P) using conjugate functions which we have seen before.

Remark 4.2.2. *If* $h \in \Gamma_0(X)$ *in* (P) *then the dual problem* (D) *is well defined and its formulation is*

$$(\mathrm{D}) \qquad \inf_{x^* \in X} \{h^*(x^*) - g^*(x^*)\}.$$

Since $h \in \Gamma_0(X)$, we have that $h^{**} \equiv h$, i.e

$$h(x) = \sup_{x^* \in X} \{\langle x^*, x\rangle - h^*(x^*)\}, \quad \forall x \in X.$$

Replacing this equality in (P)

$$\begin{aligned}
\inf_{x\in X}\{g(x)-h(x)\} &= \inf_{x\in X}\left\{g(x)-\sup_{x^*\in X}\{\langle x^*,x\rangle-h^*(x^*)\}\right\}\\
&= \inf_{x\in X}\left\{g(x)+\inf_{x^*\in X}\{h^*(x^*)-\langle x^*,x\rangle\}\right\}\\
&= \inf_{x\in X}\inf_{x^*\in X}\{g(x)+h^*(x^*)-\langle x^*,x\rangle\}\\
&= \inf_{x^*\in X}\inf_{x\in X}\{g(x)+h^*(x^*)-\langle x^*,x\rangle\}\\
&= \inf_{x^*\in X}\left\{h^*(x^*)+\inf_{x\in X}\{g(x)-\langle x^*,x\rangle\}\right\}\\
&= \inf_{x^*\in X}\left\{h^*(x^*)-\sup_{x\in X}\{\langle x^*,x\rangle-g(x)\}\right\}\\
&= \inf_{x^*\in X}\{h^*(x^*)-g^*(x^*)\}
\end{aligned}$$

Then, it makes sense the dual problem (D) of (P)

$$\text{(D)} \qquad \inf_{x^*\in X}\{h^*(x^*)-g^*(x^*)\},$$

as we wanted to see.

Remark 4.2.3. *Arguing as we did above, we can say that the dual problem of* (D) *is exactly* (P). *Besides, if we have a solution x of* (P), *then both problems have the same solution x.*

4.3 Optimality Conditions for DC Programming

In this section we are studying optimality conditions which will give us a way to know when a point is an optimal solution. We are considerating both global and local optimality conditions.

4.3.1 Global Optimality for DC Programming

Theorem 4.3.1 gives a characterization of optimal solutions in term of $\varepsilon-$subdifferentials and $\gamma-$subdifferentials (See [3]). To prove it we refer to the Definitions 2.0.13 and 2.0.14 in Chapter 2.

Notation 4.3.1. *We will denote $\mathcal{P}$ and $\mathcal{D}$ the solution sets of problems* (P) *and* (D), *respectively.*

Lemma 4.3.1. *A point $x \in \mathcal{P}$ if and only if*

$$\textit{epi } \overline{g} \subset \textit{epi } h \tag{4.1}$$

where $\overline{g}(y) = g(y) - (g(x) - h(x))$.

Proof:

For the details of the proof, we refer to [5].

□

Theorem 4.3.1. *Let $g, h : X \to \mathbb{R}$ be convex functions and let be $x \in X$. Then the following statements are equivalent:*

1) $x \in \mathcal{P}$.
2) $\partial^\gamma h(x) \subseteq \partial^\gamma g(x)$.
3) $\partial_\varepsilon h(x) \subseteq \partial_\varepsilon g(x), \forall \varepsilon > 0$.

Proof:

$\boxed{1) \Rightarrow 2)}$

By assumption we have that $g(x) - h(x) \leq g(y) - h(y), \forall y \in X$. Thus,

$$g(y) - g(x) \geq h(y) - h(x),\ \forall x \in X.$$

Let consider $\psi \in \partial^\gamma h(x)$. Then

$$h(y) \geq h(x) + \psi(y) - \psi(x),\ \forall y \in X$$

and so

$$h(y) - h(x) \geq \psi(y) - \psi(x),\ \forall y \in X.$$

Therefore, $g(y) - g(x) \geq h(y) - h(x) \geq \psi(y) - \psi(x), \forall y \in X$. Then $\psi \in \partial^\gamma g(x)$ and 2) holds.

$\boxed{2) \Rightarrow 3)}$

Let $\varepsilon \geq 0$ and $x^* \in \partial_\varepsilon h(x)$ be given. We have to show that $x^* \in \partial_\varepsilon g(x)$.

As $x^* \in \partial_\varepsilon h(x)$ we have $h(y) \geq h(x) + \langle x^*, y - x\rangle - \varepsilon, \quad \forall y \in X$.

We now choose an arbitrary $\overline{x^*} \in \partial h(x)$ and let define the function

$$\psi(y) = \sup\{\langle x^*, y - x\rangle - \varepsilon, \langle \overline{x^*}, y - x\rangle\}, \quad \forall y \in X.$$

Clearly, $\psi(x) = 0$ and $h(y) \geq h(x) + \langle \overline{x^*}, y - x\rangle$, $\forall y \in X$.

Therefore, $h(y) - h(x) \geq \psi(y)$ so $h(y) - h(x) \geq \psi(y) - \psi(x)$ and $\psi \in \partial^{\gamma} h(x)$. Then, by assumption $\psi \in \partial^{\gamma} g(x)$ and hence satisfies that

$$g(y) \geq g(x) + \psi(y) - \psi(x) = g(x) + \psi(y) \geq g(x) + \langle x^*, y - x\rangle - \varepsilon.$$

Hence, $x^* \in \partial_{\varepsilon} g(x)$.

$\boxed{3) \Rightarrow 1)}$

We prove this implication by contradiction.

Suppose that $x \notin \mathcal{P}$. By Lemma 4.3.1 this is equivalent to say that $\exists\, x_0 \in X$ such that $(x_0, g(x_0)) \notin$ epi h.

Since epi h is a closed convex set, by Theorem 4.1.1 there exists a nonvertical hyperplane separating the point $(x_0, g(x_0))$ from epi h, that is to say, $\exists\, x^* \in X$, $c \in \mathbb{R}$ such that:

$$h(y) \geq \langle x^*, y - x\rangle - c, \quad \forall y \in X$$

Hence,

$$(x_0, \overline{g}(x_0)) \notin epi\ h \Rightarrow \left\{ \begin{array}{c} x_0 \notin \mathbb{R}^n \quad \Rightarrow \quad h(x_0) < \langle x^*, x_0 - x\rangle - c \\ h(x_0) > \overline{g}(x_0) \end{array} \right\} \Rightarrow$$

$$\Rightarrow \overline{g}(x_0) = g(x_0) - (g(x) - h(x)) < h(x_0) < \langle x^*, x_0 - x\rangle - c.$$

Let $\varepsilon = c + h(x)$. One has $\varepsilon \geq 0$ because $x \in X$ by assumption and $h(x) \geq \langle x^*, x - x\rangle - c = -c \Rightarrow h(x) + c \geq 0$.

Then, we have:

$$h(y) \geq \langle x^*, y - x\rangle - c \Rightarrow h(y) \geq h(x) + \langle x^*, y - x\rangle - \varepsilon,\ \forall x \in X$$

and

$g(x_0) - g(x) + h(x) < \langle x^*, x_0 - x\rangle - c$, in other words that means,

$$g(x_0) < g(x) - h(x) + \langle x^*, x_0 - x\rangle - c.$$

Hence, $g(x_0) < g(x) + \langle x^*, x_0 - x\rangle - \varepsilon$ which implies that 3) does not hold, which is a contradiction.

□

Remark 4.3.1. *By the symmetry of the DC duality, Theorem 4.3.1 has its dual counterpart, i.e., the following statements are equivalent:*

1) $x^* \in \mathcal{D}$.

2) $\partial^\gamma g^*(x^*) \subseteq \partial^\gamma h^*(x^*)$.

3) $\partial_\varepsilon g^*(x^*) \subseteq \partial_\varepsilon h^*(x^*), \forall \varepsilon > 0$.

Corollary 4.3.1. *Let $g, h : X \to \mathbb{R}$ be convex functions. Then*

$$x \in \mathcal{P} \text{ iff } \partial h(x) \subseteq \partial g(x),$$
$$x^* \in \mathcal{D} \text{ iff } \partial g^*(x^*) \subseteq \partial h^*(x^*).$$

Proof:

Trivially, taking limits in the $\varepsilon-$subdiferential as $\varepsilon \to 0$.

□

Finally, we will see that solving the primal problem (P) implies solving the dual problem (D) and vice versa. It may be useful if one of them is easier to solve than the other.

Proposition 4.3.1. *Let $g, h : X \to \mathbb{R}$ be convex functions. Then*

- $\bigcup_{x \in \mathcal{P}} \partial h(x) \subset \mathcal{D}$.
- $\bigcup_{x^* \in \mathcal{D}} \partial g^*(x^*) \subset \mathcal{P}$.

Proof:

Let $x \in \mathcal{P}$, and let $x^* \in \partial h(x)$. Then, by Theorem 4.3.1, $x^* \in \partial g(x)$. From Theorem 4.1.4, we have

$$h^*(x^*) = \langle x^*, x\rangle - h(x), \quad g^*(x^*) = \langle x^*, x\rangle - g(x).$$

Therefore,

$$h^*(x^*) - g^*(x^*) = h(x) - g(x),$$

by Remark 4.2.3, we conclude that $x^* \in \mathcal{D}$.

Analogously, let $x^* \in \mathcal{D}$, and let $x \in \partial g^*(x^*)$. By Remark 4.3.1, $x \in \partial h^*(x^*)$. From Theorem 4.1.4, we have

$$h^*(x^*) = \langle x^*, x\rangle - h(x), \quad g^*(x^*) = \langle x^*, x\rangle - g(x).$$

Hence,

$$h^*(x^*) - g^*(x^*) = h(x) - g(x),$$

by Remark 4.2.3, we conclude that $x \in \mathcal{P}$.

The global optimality condition is dificult to use for deriving solution methods to problem (P). However, we have local optimality conditions which algorithm DCA is based on.

4.3.2 Local Optimality for DC Programming

In this subsection, we are studying conditions that allow us to ensure when we have obtained a local solution of our problems (P) and (D). These results on DC programming on which DCA relies the reader is referred to ...

Notation 4.3.2. *We will denote $\mathcal{P}_l$ and $\mathcal{D}_l$ the following sets:*

$$\mathcal{P}_l = \{x \in X : \partial h(x) \subset \partial g(x)\} \quad \mathcal{D}_l = \{x^* \in X : \partial g^*(x^*) \subset \partial h^*(x^*)\}.$$

Definition 4.3.1. *Let be $g, h \in \Gamma_0(X)$. A point x is said to be a **local minimizer** of $g-h$ if $g(x) - h(x)$ is finite and there exists a neighbourhood U of x such that*

$$g(x) - h(x) \leq g(y) - h(y), \quad \forall y \in U.$$

Definition 4.3.2. *Let be $g, h \in \Gamma_0(X)$. A point x is said to be a **critical point** of $g-h$ if $\partial g(x) \cap \partial h(x) \neq \emptyset$.*

Theorem 4.3.2. *Let be $g, h \in \Gamma_0(X)$. Then,*

1) *If x is a local minimizer of $g-h$, then $x \in \mathcal{P}_l$.*

2) *Let x be a critical point of $g-h$ and $x^* \in \partial g(x) \cap \partial h(x)$. Let U be a neighbourhood of x such that $U \cap dom\ g \subset dom\ \partial h$. If for any $y \in U \cap dom\ g$ there exists $y^* \in \partial h(y)$ such that $h^*(y^*) - g^*(y^*) \geq h^*(x^*) - g^*(x^*)$, then x is a local minimizer of $g-h$.*

Proof:

1) If x is a local minimizer of $g - h$, then there exists a neighbourhood U of x such that
$$g(y) - g(x) \geq h(y) - h(x), \quad \forall x \in U \cap \text{dom } g.$$
Besides, for $x^* \in \partial h(x)$ we have
$$h(y) \geq h(x) + \langle x^*, y - x \rangle .$$
Hence, we obtain
$$g(y) - g(x) \geq \langle x^*, y - x \rangle , \quad \forall x \in U \cap \text{dom } g.$$
Then, $x^* \in \partial g(x)$ and we conclude that $x \in \mathcal{P}_l$.

2) Since $x^* \in \partial g(x) \cap \partial h(x)$, then $g(x) + g^*(x^*) = \langle x^*, x \rangle = h(x) + h^*(x^*)$. Hence,
$$g(x) - h(x) = h^*(x^*) - g^*(x^*).$$
By assumption, for each $y \in U \cap \text{dom } g$ there is $y^* \in \partial h(y)$ such that
$$h^*(y^*) - g^*(y^*) \geq h^*(x^*) - g^*(x^*).$$
On the other hand, since $y^* \in \partial h(y)$ we have $h(y) + h^*(y^*) = \langle y^*, y \rangle \leq g(y) + g^*(y^*)$. Then, we have
$$g(y) - h(y) \geq h^*(y^*) - g^*(y^*).$$
In summary, we obtain
$$g(y) - h(y) \geq h^*(y^*) - g^*(y^*) \geq h^*(x^*) - g^*(x^*) = g(x) - h(x),$$
for every $y \in U \cap \text{dom } g$, namely x is a local minimizer.

Corollary 4.3.2. *Let x be a point that admits a neighbourhood U such that $\partial g(x) \cap \partial h(y) \neq \emptyset$ for every $y \in U \cap$ dom g. Then x is a local minimizer of $g - h$. More precisely, $g(y) - h(y) \geq g(x) - h(x)$, for every $y \in U \cap$ dom g.*

Proof:

Let $y \in U \cap \text{dom } g$ and let $y^* \in \partial g(x) \cap \partial h(y)$. Since $y^* \in \partial g(x)$ we have $g(x) + g^*(y^*) = \langle y^*, x \rangle \leq h(x) + h^*(y^*)$, then $h^*(y^*) - g^*(y^*) \geq g(x) - h(x)$. Moreover, if $x^* \in \partial g(x) \cap \partial h(x)$, then $g(x) + g^*(x^*) = \langle x^*, x \rangle = h(x) + h^*(x^*)$. Hence, $g(x) - h(x) = h^*(x^*) - g^*(x^*)$. Therefore,
$$h^*(y^*) - g^*(y^*) \geq g(x) - h(x) = h^*(x^*) - g^*(x^*).$$
By Theorem 4.3.2, we conclude that x^* is a local minimizer.

□

Remark 4.3.2. *Again, Theorem 4.3.2 has its dual counterpart. Thus,*

1) *If x^* is a local minimizer of $h^* - g^*$, then $x^* \in \mathcal{D}_l$.*

2) *Let x^* be a critical point of $h^* - g^*$ and $x \in \partial g^*(x^*) \cap \partial h^*(x^*)$. Let V be a neighbourhood of x^* such that $V \cap$ dom $\partial g^* \subset$ dom h. If for any $y^* \in V \cap$ dom ∂g^* there is $y \in \partial g^*(y^*)$ such that $g(y) - h(y) \geq g(x) - h(x)$, then x^* is a local minimizer of $h^* - g^*$.*

It might happen that (D) is easier to locally solve than (P). So it is useful to state results relative to the d.c. duality transportation of local minimizers.

Corollary 4.3.3. *Let $x \in$ dom ∂h be a local minimizer of $g - h$ and let $x^* \in \partial h(x)$ (i.e. x admits a neighbourhood U such that $g(y) - h(y) \geq g(x) - h(x)$, $\forall y \in U \cap$ dom g). If*

$$x^* \in \textit{int}(\textit{dom } g^*) \textit{ and } \partial g^*(x^*) \subset U,$$

then x^* is a local minimizer of $h^* - g^*$.

Proof:

According to 1) of Theorem 4.3.2, $x \in \mathcal{P}_l$. Thus, since $x^* \in \partial h(x)$ we have that $x^* \in \partial g(x)$. Hence, $x^* \in \partial g(x) \cap \partial h(x)$. By Theorem 4.1.5, $x \in \partial g^*(x^*) \cap \partial h^*(x^*)$. Under the assumption that $x^* \in$ dom g^* and $\partial g^*(x^*) \subset U$, and upper semicontinuity of ∂g^*, x^* admits a neighbourhood $V \subset$ int(dom g^*) such that $\partial g^*(V) \subset U \cap$ dom g, since we have range $\partial g^* =$ dom $\partial g \subset$ dom g. Thus, we have

- x^* is a critical point of $h^* - g^*$.
- $x \in \partial g^*(x^*) \cap \partial h^*(x^*)$.
- V is a neighbourhood of x^* such that $V \cap$ dom $\partial g^* \subset U \cap$ dom $g \subset$ dom h.
- For each $y^* \in V \cap$ dom ∂g^* there is $y \in \partial g^*(y^*) \subset U \cap$ dom g such that $g(y) - h(y) \geq g(x) - h(x)$.

By Remark 4.3.2, we conclude that x^* is a local minimizer of $h^* - g^*$.

4.4 The DCA Algorithm

In this section we will expose an algorithm that solves (P) or (D) using the framework studied. This algorithm is called *DCA Algorithm* and we will study its convergence and applications in different problems. Finally, we will use Python to program it.

4.4.1 Description of DCA for general DC programs

For each fixed $x \in X$, we consider the problem

$$(S(x)) \qquad \inf_{x^* \partial h(x)} \{h^*(x^*) - g^*(x^*)\},$$

which is equivalent to

$$\inf_{x^* \partial h(x)} \{\langle x^*, x\rangle - g^*(x^*)\}.$$

Analogously, for each fixed $x^* \in X^*$, for duality, we define the problem

$$(T(x^*)) \qquad \inf_{x \in \partial g^*(x^*)} \{g(x) - h(x)\}.$$

This problem is equivalent to

$$\inf_{x \in \partial g^*(x^*)} \{\langle x^*, x\rangle - h(x)\}.$$

Let $\mathcal{S}(x)$, $\mathcal{T}(x^*)$ the solution sets of Problems $(S(x))$ and $(T(x^*))$, respectively. The complete form of DCA is based upon duality of DC optimization defined by (P) and (D). It allows approximating a point $(x, x^*) \in \mathcal{P}_l \times \mathcal{D}_l$. From a point of $x_0 \in$ dom ∂g, the algorithm consists on constructing two sequences $\{x_k\}$ and $\{x_k^*\}$ defined by

$$x_k^* \in \mathcal{S}(x_k); \quad x_{k+1} \in \mathcal{T}(x_k^*).$$

From a practical point of view, although $(S(x_k))$ and $(T(x_k^*))$ are easier to solve than (P) and (D), it is often used the simplified form of DCA.

Simplified form of DCA

The idea of the simplified DCA is to construct two sequences $\{x_k\}$ and $\{x_k^*\}$ easy to compute and satisfying the following statements:

- The sequences $(g - h)(x_k)$ and $(h^* - g^*)(x_k^*)$ are decreasing.
- If

$$x = \lim_{k \to +\infty} x_k \quad \text{and} \quad x^* = \lim_{k \to +\infty} x_k^*,$$

  then x and x^* either are critical points of $g - h$ and $h^* - g^*$, respectively, or $(x, x^*) \in \mathcal{P}_l \times \mathcal{D}_l$.

Then, starting from a point $x_0 \in$ dom g, we shall construct the sequences by setting

$$x_k^* \in \partial h(x_k), \qquad x_{k+1} \in \partial g^*(x_k^*).$$

Thus, we can give the following interpretation to the DCA. At each iteration k we have:

$$(\mathrm{P}_k) \quad x_k^* \in \partial h(x_k) \to x_{k+1} \in \partial g^*(x_k^*) = \operatorname*{argmin}_{x\in X}\{g(x) - [h(x_k) + \langle x_k^*, x - x_k\rangle]\}.$$

$$(\mathrm{D}_k) \quad x_k \in \partial g^*(x_{k-1}^*) \to x_k^* \in \partial h(x_k) = \operatorname*{argmin}_{x^*\in Y}\{h^*(x^*) - \big[g^*(x_{k-1}^*) + \big\langle x^* - x_{k-1}^*, x_k\big\rangle\big]\}.$$

Using the Theorem 4.1.4, we can obtain the previous results:
For every $x \in X$, we have

$$x_k^* \in \partial h(x) \Longleftrightarrow h(x) + h^*(x_k^*) = \langle x_k^*, x\rangle \,,$$

in particular, for $x = x_k$:

$$x_k^* \in \partial h(x_k) \Longleftrightarrow h(x_k) + h^*(x_k^*) = \langle x_k^*, x_k\rangle \,.$$

Subtracting both equations:

$$h(x) = h(x_k) + \langle x_k^*, x - x_k\rangle \,.$$

Hence

$$\operatorname*{argmin}_{x\in X}\{g(x) - h(x)\} = \operatorname*{argmin}_{x\in X}\{g(x) - [h(x_k) + \langle x_k^*, x - x_k\rangle]\}.$$

We can argue in an equivalent way to see that

$$\operatorname*{argmin}_{x^*\in Y}\{h^*(x^*) - g^*(x^*)\} = \operatorname*{argmin}_{x^*\in Y}\{h^*(x^*) - \big[g^*(x_{k-1}^*) + \big\langle x^* - x_{k-1}^*, x_k\big\rangle\big]\}.$$

Now, we are proving that this algorithm is well-defined from the construction of the sequences.

Theorem 4.4.1. *Let be $x_0 \in$ dom ∂g. If*

$$\textit{dom}\ \partial g \subset \textit{dom}\ \partial h \quad \textit{and} \quad \textit{dom}\ \partial h^* \subset \textit{dom}\ \partial g^*$$

then the sequences $\{x_k\}$, $\{y_k\}$ are well defined.

<u>Proof:</u>

We have $x_{k+1} \in \partial g^*(x_k^*)$ and $x_k^* \in \partial h(x_k)$, for every $k \geq 0$. With these assumptions, we get that $\{x_k\} \subset$ range $\partial g^* =$ dom $\partial g \subset$ dom ∂h. Then, we can obtain x_k^* from $\partial h(x_k)$. Analogously, $\{x_k^*\} \subset$ range $\partial h =$ dom $\partial h^* \subset$ dom ∂g^*, therefore, it makes sense to have $x_{k+1} \in \partial g^*(x_k^*)$.

□

4.4.2 Convergence of DCA for general DC programs

Now, we are going to see that this algorithm is convergent. We are also going to study interesting properties that makes this algorithm good. Firstly, we need to introduce some notation to simplify the following statements.

Let $0 \leq \rho_g < \rho(g)$ and $0 \leq \rho_h < \rho(h)$ be real nonnegative numbers (resp. ρ_g^*, ρ_h^*). Also let $\Delta x_k = x_{k+1} - x_k$ and $\Delta x_k^* = x_{k+1}^* - x_k^*$.

Proposition 4.4.1. *Let be $\{x_k\}$ and $\{x_k^*\}$ the sequences generated by the simplified DCA. Then we have*

1. $(g-h)(x_{k+1}) \leq (h^*-g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k\|^2 \leq (g-h)(x_k) - \frac{\rho_g+\rho_h}{2}\|\Delta x_k\|^2.$
   The equality $(g-h)(x_{k+1}) = (g-h)(x_k)$ holds if and only if
   $$x_k \in \partial g^*(x_k^*), \quad x_k^* \in \partial h(x_{k+1}) \quad \textit{and} \quad (\rho_g+\rho_h)\|\Delta x_k\| = 0.$$

2. *By duality, we have*
   $$(h^*-g^*)(x_{k+1}^*) \leq (g-h)(x_{k+1}) - \frac{\rho_g^*}{2}\|\Delta x_k^*\|^2 \leq (h^*-g^*)(x_k^*) - \frac{\rho_g^*+\rho_h^*}{2}\|\Delta x_k^*\|^2.$$
   The equality $(h^*-g^*)(x_{k+1}^*) = (h^*-g^*)(x_k^*)$ holds if and only if
   $$x_{k+1} \in \partial g^*(x_{k+1}^*), \quad x_k^* \in \partial h(x_{k+1}) \quad \textit{and} \quad (\rho_g^*+\rho_h^*)\|\Delta x_k^*\| = 0.$$

Proof:

We are going to prove Property 1. The dual case is analogously proved.
Since $x_k^* \in \partial h(x_k)$ we have that

$$h(x_{k+1}) \geq h(x_k) + \langle x_k^*, x_{k+1} - x_k \rangle + \frac{\rho_h}{2}\|\Delta x_k\|^2.$$

Hence,

$$(g-h)(x_{k+1}) \leq g(x_{k+1}) - \langle x_k^*, x_{k+1} - x_k \rangle - h(x_k) - \frac{\rho_h}{2}\|\Delta x_k\|^2. \tag{4.2}$$

In the same way, $x_{k+1} \in \partial g^*(x_k^*) \Longleftrightarrow x_k^* \in \partial g(x_{k+1})$ follows that

$$g(x_k) \geq g(x_{k+1}) + \langle x_k^*, x_k - x_{k+1} \rangle + \frac{\rho_g}{2}\|\Delta x_k\|^2.$$

Thus,

$$g(x_{k+1}) - \langle x_k^*, x_{k+1} - x_k \rangle - h(x_k) \leq (g-h)(x_k) - \frac{\rho_g}{2}\|\Delta x_k\|^2. \tag{4.3}$$

On the other hand, by the Theorem 4.1.4 we have:

$$x_{k+1} \in \partial g^*(x_k^*) \iff g(x_{k+1}) + g^*(x_k^*) = \langle x_k^*, x_{k+1} \rangle \,,$$
$$x_k^* \in \partial h(x_k) \iff h(x_k) + h^*(x_k^*) = \langle x_k^*, x_k \rangle \,.$$

Substracting both expression, we obtain:

$$g(x_{k+1}) - \langle x_k^*, x_{k+1} - x_k \rangle - h(x_k) = h^*(x_k^*) - g^*(x_k^*). \tag{4.4}$$

Finally, combining (4.2), (4.3) and (4.4) we have that

$$(g-h)(x_{k+1}) \le (h^* - g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k\|^2 \le (g-h)(x_k) - \frac{\rho_g + \rho_h}{2}\|\Delta x_k\|^2.$$

Now, we are going to prove when the equality holds:

- $\boxed{\Longrightarrow}$ If $(g-h)(x_{k+1}) = (g-h)(x_k)$, the last equation yields that

$$\frac{\rho_g + \rho_h}{2}\|\Delta x_k\|^2 \le 0 \iff (\rho_g + \rho_h)\|\Delta x_k\| = 0,$$

  since ρ_g and ρ_h are non-negative. We have two choices:

  – $\boxed{\rho_g + \rho_h > 0}$ This implies that $\|\Delta x_k\| = 0$ if and only if $x_{k+1} = x_k$. Thus

$$x_{k+1} \in \partial g^*(x_k^*) \Longrightarrow x_k \in \partial g^*(x_k^*),$$
$$x_k^* \in \partial h(x_k) \Longrightarrow x_k^* \in \partial h(x_{k+1}).$$

  – $\boxed{\rho_g = \rho_h = 0}$ This yields

$$(g-h)(x_{k+1}) \le (h^* - g^*)(x_k^*) \le (g-h)(x_k),$$

  which, from assumption, gives that

$$(h^* - g^*)(x_k^*) = (g-h)(x_k)$$
$$(g-h)(x_{k+1}) = (h^* - g^*)(x_k^*)$$

  Then

$$\begin{cases} x_k^* \in \partial h(x_k) \Leftrightarrow & \langle x_k^*, x_k \rangle = h(x_k) + h^*(x_k^*), \\ & h(x_k) = g(x_k) + g^*(x_k^*) - h^*(x_k^*), \end{cases}$$

  and

$$\begin{cases} x_{k+1} \in \partial g^*(x_k^*) \Leftrightarrow & \langle x_k^*, x_{k+1} \rangle = g(x_{k+1}) + g^*(x_k^*), \\ & g(x_{k+1}) = h(x_{k+1}) + h^*(x_k^*) - g^*(x_k^*), \end{cases}$$

  produces that

$$\langle x_k^*, x_k \rangle = g(x_k) + g^*(x_k^*) \iff x_k \in \partial g^*(x_k^*),$$
$$\langle x_k^*, x_{k+1} \rangle = h(x_{k+1}) + h^*(x_k^*) \iff x_k^* \in \partial h(x_{k+1}).$$

- $\boxed{\Longleftarrow}$ Actually, we have just prove the reciprocal, because

$$\left\{\begin{array}{lcrcl} x_{k+1} \in \partial g^*(x_k^*) & \Leftrightarrow & \langle x_k^*, x_{k+1}\rangle & = & g(x_{k+1}) + g^*(x_k^*), \\ x_k^* \in \partial h(x_{k+1}) & \Leftrightarrow & \langle x_k^*, x_{k+1}\rangle & = & h(x_{k+1}) + h^*(x_k^*), \end{array}\right.$$

and

$$\left\{\begin{array}{lcrcl} x_k^* \in \partial h(x_k) & \Leftrightarrow & \langle x_k^*, x_k\rangle & = & h(x_k) + h^*(x_k^*), \\ x_k \in \partial g^*(x_k^*) & \Leftrightarrow & \langle x_k^*, x_k\rangle & = & g(x_k) + g^*(x_k^*), \end{array}\right.$$

yields, respectively

$$\begin{array}{l} (g-h)(x_{k+1}) = (h^* - g^*)(x_k^*), \\ (h^* - g^*)(x_k^*) = (g-h)(x_k). \end{array}$$

which says that the equality holds.

□

Corollary 4.4.1. *Let be $\{x_k\}$ and $\{x_k^*\}$ the sequences generated by DCA. Then we have*

1. *(a)* $$\begin{aligned}(g-h)(x_{k+1}) &\leq (h^*-g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k\|^2 \\ &\leq (g-h)(x_k) - \left[\frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2 + \frac{\rho_h}{2}\|\Delta x_k\|^2\right].\end{aligned}$$

   (b) $$\begin{aligned}(g-h)(x_{k+1}) &\leq (h^*-g^*)(x_k^*) - \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2 \\ &\leq (g-h)(x_k) - \left[\frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2 + \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2\right].\end{aligned}$$

   The equality $(g-h)(x_{k+1}) = (g-h)(x_k)$ holds if and only if

   $$x_k \in \partial g^*(x_k^*),\ x_k^* \in \partial h(x_{k+1}) \textit{ and } (\rho_g + \rho_h)\Delta x_k = \rho_g^*\Delta x_{k-1}^* = \rho_h^*\Delta x_k^* = 0.$$

2. *By duality, we have*

   (c) $$\begin{aligned}(h^*-g^*)(x_{k+1}^*) &\leq (g-h)(x_{k+1}) - \frac{\rho_g^*}{2}\|\Delta x_k^*\|^2 \\ &\leq (h^*-g^*)(x_k^*) - \left[\frac{\rho_g^*}{2}\|\Delta x_k^*\|^2 + \frac{\rho_h}{2}\|\Delta x_k\|^2\right]\end{aligned}$$

   (d) $$\begin{aligned}(h^*-g^*)(x_{k+1}^*) &\leq (g-h)(x_{k+1}) - \frac{\rho_g^*}{2}\|\Delta x_{k+1}\|^2 \\ &\leq (h^*-g^*)(x_k^*) - \left[\frac{\rho_g}{2}\|\Delta x_{k+1}\|^2 + \frac{\rho_h}{2}\|\Delta x_k\|^2\right]\end{aligned}$$

The equality $(h^* - g^*)(x_{k+1}^*) = (h^* - g^*)(x_k^*)$ *holds if and only if*

$x_{k+1} \in \partial g^*(x_{k+1}^*), x_k^* \in \partial h(x_{k+1})$ *and* $(\rho_g^* + \rho_h^*)\Delta x_k^* = \rho_h \Delta x_k = \rho_g \Delta x_{k+1} = 0.$

Proof:

It is an immediate consequence of the Proposition 4.4.1. We only prove (a) and (b) combining previous expressions. From

$$(g-h)(x_{k+1}) \leq (h^* - g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k\|^2$$

and

$$(h^* - g^*)(x_k^*) \leq (g-h)(x_k) - \frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2$$

we obtain that

$$\begin{aligned}(g-h)(x_{k+1}) &\leq (h^* - g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k\|^2 \\ &\leq (g-h)(x_k) - \left[\frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2 + \frac{\rho_h}{2}\|\Delta x_k\|^2\right].\end{aligned}$$

Similarly, equations

$$(g-h)(x_{k+1}) \leq (h^* - g^*)(x_k^*) - \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2$$

and

$$(h^* - g^*)(x_k^*) \leq (g-h)(x_k) - \frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2$$

yields

$$\begin{aligned}(g-h)(x_{k+1}) &\leq (h^* - g^*)(x_k^*) - \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2 \\ &\leq (g-h)(x_k) - \left[\frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2 + \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2\right].\end{aligned}$$

We can argue in the same way as we did in the Proposition 4.4.1 to see when equality holds.

□

The basic convergence theorem of DCA for general d.c. programming will be stated below.

Theorem 4.4.2. *Suppose that the sequences* $\{x_k\}$ *and* $\{x_k^*\}$ *are defined by DCA. Then we have*

1. $$(g-h)(x_{k+1}) \le (h^*-g^*)(x_k^*) - \max\left\{\frac{\rho_h}{2}\|\Delta x_k\|^2, \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2\right\}$$
$$\le (g-h)(x_k) - \max\left\{\frac{\rho_g+\rho_h}{2}\|\Delta x_k\|^2, \frac{\rho_g^*}{2}\|\Delta x_{k-1}\|^2 + \frac{\rho_h}{2}\|\Delta x_k\|^2, \frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2 + \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2\right\}.$$
The equality $(g-h)(x_{k+1}) = (g-h)(x_k)$ holds if and only if $x_k \in \partial g^*(x_k^*)$, $x_k^* \in \partial h(x_{k+1})$ and $(\rho_g+\rho_h)\Delta x_k = \rho_g^*\Delta x_{k-1}^* = \rho_h^*\Delta x_k^* = 0$. In this case
   - *$(g-h)(x_{k+1}) = (h^*-g^*)(x_k^*)$ and x_k, x_{k+1} are the critical points of $g-h$ satisfying $x_k^* \in \partial g(x_k) \cap \partial h(x_k)$ and $x_k^* \in \partial g(x_{k+1}) \cap \partial h(x_{k+1})$.*
   - *x_k^* is a critical point of h^*-g^* satisfying $[x_k, x_{k+1}] \subset (\partial g^*(x_k^*) \cap \partial h^*(x_k^*))$.*
   - *$x_{k+1} = x_k$ if $\rho_g+\rho_h > 0$, $x_k^* = x_{k-1}^*$ if $\rho_g^* > 0$ and $x_k^* = x_{k+1}^*$ if $\rho_h^* > 0$.*
2. *Similarly, for the dual problem we have*
$$(h^*-g^*)(x_{k+1}^*) \le (g-h)(x_{k+1}) - \max\left\{\frac{\rho_g}{2}\|\Delta x_{k+1}\|^2, \frac{\rho_g^*}{2}\|\Delta x_k^*\|^2\right\}$$
$$\le (h^*-g^*)(x_k^*) - \max\left\{\frac{\rho_g^*+\rho_h^*}{2}\|\Delta x_k^*\|^2, \frac{\rho_g}{2}\|\Delta x_{k+1}\|^2 + \frac{\rho_h}{2}\|\Delta x_k\|^2, \frac{\rho_g^*}{2}\|\Delta x_k^*\|^2 + \frac{\rho_h}{2}\|\Delta x_k\|^2\right\}.$$
The equality $(h^*-g^*)(x_{k+1}^*) = (h^*-g^*)(x_k^*)$ holds if and only if $x_{k+1} \in \partial g^*(x_{k+1}^*)$, $x_k^* \in \partial h(x_{k+1})$ and $(\rho_g^*+\rho_h^*)\Delta x_k^* = \rho_h\Delta x_k = \rho_g\Delta x_{k+1} = 0$. In this case
   - *$(h^*-g^*)(x_{k+1}^*) = (g-h)(x_{k+1})$ and x_k^*, x_{k+1}^* are the critical points of h^*-g^* satisfying $x_{k+1} \in \partial g^*(x_k^*) \cap \partial h^*(x_k^*)$ and $x_{k+1} \in \partial g^*(x_{k+1}^*) \cap \partial h^*(x_{k+1}^*)$.*
   - *x_{k+1} is a critical point of $g-h$ satisfying $[x_k^*, x_{k+1}^*] \subset (\partial g(x_{k+1}) \cap \partial h(x_{k+1}))$.*
   - *$x_{k+1}^* = x_k^*$ if $\rho_g^*+\rho_h^* > 0$, $x_{k+1} = x_{k+2}$ if $\rho_g > 0$ and $x_k = x_{k+1}$ if $\rho_h > 0$.*
3. *If α is finite then the decreasing sequences $\{(g-h)(x_k)\}$ and $\{(h^*-g^*)(x_k^*)\}$ converge to the same limit $\beta \ge \alpha$. If $\rho_g+\rho_h > 0$ (resp. $\rho_g^*+\rho_h^* > 0$), then*
$$\lim_{k\to+\infty}(x_{k+1}-x_k) = 0 \quad (\textit{resp.} \lim_{k\to+\infty}(x_{k+1}^*-x_k^*) = 0).$$
Moreover,
$$\lim_{k\to+\infty}\{g(x_k)+g^*(x_k^*)-\langle x_k^*, x_k\rangle\} = 0 = \lim_{k\to+\infty}\{h(x_{k+1})+h^*(x_k^*)-\langle x_k^*, x_{k+1}\rangle\}.$$

4. *If α is finite and the sequences $\{x_k\}$ and $\{x_k^*\}$ are bounded, then for every limit x of $\{x_k\}$ (resp. x^* of $\{x_k^*\}$) there exists a cluster point x of $\{x_k\}$ (resp. x^* of $\{x_k^*\}$) such that*

   - $(x, x^*) \in (\partial g^*(x^*) \cap \partial h^*(x^*)) \times (\partial g(x) \cap \partial h(x))$.
   - $(g - h)(x) = \beta = (h^* - g^*)(x^*)$.
   - $\lim\limits_{k\to+\infty} \{g(x_k) + g^*(x_k^*)\} = \lim\limits_{k\to+\infty} \langle x_k^*, x_k \rangle$.

Proof:

Properties 1 and 2 are proved analogously, therefore we give the proof for 1 only.

1. From 1 of Corollary 4.4.1 we have that:

$$\begin{cases} (g-h)(x_{k+1}) \le (h^* - g^*)(x_k^*) - \dfrac{\rho_h}{2}\|\Delta x_k\|^2, \\ (g-h)(x_{k+1}) \le (h^* - g^*)(x_k^*) - \dfrac{\rho_h^*}{2}\|\Delta x_k^*\|^2. \end{cases}$$

   Hence,

$$\begin{aligned} (g-h)(x_{k+1}) &\le (h^* - g^*)(x_k^*) + \min\left\{-\frac{\rho_h}{2}\|\Delta x_k\|^2, -\frac{\rho_h^*}{2}\|\Delta x_k^*\|^2\right\} = \\ &= (h^* - g^*)(x_k^*) - \max\left\{\frac{\rho_h}{2}\|\Delta x_k\|^2, \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2\right\}. \end{aligned}$$

   We have two choices:

   - If $\rho_h\|\Delta x_k\|^2 \le \rho_h^*\|\Delta x_k^*\|^2$, from paragraph 1 of the Proposition 4.4.1:

$$\begin{aligned} (h^* - g^*)(x_k^*) - \frac{\rho_h^*}{2} &\le (h^* - g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k\|^2 \\ &\le (g-h)(x_k) - \frac{\rho_g + \rho_h}{2}\|\Delta x_k\|^2 \end{aligned}$$

     Similarly, from 1 of Corollary 4.4.1 we obtain that

$$\begin{aligned} (h^* - g^*)(x_k^*) - \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2 &\le (h^* - g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k\|^2 \\ &\le (g-h)(x_k) - \left[\frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2 + \frac{\rho_h}{2}\|\Delta x_k\|^2\right]. \end{aligned}$$

     Moreover, paragraph 2 of the same corollary yields that

$$(h^* - g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k^*\|^2 \le (g-h)(x_k) - \left[\frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2 + \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2\right].$$

Combining these inequalities, we get

$$\begin{aligned}(g-h)(x_{k+1}) &\leq (h^*-g^*)(x_k^*) - \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2 \\ &\leq (g-h)(x_k) - \max\left\{\frac{\rho_g+\rho_h}{2}\|\Delta x_k\|^2, \frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2\right. \\ &+ \left.\frac{\rho_h}{2}\|\Delta x_k\|^2, \frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2 + \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2\right\}.\end{aligned}$$

- If $\rho_h^*\|\Delta x_k^*\|^2 \leq \rho_h\|\Delta x_k\|^2$, from paragraph 1 of the Corollary 4.4.1:

$$(h^*-g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k\|^2 \leq (g-h)(x_k) - \frac{\rho_g+\rho_h}{2}\|\Delta x_k\|^2.$$

Similarly, using again this corollary, we obtain that

$$(h^*-g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k\|^2 \leq (g-h)(x_k) - \left[\frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2 + \frac{\rho_h}{2}\|\Delta x_k\|^2\right].$$

Finally,

$$\begin{aligned}(h^*-g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k\|^2 &\leq (h^*-g^*)(x_k^*) - \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2 \\ &\leq (g-h)(x_k) - \left[\frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2 + \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2\right].\end{aligned}$$

Again, combining these equations yields that

$$\begin{aligned}(g-h)(x_{k+1}) &\leq (h^*-g^*)(x_k^*) - \frac{\rho_h}{2}\|\Delta x_k\|^2 \\ &\leq (g-h)(x_k) - \max\left\{\frac{\rho_g+\rho_h}{2}\|\Delta x_k\|^2, \frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2\right. \\ &+ \left.\frac{\rho_h}{2}\|\Delta x_k\|^2, \frac{\rho_g^*}{2}\|\Delta x_{k-1}^*\|^2 + \frac{\rho_h^*}{2}\|\Delta x_k^*\|^2\right\}.\end{aligned}$$

Therefore, we have just obtained the inequality 1. We don't prove when equality holds because it is done in the Proposition 4.4.1. However, we are going to see the consequences:

- We have already proved that $(g-h)(x_{k+1}) = (h^*-g^*)(x_k^*)$ in the Proposition 4.4.1. On the other hand, we have that $x_k^* \in \partial h(x_k)$ by construction and $x_k^* \in \partial g(x_k)$ because $x_k \in \partial g^*(x_k^*)$. Besides, $x_k^* \in \partial g(x_{k+1})$ because $x_{k+1} \in \partial g^*(x_k^*)$ by construction and $x_k^* \in \partial h(x_{k+1})$. Therefore,

$$\begin{cases} x_k^* \in \partial g(x_k) \cap \partial h(x_k), \\ x_k^* \in \partial g(x_{k+1}) \cap \partial h(x_{k+1}), \end{cases}$$

i.e., x_k and x_{k+1} are critical points of $g-h$.

- From the previous paragraph, we have that $x_k^* \in \partial g(x_k) \cap \partial h(x_k)$, which is equivalent to $x_k \in \partial g^*(x_k^*) \cap \partial h^*(x_k^*)$, i.e., x_k^* is a critical point of $h^* - g^*$. Similarly, $x_{k+1} \in \partial g^*(x_k^*) \cap \partial h^*(x_k^*)$ since $x_k^* \in \partial g(x_{k+1}) \cap \partial h(x_{k+1})$. But $\partial g^*(x_k^*) \cap \partial h^*(x_k^*)$ is convex so that $[x_k, x_{k+1}] \subset (\partial g^*(x_k^*) \cap \partial h^*(x_k^*))$.
- This part is straightforward, and thus its proof is omitted.

3. From paragraph 1, we have that

$$(g-h)(x_{k+1}) \leq (h^* - g^*)(x_k^*) \leq (g-h)(x_k). \tag{4.5}$$

Therefore, if $\lim_{k\to+\infty}(g-h)(x_k) = \beta$, by the sandwich theorem, we conclude that $\lim_{k\to+\infty}(h^*-g^*)(x_k^*) = \beta$. Moreover, from paragraph 1:

$$(g-h)(x_{k+1}) \leq (g-h)(x_k) - \frac{\rho_g + \rho_h}{2}\|\Delta x_k\|^2.$$

If we take limit, we obtain that

$$\beta \leq \beta - \lim_{k\to+\infty} \frac{\rho_g + \rho_h}{2}\|\Delta x_k\|^2,$$

which is similar to

$$\lim_{k\to+\infty}(\rho_g + \rho_h)\|\Delta x_k\|^2 \leq 0.$$

Since $\rho_g + \rho_h > 0$, we conclude that

$$\lim_{k\to+\infty}\|\Delta x_k\|^2 = 0 = \lim_{k\to+\infty} \Delta x_k.$$

Moreover, using (4.4) and (4.5) yields

$$(g-h)(x_{k+1}) \leq g(x_{k+1}) - \langle x_k^*, x_{k+1} - x_k\rangle - h(x_k) \leq (g-h)(x_k).$$

Taking limits again, by the sandwich theorem:

$$\begin{aligned}\lim_{k\to+\infty}(g-h)(x_{k+1}) &= \lim_{k\to+\infty}\{g(x_{k+1}) - \langle x_k^*, x_{k+1} - x_k\rangle - h(x_k)\} \\ &= \lim_{k\to+\infty}(g-h)(x_k).\end{aligned}$$

The second equality implies that

$$\lim_{k\to+\infty}\{g(x_{k+1}) - \langle x_k^*, x_{k+1} - x_k\rangle - g(x_k)\} = 0.$$

Since $x_{k+1} \in \partial g^*(x_k^*)$, this implies that $\langle x_k^*, x_{k+1}\rangle = g(x_{k+1}) + g^*(x_k^*)$. Therefore,

$$\lim_{k\to+\infty}\{g(x_k) + g^*(x_k^*) - \langle x_k^*, x_k\rangle\}.$$

Likewise, the first equality implies that

$$\lim_{k\to+\infty}\{h(x_{k+1})-\langle x_k^*,x_{k+1}-x_k\rangle-h(x_k)\}=0.$$

Since $x_k^*\in\partial h(x_k)$, this implies that $\langle x_k^*,x_{k+1}\rangle=h(x_k)+h^*(x_k^*)$, then

$$\lim_{k\to+\infty}\{h(x_{k+1})+h^*(x_k^*)-\langle x_k^*,x_{k+1}\rangle\}.$$

4. Suppose that α is finite and $\{x_k\}$, $\{x_k^*\}$ bounded. Let be x a limit point of $\{x_k\}$, i.e., $x=\lim_{k\to+\infty}x_k$. By being bounded in a finite-dimensional space, we can suppose (substracting a subsequence) that there exists a sequence $\{x_k^*\}$ that converges to $x^*\in\partial h(x)$. By property 3:

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

Let be $\theta(x^*,x)=g(x)+g^*(x^*)$ for $(x^*,x)\in X^*\times X$. It is clear that $\theta\in\Gamma_0(X^*\times X)$. Then the lower semicontinuity of θ implies

$$\theta(x^*,x)\le\liminf_{k\to+\infty}\theta(x_k^*,x_k)=\lim_{k\to+\infty}\theta(x_k^*,x_k)=\lim_{k\to+\infty}\{\langle x_k^*,x_k\rangle\}=\langle x^*,x\rangle.$$

By Young's inequality, $\theta(x^*,x)=g(x)+g^*(x^*)\ge\langle x^*,x\rangle$. Therefore,

$$\theta(x^*,x)=g(x)+g^*(x^*)=\langle x^*,x\rangle\Longleftrightarrow x^*\in\partial g(x).$$

So that $x^*\in\partial g(x)\cap\partial h(x)\Longleftrightarrow x\in\partial g^*(x^*)\cap\partial h^*(x^*)$.

On the other hand,

$$\begin{aligned}x^*\in\partial h(x)&\Longrightarrow h(x_k)\ge h(x)+\langle x^*,x_k-x\rangle, \quad \forall k.\\ x_k\in\partial h(x_k)&\Longrightarrow h(x)\ge h(x_k)+\langle x_k^*,x-x_k\rangle, \quad \forall k.\end{aligned}$$

so that

$$h(x)+\langle x^*,x_k-x\rangle\le h(x_k)\le h(x)-\langle x_k^*,x-x_k\rangle.$$

Letting $k\to+\infty$, by the sandwich theorem, we obtain that $\lim_{k\to+\infty}h(x_k)=h(x)$. Analogously, we can see that $\lim_{k\to+\infty}h^*(x_k^*)=h^*(x^*)$. Therefore, by Paragraph 3

$$\lim_{k\to+\infty}(g-h)(x_k)=\lim_{k\to+\infty}g(x_k)-\lim_{k\to+\infty}h(x_k)=\lim_{k\to+\infty}g(x_k)-h(x)=\beta,$$
$$\lim_{k\to+\infty}(h^*-g^*)(x_k^*)=\lim_{k\to+\infty}h^*(x_k^*)-\lim_{k\to+\infty}g^*(x_k^*)=h^*(x^*)-\lim_{k\to+\infty}g^*(x_k^*)=\beta.$$

Then, it is sufficient to prove that

$$\begin{aligned}&\lim_{k\to+\infty}g(x_k)=g(x),\\ &\lim_{k\to+\infty}g^*(x_k^*)=g^*(x^*).\end{aligned}$$

Since both limits exist, by Paragraph 3

$$g(x) + g^*(x^*) = \lim_{k\to+\infty} \{g(x_k) + g^*(x_k^*)\} = \lim_{k\to+\infty} g(x_k) + \lim_{k\to+\infty} g^*(x_k^*).$$

By the semicontinuity of g and g^*:

$$\lim_{k\to+\infty} g(x_k) = \liminf_{k\to+\infty} g(x_k) \geq g(x)$$
$$\lim_{k\to+\infty} g^*(x_k^*) = \liminf_{k\to+\infty} g^*(x_k^*) \geq g^*(x^*).$$

Hence, $g(x) + g^*(x^*) \geq \ldots \geq g(x) + g^*(x^*)$. Then, inequalities are equalities and

$$\lim_{k\to+\infty} g(x_k) = g(x),$$
$$\lim_{k\to+\infty} g^*(x_k^*) = g^*(x^*),$$

which concludes the proof.

□

We finish this section with interesting remarks about this theorem.

Remark 4.4.1.

Properties 1 and 2 prove that DCA is a descent method for both primal and dual problems. DCA provides critical points for (P) and (D) after finitely many operations if there is no strict decrease of the primal (or dual) objective function.

If C and D are convex sets such that $\{x_k\} \subset C$ and $\{x_k^*\} \subset D$, then Theorem 4.4.2 remains valid if we replace $\rho(g), \rho(h)$ by $\rho(g, C), \rho(h, C)$ and $\rho(g^*), \rho(h^*)$ by $\rho(g^*, D), \rho(h^*, D)$.

In general, the qualities of DCA (robustness, stability or rate of convergence) depende upon the DC decomposition of the function f. Theorem 4.4.2 shows that strong convexity of DC components influence on DCA.

The DCA provides a x such that $\partial h(x) \subset \partial g(x)$

4.5 Polyhedral DC optimization problems

Polyhedral DC optimization occurs when either g or h is polyhedral convex [7]. This class of DC optimization problems, which is frequently encountered in practice, enjoys interesting properties (from both theoretical and practical viewpoints) concerning local optimality and the convergence of DCA.

Definition 4.5.1. *Let be $C \subset X$. We define the **indicator function**, denoted by χ_C, as:*

$$\chi_C(x) = \begin{cases} 0 & \text{if } x \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

Remark 4.5.1. *The indicator function χ_C is convex if and only if C is convex.*

Definition 4.5.2. *A **function** $\theta \in \Gamma_0(X)$ is said to be **polyhedral convex** if*

$$\theta(x) = \max_{i=1,\ldots,m} \{\langle a_i, x\rangle - \alpha_i\} + \chi_C(x) \quad \forall x \in X,$$

where C is a nonempty polyhedral convex set in X.

In what follows we suppose that in Problem (P) either g or h is polyhedral convex. We may assume that h is a polyhedral convex function. If in (P) g is polyhedral and h is not so, then we consider the dual problem (D), since g^* is polyhedral (see [7]).

Throughout this section we assume that the optimal value α of problem (P) is finite which, by Remark 4.2.1, implies that dom $g \subset$ dom $h = C$. Thus, (P) is equivalent to the problem

$$(\tilde{\mathrm{P}}) \quad \alpha = \inf_{x \in X} \left\{ g(x) - \tilde{h}(x) \right\},$$

where $\tilde{h}(x) = \max_{i \in I}\{\langle a_i, x\rangle - \alpha_i\}$, with $I = \{1, \ldots, m\}$. Clearly

$$\alpha = \inf_{i \in I} \inf_{x \in X} \{g(x) - (\langle a_i, x\rangle - \alpha_i)\}. \tag{4.6}$$

For each $i \in I$, let

$$(\mathrm{P}_i) \quad \beta_i = \inf_{x \in X} \{g(x) - (\langle a_i, x\rangle - \alpha_i)\}.$$

Notation 4.5.1. *We denote by $\tilde{\mathcal{P}}$ the solution set of* $(\tilde{\mathrm{P}})$.

Remark 4.5.2. *If we denote by $\mathcal{P}_i$ the solution set of the problem* (P_i) *we have that* $\mathcal{P}_i = \partial g^*(a_i)$.

Proof:

The proof is very trivial as

$$\begin{aligned} x \in \mathcal{P}_i &\Longleftrightarrow g(x) - \langle a_i, x\rangle + \alpha_i \le g(y) - \langle a_i, y\rangle + \alpha_i, && \forall y \in X \\ &\Longleftrightarrow g(y) \ge g(x) + \langle a_i, y - x\rangle, && \forall y \in X \\ &\Longleftrightarrow a_i \in \partial g(x) \\ &\Longleftrightarrow x \in \partial g^*(a_i). \end{aligned}$$

Therefore, the equality holds. □

Notation 4.5.2. *We are going to denote*

$$J(\alpha) = \{i \in I : \beta_i = \alpha\} \quad \text{and} \quad I(x) = \{i \in I : \langle a_i, x\rangle - \alpha_i = \tilde{h}(x)\}.$$

Theorem 4.5.1. *It is verified that*

1. $x \in \tilde{\mathcal{P}}$ *if and only if* $I(x) \subset J(\alpha)$ *and* $x \in \bigcap_{i \in I(x)} \partial g^*(a_i)$.

2. $\tilde{\mathcal{P}} = \bigcup_{i \in J(\alpha)} \partial g^*(a_i)$. *If* $\{a_i : i \in I\} \subset$ *dom* ∂g^* *then* $\tilde{\mathcal{P}} \neq \emptyset$.

Proof:

1. Let $x \in \tilde{\mathcal{P}}$ and $i \in I(x)$. Then

$$\alpha = g(x) - \tilde{h}(x) = g(x) - (\langle a_i, x\rangle - \alpha_i) = \beta_i,$$

which means that $i \in J(\alpha)$ and $x \in \mathcal{P}_i = \partial g^*(a_i)$ for every $i \in I(x)$. Therefore,

$$I(x) \subset J(\alpha) \quad \text{and} \quad x \in \bigcap_{i \in I(x)} \partial g^*(a_i).$$

Conversely, if $i \in J(\alpha)$ and $x \in \partial g^*(a_i) = \mathcal{P}_i$ then

$$\alpha = \beta_i = g(x) - (\langle a_i, x\rangle - \alpha_i) \geq g(x) - \max_{i \in I}\{\langle a_i, x\rangle - \alpha_i\} = g(x) - \tilde{h}(x) \geq \alpha$$

which implies that $\alpha = g(x) - \tilde{h}(x)$ and $g(x) - (\langle a_i, x\rangle - \alpha_i) = g(x) - \tilde{h}(x)$, i.e., $\langle a_i, x\rangle - \alpha = \tilde{h}(x)$ namely $i \in I(x)$ and $x \in \tilde{\mathcal{P}}$.

2. It is obvious from 1.

□

Lemma 4.5.1. *It is verified that*

1. $\tilde{h}^*(a_i) \leq \alpha_i$, $\forall i \in I$. *Equality holds if and only if there exists* $x \in X$ *such that* $i \in I(x)$.

2. $\tilde{h}(x) = \max_{x^* \in \operatorname{co}\{a_i : i \in I\}}\{\langle x^*, x\rangle - \tilde{h}^*(x^*)\} = \max_{i \in I}\{\langle a_i, x\rangle - \tilde{h}^*(a_i)\}$.

Proof:

1. From the definition of $\tilde{h}$ we have

$$\langle a_i, x\rangle - \tilde{h}(x) \le \alpha_i, \quad \forall x \in X, \forall i \in I,$$

in particular,

$$\tilde{h}^*(a_i) = \sup_{x\in X}\{\langle a_i, x\rangle - \tilde{h}(x)\} \le \alpha_i, \quad \forall i \in I.$$

If there exists $x \in X$ such that $i \in I(x)$, then

$$\tilde{h}(x) + \tilde{h}^*(a_i) \ge \langle a_i, x\rangle = \tilde{h}(x) + \alpha_i \Longleftrightarrow \tilde{h}^*(a_i) \ge \alpha_i,$$

which implies that $\tilde{h}^*(a_i) = \alpha_i$.

Conversely, suppose that $\tilde{h}^*(a_i) = \alpha_i$ for some $i \in I$. Then

$$\tilde{h}^*(a_i) = \sup_{x\in X}\{\langle a_i, x\rangle - \tilde{h}(x)\} = \alpha_i$$

Let $x \in X$ such that $\alpha_i = \langle a_i, x\rangle - \tilde{h}(x)$. Therefore, $\tilde{h}(x) = \langle a_i, x\rangle - \alpha_i$, i.e., $i \in I(x)$.

2. By the fact that dom $\tilde{h}^* =$ co$\{a_i : i \in I\}$ (see [7]) and $\tilde{h} \in \Gamma_0(X)$ we have

$$\tilde{h}(x) = \tilde{h}^{**}(x) = \max_{x^*\in \text{ co}\{a_i:i\in I\}}\{\langle x^*, x\rangle - \tilde{h}^*(x)\}.$$

On the other hand, there exists $x \in X$ such that $i \in I(x)$. Hence, from 1

$$\tilde{h}(x) = \max_{i\in I}\{\langle a_i, x\rangle - \alpha_i\} = \max_{i\in I}\{\langle a_i, x\rangle - \tilde{h}^*(a_i)\},$$

which ends the proof of this lemma.

□

By Lemma 4.5.1, we can write $(\tilde{\text{P}})$ as

$$\alpha = \inf_{x^*\in \text{ co}\{a_i:i\in I\}} \inf_{x\in X} \left\{g(x) - \langle x^*, x\rangle + \tilde{h}^*(x^*)\right\} \tag{4.7}$$

$$\alpha = \inf_{x^*\in \{a_i:i\in I\}} \inf_{x\in X} \left\{g(x) - \langle x^*, x\rangle + \tilde{h}^*(x^*)\right\}. \tag{4.8}$$

Remark 4.5.3. *Problem* (4.7) *is the dual problem* $(\tilde{\text{D}})$ *of* $(\tilde{\text{P}})$

$$(\tilde{\text{D}}) \quad \alpha = \inf_{x^*\in \text{ co}\{a_i:i\in I\}}\{\tilde{h}^*(x^*) - g^*(x^*)\}.$$

However, Problem (4.8) becomes

$$\alpha = \inf_{x^* \in \{a_i : i \in I\}} \{\tilde{h}^*(x^*) - g^*(y)\}.$$

The following result concerning the solution set $\tilde{\mathcal{D}}$ of the dual problem $(\tilde{\mathrm{D}})$ can be proven without using the results in the subsection 4.3.1.

Lemma 4.5.2. *It is verified that*

1. $J(\alpha) = \{i \in I : a_i \in \tilde{\mathcal{D}}$ *and* $\tilde{h}^*(a_i) = \alpha_i\}$.
2. $\{a_i : i \in J(\alpha)\} \subset \tilde{\mathcal{D}}$.

Proof:

If we prove 1, the second result is trivial. So, let $i \in J(\alpha)$ then

$$\begin{aligned}\alpha = \beta_i &= \inf_{x \in X} \{g(x) - (\langle a_i, x\rangle - \alpha_i)\} \\ &= \alpha_i - \inf_{x \in X} \{g(x) - \langle a_i, x\rangle\} \\ &= \alpha_i - \sup_{x \in X} \{\langle a_i, x\rangle - g(x)\} \\ &= \alpha_i - g^*(a_i) \geq \tilde{h}^*(a_i) - g^*(a_i).\end{aligned}$$

But,

$$\alpha = \inf_{x^* \in\, co\{a_i : i \in I\}} \{\tilde{h}^*(x^*) - g^*(x^*)\} \leq \tilde{h}^*(a_i) - g^*(a_i).$$

Hence, the inequality becomes equality, i.e., $a_i \in \tilde{\mathcal{D}}$ and $\tilde{h}^*(a_i) = \alpha_i$.

Conversely, let $i \in I$ such that $a_i \in \tilde{\mathcal{D}}$ and $\tilde{h}^*(a_i) = \alpha_i$. Then

$$\alpha = \tilde{h}^*(a_i) - g^*(a_i) = \alpha_i - g^*(a_i).$$

Thus

$$\beta_i = \inf_{x \in X} \{g(x) - (\langle a_i, x\rangle - \alpha_i)\} = \alpha_i - g^*(a_i) = \alpha.$$

Hence, $i \in J(\alpha)$ as we wanted to see. □

We can summarize the results of this section in the following theorem:

Theorem 4.5.2.

1. $x \in \tilde{\mathcal{P}}$ *if and only if* $I(x) \subset J(\alpha)$ *and* $x \in \bigcap_{i \in I(x)} \partial g^*(a_i)$.
2. $\tilde{\mathcal{P}} = \bigcup_{i \in J(\alpha)} \partial g^*(a_i)$. *If* $\{a_i : i \in I\} \subset$ *dom* ∂g^* *then* $\tilde{\mathcal{P}} \neq \emptyset$.
3. $\tilde{h}(x) = \max_{x^* \in \mathrm{co}\{a_i : i \in I\}} \{\langle x^*, x\rangle - \tilde{h}^*(x^*)\} = \max_{i \in I} \{\langle a_i, x\rangle - \tilde{h}^*(a_i)\}$.
4. $J(\alpha) = \{i \in I : a_i \in \tilde{\mathcal{D}}$ *and* $\tilde{h}^*(a_i) = \alpha_i\}$ *and* $\{a_i : i \in J(\alpha)\} \subset \tilde{\mathcal{D}}$.

Chapter 5

Applications of DCA

In this chapter we present some non-convex problems that DCA have been applied to. The structure of each section will be the same. First, we will introduce the problem. Afterwards, we will obtain a DC decomposition of the objective function and finally we will apply DCA to this decomposition.

We remember that the simplified DCA is given by

Algorithm 1: DCA scheme

input : $x_0 \in X$, tol

$k \leftarrow 0$

repeat

  Compute some $x_k^* \in \partial h(x_k)$.

  Compute $x_{k+1} = \underset{x \in C}{\operatorname{argmin}} \{g(x) - \langle x_k^*, x \rangle\}$.

  $k \leftarrow k + 1$

until $|f(x_{k+1}) - f(x_k)| < \text{tol}$;

output: x_k

5.1 The trust-region subproblem (TSRP)

Let there be given a symmetric matrix $A \in \mathbb{R}^{n \times n}$, a vector $b \in \mathbb{R}^n$ and a real number $R > 0$. The optimization problem

$$\min_{x \in C} \left\{ f(x) := \frac{1}{2} \langle Ax, x \rangle + \langle b, x \rangle \right\},$$

where $C = \{x \in \mathbb{R}^n : \|x\| \leq R\}$, is the *trust-region subproblem.* Since we are minimizing, this problem is equivalent to

$$\min_{x\in X} \left\{ f(x) := \frac{1}{2} \langle Ax, x\rangle + \langle b, x\rangle + \chi_C(x) \right\},$$

where χ_C denotes the indicator function.

Clearly, this problem is a DC program with different DC decompositions. Let us see some examples.

1. $f(x) = g(x) - h(x)$ with

$$g(x) = \frac{1}{2} \langle A_1 x, x\rangle + \langle b, x\rangle + \chi_C(x),$$
$$h(x) = \frac{1}{2} \langle A_2 x, x\rangle .$$

The matrices A_1 and A_2 are symmetric positive semidefinite related to the spectral decomposition of A.

$$A_1 = \sum_{i\in I_+} \lambda_i \langle u_i, u_i\rangle , \quad A_2 = -\sum_{i\in I_-} \lambda_i \langle u_i, u_i\rangle .$$

Here $\lambda_1 \leq \ldots \leq \lambda_n$ are the eigenvalues of A and $\{u_1, \ldots, u_n\}$ is an orthonormal basis of $\mathbb{R}^n$ of eigenvectors of A, $I_+ = \{i : \lambda_i \geq 0\}$ and $I_- = \{i : \lambda_i < 0\}$.

Obviously, since $\langle A_i x, x\rangle , i = 1, 2$ is twice differentiable and its Hessian is $A_i, i = 1, 2$ which is positive semidefinite, we conclude that $\langle A_i x, x\rangle , i = 1, 2$ is convex. In addition, $\langle b, x\rangle$ is affine and C, convex. Therefore, $g(x)$ and $h(x)$ are convex and this is a DC decomposition of $f(x)$.

2. $f(x) = g(x) - h(x)$ with

$$g(x) = \frac{1}{2} \langle (A + \rho I)x, x\rangle + \langle b, x\rangle + \chi_C(x),$$
$$h(x) = \frac{1}{2}\rho\|x\|^2.$$

The positive number ρ is chosen such that $A + \rho I$ is positive semidefinite, i.e., $\rho \geq -\lambda_1$. This is another DC decomposition of $f(x)$ for the same reason as before. The problem is that, at each iteration, the algorithm requires solving a convex quadratic program and so is expensive.

3. $f(x) = g(x) - h(x)$ with

$$g(x) = \frac{\rho}{2}\|x\|^2 + \langle b, x\rangle + \chi_C(x),$$
$$h(x) = \frac{1}{2} \langle (\rho I - A)x, x\rangle .$$

The positive number ρ, in this case, should render the matrix $\rho I - A$ positive semidefinite, i.e., $\rho \geq \lambda_n$. From the computational viewpoint, this decomposition is the most efficient. See [8].

The simplified DCA scheme which is described in the Algorithm 1 can be formulated explicity with the DC decomposition 3.

Since $h(x) = \frac{1}{2}\langle(\rho I - A)x, x\rangle$ is differentiable, $\partial h(x_k) = (\rho I - A)x_k$. Besides, we need to compute

$$x_{k+1} = \underset{x \in C}{\operatorname{argmin}} \{g(x) - \langle x_k^*, x\rangle\},$$

where $x_k^* = (\rho I - A)x_k$.

Then,

$$x_{k+1} = \underset{x \in C}{\operatorname{argmin}} \left\{\frac{\rho}{2}\|x\|^2 + \langle b, x\rangle - \langle x_k^*, x\rangle\right\}.$$

Since multiplying by a constant and adding a constant do not affect the argument function:

$$x_{k+1} = \underset{x \in C}{\operatorname{argmin}} \left\{\|x\|^2 - 2\left\langle \frac{x_k^* - b}{\rho}, x\right\rangle + \left\|\frac{x_k^* - b}{\rho}\right\|^2\right\} = \underset{x \in C}{\operatorname{argmin}} \left\|x - \frac{x_k^* - b}{\rho}\right\|^2.$$

Thus, x_{k+1} is, in fact, the projection of $(x_k^* - b)/\rho$ onto C, i.e.,

$$x_{k+1} = \mathcal{P}_C\left(\frac{x_k^* - b}{\rho}\right) = \begin{cases} R\dfrac{x_k^* - b}{\|x_k^* - b\|}, & \text{if } \|x_k^* - b\| \geq \rho R, \\[2ex] \dfrac{x_k^* - b}{\rho}, & \text{otherwise.} \end{cases}$$

Consequently, the explicit DCA is given by Algorithm 2.

5.2 Least-squares fitting by circles

In this section we are going to consider the following problem: given a set of distinct points $\{a_i\}_{i=1}^p \subseteq \mathbb{R}^n$, find the spherical surface S that minimizes the sum of the squared distances from the points to S, i.e.,

$$\text{(P)} \quad \min_{C,R} f(C, R) := \frac{1}{2p}\sum_{i=1}^{p}\left(\|C - a_i\| - R\right)^2,$$

where C and R are the center and the radius of the spherical surface.

Algorithm 2: DCA scheme for (TSRP)

input : $x_0 \in X$, tol

$k \leftarrow 0$

repeat

  Compute $x_k^* = (\rho I - A)x_k$.

  Compute $x_{k+1} = \begin{cases} R\dfrac{x_k^* - b}{\|x_k^* - b\|}, & \text{if } \|x_k^* - b\| \geq \rho R, \\ \dfrac{x_k^* - b}{\rho}, & \text{otherwise.} \end{cases}$

  $k \leftarrow k + 1$

until $|f(x_{k+1}) - f(x_k)| < \text{tol}$;

output: x_k

Firstly, we need to get a DC decomposition of $f(C, R)$. Since the linear combination of DC functions are DC, we have to obtain a decomposition of one of the sums:

$$\begin{aligned} f_i(C, R) &= (\|C - a_i\| - R)^2 = \|C - a_i\|^2 + R^2 - 2\|C - a_i\|R = \\ &= \|C - a_i\|^2 + R^2 - \left[(\|C - a_i\| + R)^2 - \left(\|C - a_i\|^2 + R^2\right)\right] = \\ &= 2\left(\|C - a_i\|^2 + R^2\right) - (\|C - a_i\| + R)^2 = g_i(C, R) - h_i(C, R). \end{aligned}$$

Since g_i and h_i are convex, we have just found a DC decomposition of $f(C, R)$:

$$\begin{aligned} f(C, R) &= \frac{1}{2p}\sum_{i=1}^{p}\left(\|C - a_i\|^2 + R^2\right) - \frac{1}{2p}\sum_{i=1}^{p}(\|C - a_i\| + R)^2 = \\ &= g(C, R) - h(C, R). \end{aligned}$$

This DC decomposition also provides an explicit DCA that we are going to compute. First of all, we are going to compute a subgradient of h at (C_k, R_k):

$$\begin{aligned} \partial h(C, R_k) &= \partial\left(\frac{1}{p}\sum_{i=1}^{p} h_i(C, R_k)\right) = \frac{1}{2p}\sum_{i=1}^{p}\partial(\|C - a_i\| + R_k)^2 = \\ &= \frac{1}{2p}\sum_{i=1}^{p}\partial(\|C - a_i\|^2 + R_k^2 + 2\|C - a_i\|R_k) = \frac{1}{p}\sum_{i=1}^{p}\left((C - a_i) + \partial\|C - a_i\|R_k\right). \end{aligned}$$

Besides, the subgradient of the Euclidean norm is given by:

$$\partial(\|C - a_i\|) = \begin{cases} \dfrac{C - a_i}{\|C - a_i\|}, & \text{if } C - a_i \neq 0, \\ \{x^* : \|x^*\| \leq 1\}, & \text{otherwise.} \end{cases}$$

Thus, we have:

$$\partial h(C, R_k) = \begin{cases} \dfrac{1}{p}\displaystyle\sum_{i \neq j} \left(1 + \dfrac{R_k}{\|C - a_i\|}\right)(C - a_i), & \text{if } C = a_j, \\ \\ \dfrac{1}{p}\displaystyle\sum_{i=1}^{p} \left(1 + \dfrac{R_k}{\|C - a_i\|}\right)(C - a_i), & \text{otherwise.} \end{cases}$$

On the other hand,

$$\begin{aligned} \partial h(C_k, R) = \partial \left(\frac{1}{2p}\sum_{i=1}^{p} h_i(C_k, R)\right) = \frac{1}{2p}\sum_{i=1}^{p} \partial(\|C_k - a_i\| + R)^2 = \\ = \frac{1}{2p}\sum_{i=1}^{p} \partial(\|C_k - a_i\|^2 + R^2 + 2\|C_k - a_i\|R) = \frac{1}{p}\sum_{i=1}^{p} (R + \|C_k - a_i\|) . \end{aligned}$$

Hence,

$$\partial h(C_k, R) = R + \frac{1}{p}\sum_{i=1}^{p} \|C_k - a_i\|.$$

Therefore, we obtain:

$$(C_k^*, R_k^*) \in \left(\partial h(C, R_k)\Big|_{C_k} \times \partial h(C_k, R)\Big|_{R_k}\right).$$

We also have to compute

$$\begin{pmatrix} C_{k+1} \\ R_{k+1} \end{pmatrix} = \operatorname*{argmin}_{C,R} \left\{ g(C, R) - \langle (C_k^*, R_k^*), (C, R) \rangle \right\},$$

where $(C_k^*, R_k^*) \in \left(\partial h(C, R_k)\Big|_{C_k} \times \partial h(C_k, R)\Big|_{R_k}\right)$. Then,

$$\begin{pmatrix} C_{k+1} \\ R_{k+1} \end{pmatrix} = \operatorname*{argmin}_{C,R} \left\{ \tilde{g}(C, R) := \frac{1}{p}\sum_{i=1}^{p} \left(\|C - a_i\|^2 + R^2\right) - \langle (C_k^*, R_k^*), (C, R) \rangle \right\},$$

but this is a convex quadratic function whose minimum is attained at the point $(\hat{C}, \hat{R})$ such that $\nabla g(\hat{C}, \hat{R}) = 0$. Hence, we need to figure out the gradient:

$$\begin{aligned} \nabla_C g(C, R) &= \frac{1}{p}\sum_{i=1}^{p}(2(C - a_i) - C_k^*), \\ \frac{\partial g(C, R)}{\partial R} &= \frac{1}{p}\sum_{i=1}^{p}(2R - R_k^*). \end{aligned}$$

Thus, we are going to find $(\hat{C}, \hat{R})$ such that

$$\begin{cases} \dfrac{1}{p}\displaystyle\sum_{i=1}^{p}(2(\hat{C}-a_i)-C_k^*)=0, \\ \\ \dfrac{1}{p}\displaystyle\sum_{i=1}^{p}(2\hat{R}-R_k^*)=0. \end{cases}$$

This gives the following solution:

$$\begin{cases} \hat{C}=\dfrac{C_k^*}{2}+\dfrac{1}{p}\displaystyle\sum_{i=1}^{p}a_i. \\ \\ \hat{R}=\dfrac{R_k^*}{2}. \end{cases}$$

Hence, we set

$$\begin{pmatrix} C_{k+1} \\ R_{k+1} \end{pmatrix} = \frac{1}{2}\begin{pmatrix} C_k^* \\ R_k^* \end{pmatrix} + \frac{1}{p}\begin{pmatrix} \sum_{i=1}^{p} a_i \\ 0 \end{pmatrix}.$$

So, Algorithm 3 shows the explicit DCA in this case.

Now, we are going to implement this algorithm in Python and solve an example with this algorithm.

Example 5.2.1. *Given the point cloud of the Picture 5.1, we seek the circumference that fits these points.*

Algorithm 3: DCA scheme for (P)

input : $x_0 \in X$, tol

$k \leftarrow 0$

repeat

Compute $C_k^* = \begin{cases} \dfrac{1}{p} \displaystyle\sum_{i \neq j} \left(1 + \dfrac{R_k}{\|C_k - a_i\|}\right)(C_k - a_i), & \text{if } C_k = a_j, \\ \dfrac{1}{p} \displaystyle\sum_{i=1}^{p} \left(1 + \dfrac{R_k}{\|C_k - a_i\|}\right)(C_k - a_i), & \text{otherwise.} \end{cases}$

Compute $R_k^* = R_k + \dfrac{1}{p} \displaystyle\sum_{i=1}^{p} \|C_k - a_i\|.$

Compute $\begin{pmatrix} C_{k+1} \\ R_{k+1} \end{pmatrix} = \dfrac{1}{2} \begin{pmatrix} C_k^* \\ R_k^* \end{pmatrix} + \dfrac{1}{p} \begin{pmatrix} \sum_{i=1}^{p} a_i \\ 0 \end{pmatrix}.$

$k \leftarrow k + 1$

until $|f(x_{k+1}) - f(x_k)| < \text{tol}$;

output: x_k

The implementation in Python is given by

```python
import math
from numpy import *
from numpy.random import *
import random
import matplotlib.pyplot as plt
from numpy import linalg as LA
from scipy.optimize import minimize
import time
from scipy import stats


def dcafit(x0,nube,timelimit,tol):

    n = len(nube[0])
    n1 = n+1
    p = len(nube)
    cte = 1/p

    def f(x):
        C = x[0:n]
        R = x[-1]
        return sum((LA.norm(C-nube[i])-R)**2 for i in range(p))/(2*p)


    start_time = time.clock()

    def subgrad(x):
```

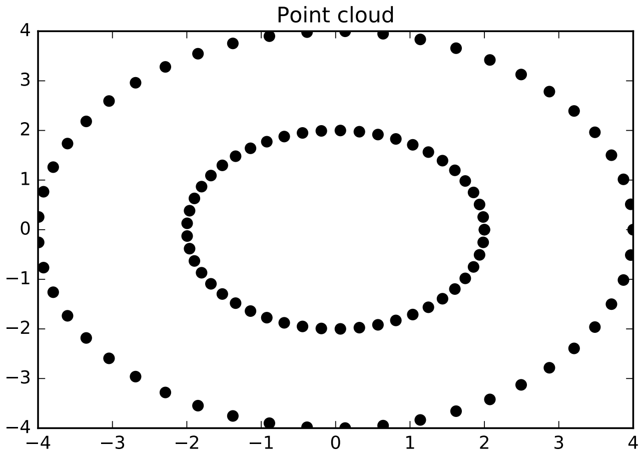

Figure 5.1: Point cloud made of two concentric circles.

```
        C = x[:n]
        R = x[-1]
        vector = zeros(n1)
        vector[0:n] = list(cte*sum((1+R/LA.norm(C-nube[i]))*(C-nube[i]) for i in range(p)))
        vector[-1] = R+cte*(sum(LA.norm(C-nube[i]) for i in range(p)))
        return vector

    def arming(xstar):
        vector = zeros(n1)
        for i in range(n):
            vector[i] = cte*sum(nube[:,i])
        return (1/2)*xstar+vector

    xold = x0
    xstar = subgrad(xold)
    xnew = arming(xstar)

    tiempos = [0]

    tiempo = time.clock()-start_time
    tiempos += [tiempo]

    niter = 1
    while tiempo<=timelimit and abs(f(xnew)-f(xold))>tol:
        xold = xnew
        xstar = subgrad(xold)
        xnew = arming(xstar)
        tiempo = time.clock()-start_time
        tiempos += [tiempo]
        niter += 1

    return [f(xold),xold,tiempos[-2],niter]


def dcamin(x0,nube,timelimit,tol,method):

    n = len(nube[0])
    p = len(nube)

```

```
    def f(x):
        C = x[0:n]
        R = x[-1]
        return (1/(2*p))*sum((LA.norm(C-nube[i])-R)**2 for i in range(p))

    start_time = time.clock()

    xold = x0
    res = minimize(f,xold,method=method,options={'maxiter': 1})
    xnew = res.x

    tiempos = [0]

    tiempo = time.clock()-start_time
    tiempos += [tiempo]

    niter = 1
    while tiempo <= timelimit and abs(f(xnew)-f(xold))>tol:
        xold = xnew
        res = minimize(f,xold,method=method,options={'maxiter': 1})
        xnew = res.x
        tiempo = time.clock()-start_time
        tiempos += [tiempo]
        niter += 1

    return [f(xold),xold,tiempos[-2],niter]
```

The function `dcafit` is the code of the Algorithm 3, whereas that `dcamin` provides the solution of the problem using methods of SciPy. Then, we are going to utilize that to compare results.

Applying this code to this example produces the following code:

```
import math
from numpy import *
from numpy.random import *
import random
import matplotlib.pyplot as plt
from numpy import linalg as LA
from scipy.optimize import minimize
import time
from scipy import stats
from dcafit import *

# Computing the circumference that best adjusts to two concentric circles.

#==============================================================================
# Visualization
#==============================================================================

t = linspace(0,2*pi,50)

circle1 = [2*cos(t),2*sin(t)]
circle2 = [4*cos(t),4*sin(t)]

fig, ax = plt.subplots()
ax.plot(circle1[0],circle1[1],'ko')
ax.plot(circle2[0],circle2[1],'ko')
plt.title('Point cloud')
fig.savefig('nubepuntos.pdf', dpi=fig.dpi)
plt.show()
```

```
#===========================================================================
# Data
#===========================================================================

x0 = zeros(3)

data = array([[2*cos(s),2*sin(s)] for s in t] + [[4*cos(s),4*sin(s)] for s in t])

timelimit = 5

tol = 1e-5

#===========================================================================
# Resolution and visualization
#===========================================================================

res = dcafit(x0,data,timelimit,tol)

obj = res[0]
C = res[1][:-1]
R = res[1][-1]
tiempo = res[2]

print('Objective value: ',obj)
print('Center of the circumference: ',C)
print('Radius of the circumference: ',R)
print('Execution time: ',tiempo)

circlesol = C+array([[R*cos(s),R*sin(s)] for s in t])

fig, ax = plt.subplots()
ax.plot(circle1[0],circle1[1],'ko')
ax.plot(circle2[0],circle2[1],'ko')
ax.plot(circlesol[:,0],circlesol[:,1],'r',label='Circumference solution')

plt.title('Point cloud with circumference solution')
#handles, labels = ax.get_legend_handles_labels()
#ax.legend(handles, labels)
fig.savefig('nubepuntoscirc.pdf', dpi=fig.dpi)
plt.show()
```

DCA algorithm provides the solution 5.2.

Numerical results

In this subsection we will study two experiments to compare our algorithm with others that are implemented in the funcion `optimize` of the package SciPy. First experiment consists in using Multistart Algorithm with 100 uniform samples in the interval $(-2000, 2000)$ to do a boxplot to see the results of the objective values and execution times. In the second experiment, we will present a table to see how the results change in terms of the size p and dimension n in the different algorithms.

First experiment

Given 50 points distributed by an uniform sample in the interval $(0, 100)$ in the Euclidean space, we will compute the objective values and times provided by the different

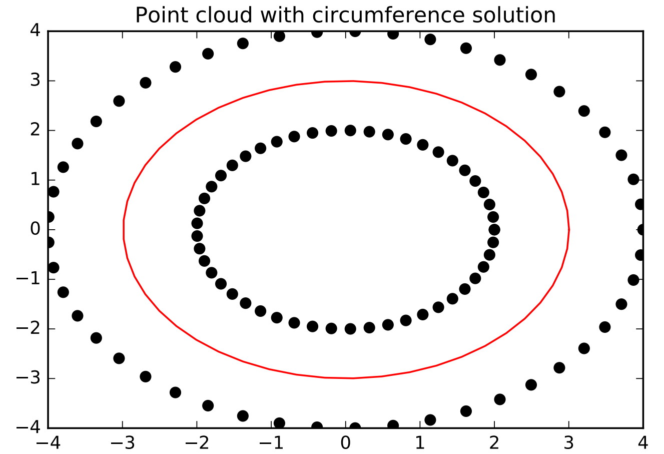

Figure 5.2: The red circumference is the solution of the example algorithms.

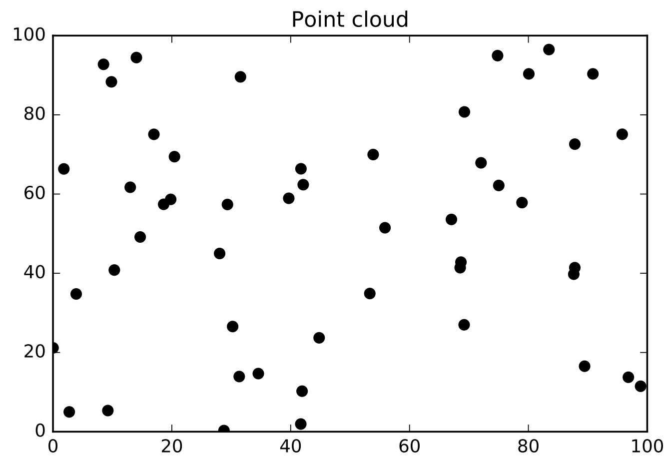

Figure 5.3: Point cloud of the first experiment

The Python's code is the following:

```
import math
from numpy import *
import matplotlib.pyplot as plt
from numpy import linalg as LA
from scipy.optimize import minimize
import time
from scipy import stats
from dcafit import *


```

```
seed(1)

# Multistart and comparation with other algorithms.

#==============================================================================
# Visualization
#==============================================================================
n = 2
n1 = n + 1
p = 50

pointcloud = uniform(0,100,size=(n,p))

fig, ax = plt.subplots()
ax.plot(pointcloud[0],pointcloud[1],'ko')
plt.title('Point cloud')
fig.savefig('nubepuntos2.pdf', dpi=fig.dpi)
plt.show()

#==============================================================================
# Data
#==============================================================================


M = 2000
numsamples = 100


initialpoints = uniform(-M,M,size=(n1,numsamples))

data = uniform(0,100,size=(p,n))

timelimit = 0.03

tol = 1e-3

#==============================================================================
# Multistart DCA
#==============================================================================
print('Multistart DCA')

DCAobj = []

DCAtimes = []

niter = 0

for i in range(numsamples):
    if niter%10==0:
        print('Iteration ',niter)
    res = dcafit(initialpoints[:,i],data,timelimit,tol)
    DCAobj += [res[0]]
    if res[2]!=0:
        DCAtimes += [res[2]]
    niter += 1

asarray(DCAobj)

asarray(DCAtimes)
#==============================================================================
# Multistart Nelder-Mead
#==============================================================================
print('\nMultistart Nelder-Mead')

```

```
NMobj = []

NMtimes = []

niter = 0

for i in range(numsamples):
    if niter%10==0:
        print('Iteration ',niter)
    res = dcamin(initialpoints[:,i],data,timelimit,tol,method='Nelder-Mead')
    NMobj += [res[0]]
    if res[2]!=0:
        NMtimes += [res[2]]
    niter += 1

asarray(NMobj)

asarray(NMtimes)

#===============================================================================
# Multistart Powell
#===============================================================================
#print('\nMultistart Powell')
#
#Powellobj = []
#
#Powelltimes = []
#
#niter = 0
#
#for i in range(numsamples):
#    if niter%10==0:
#        print('Iteration ',niter)
#    res = dcamin(initialpoints[:,i],data,timelimit,tol,method='Powell')
#    Powellobj += [res[0]]
#    if res[2]!=0:
#        Powelltimes += [res[2]]
#    niter += 1
#
#asarray(Powellobj)
#
#asarray(Powelltimes)

#===============================================================================
# Multistart CG
#===============================================================================
print('\nMultistart CG')

CGobj = []

CGtimes = []

niter = 0

for i in range(numsamples):
    if niter%10==0:
        print('Iteration ',niter)
    res = dcamin(initialpoints[:,i],data,timelimit,tol,method='CG')
    CGobj += [res[0]]
    if res[2]!=0:
        CGtimes += [res[2]]
    niter += 1

asarray(CGobj)
```

```

asarray(CGtimes)
stats.describe(CGobj)
#==============================================================================
# Multistart BFGS
#==============================================================================
print('\nMultistart BFGS')

BFGSobj = []

BFGStimes = []

niter = 0

for i in range(numsamples):
    if niter%10==0:
        print('Iteration ',niter)
    res = dcamin(initialpoints[:,i],data,timelimit,tol,method = 'BFGS')
    BFGSobj += [res[0]]
    if res[2]!=0:
        BFGStimes += [res[2]]
    niter += 1

asarray(BFGSobj)

asarray(BFGStimes)

#==============================================================================
# Boxplot
#==============================================================================

# Objective values

fig, ax = plt.subplots()

ax.boxplot([DCAobj,NMobj,CGobj,BFGSobj])

algorithms = ['DCA', 'Nelder-Mead','CGobj','BFGS']
labels = algorithms

ax.set_xticklabels(labels)
plt.title('Objective values of the algorithms')
fig.savefig('objval.pdf', dpi=fig.dpi)
plt.show()

# Objective value DCA

fig, ax = plt.subplots()

ax.boxplot(DCAobj)

algorithms = ['DCA']
labels = algorithms

ax.set_xticklabels(labels)
plt.title('Objective values of the DCA')
fig.savefig('objvaldca.pdf', dpi=fig.dpi)
plt.show()

# Execution times

fig, ax = plt.subplots()

ax.boxplot([DCAtimes,NMtimes,CGtimes,BFGStimes])
```

```
algorithms = ['DCA', 'Nelder-Mead','CG', 'BFGS']
labels = algorithms

ax.set_xticklabels(labels)
plt.title('Execution times of the algorithms')
fig.savefig('extimes.pdf', dpi=fig.dpi)
plt.show()
```

The boxplot 5.4 gives the comparation of the objective values.

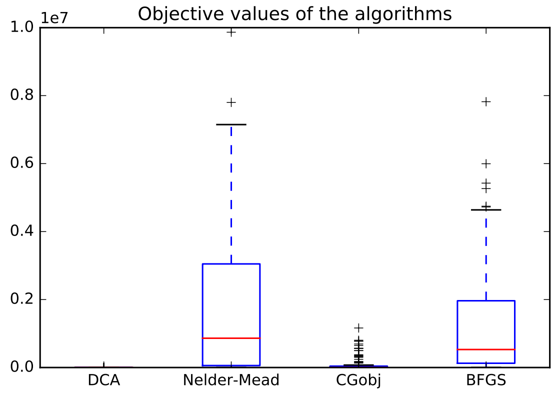

Figure 5.4: Boxplot of objective values

Clearly, the objective values provided by DCA are much better than the other algorithms. If we focus on the DCA values, we have this boxplot

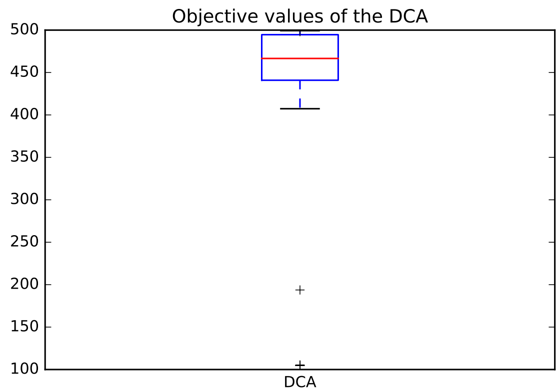

Figure 5.5: Boxplot of DCA objective values

Lastly, if we see the execution times, we obtain the boxplot 5.6. In this case, the execution times are more similar, but the DCA times seem to be better than the others.

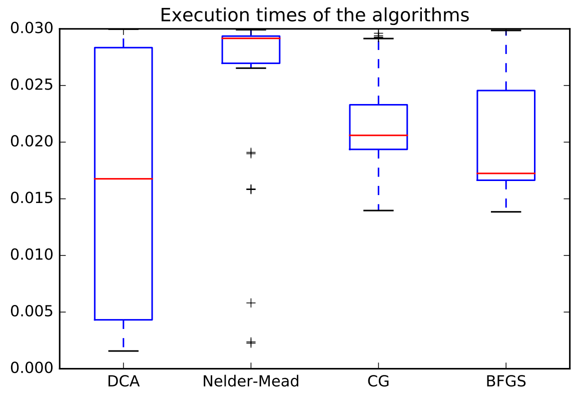

Figure 5.6: Boxplot of execution times

Second experiment

In this experiment, using the same Multistart Algorithm, we are varying the parameters p and n and we will expose the main features of the distinct algorithms for each pair of parameters in a table. The code to implement this experiment is the following:

```
import math
from numpy import *
import matplotlib.pyplot as plt
from numpy import linalg as LA
from scipy.optimize import minimize
import time
from scipy import stats
from dcafit import *



seed(1)


# Mean and Multistart and comparation with other algorithms.
#==============================================================================
# Data
#==============================================================================

pointcloud = [10,50,100,200]

dimension = [1,2,5,10]

M = 2000
numsamples = 100

#==============================================================================
# Initialization
#==============================================================================
DCAmean = []
NMmean = []
CGmean = []
BFGSmean = []

```

```
DCAmin = []
NMmin = []
CGmin = []
BFGSmin = []

DCAmax = []
NMmax = []
CGmax = []
BFGSmax = []


#=========================================================================
# Loops
#=========================================================================
for p in pointcloud:
    for n in dimension:
        n1 = n + 1

        initialpoints = uniform(-M,M,size=(n1,numsamples))

        data = uniform(0,100,size=(p,n))

        timelimit = 0.0003*n*p

        tol = 1e-3

        #=================================================================
        # Multistart DCA
        #=================================================================
        print('\nMultistart DCA: p = '+str(p)+', n = '+str(n))

        DCAobj = []

        DCAtimes = []

        DCAniter = []

        for i in range(numsamples):
            res = dcafit(initialpoints[:,i],data,timelimit,tol)
            DCAobj += [res[0]]
            if res[2]!=0:
                DCAtimes += [res[2]]
                DCAniter += [res[3]]

        asarray(DCAobj)

        asarray(DCAtimes)

        asarray(DCAniter)



        #=================================================================
        # Multistart Nelder-Mead
        #=================================================================
        print('\nMultistart Nelder-Mead: p = '+str(p)+', n = '+str(n))

        NMobj = []
        NMtimes = []
        NMniter = []

        for i in range(numsamples):
            res = dcamin(initialpoints[:,i],data,timelimit,tol,method='Nelder-Mead')
            NMobj += [res[0]]
```

```
        if res[2]!=0:
            NMtimes += [res[2]]
            NMniter += [res[3]]

    asarray(NMobj)
    asarray(NMtimes)
    asarray(NMniter)

    #======================================================================
    # Multistart Powell
    #======================================================================
    #print('\nMultistart Powell')
    #
    #Powellobj = []
    #
    #Powelltimes = []
    #
    #niter = 0
    #
    #for i in range(numsamples):
    #    if niter%10==0:
    #        print('Iteration ',niter)
    #    res = dcamin(initialpoints[:,i],data,timelimit,tol,method='Powell')
    #    Powellobj += [res[0]]
    #    if res[2]!=0:
    #        Powelltimes += [res[2]]
    #    niter += 1
    #
    #asarray(Powellobj)
    #
    #asarray(Powelltimes)

    #======================================================================
    # Multistart CG
    #======================================================================
    print('\nMultistart CG: p = '+str(p)+', n = '+str(n))

    CGobj = []
    CGtimes = []
    CGniter = []

    for i in range(numsamples):
        res = dcamin(initialpoints[:,i],data,timelimit,tol,method='CG')
        CGobj += [res[0]]
        if res[2]!=0:
            CGtimes += [res[2]]
            CGniter += [res[3]]

    asarray(CGobj)
    asarray(CGtimes)
    asarray(CGniter)

    #======================================================================
    # Multistart BFGS
    #======================================================================
    print('\nMultistart BFGS: p = '+str(p)+', n = '+str(n))

    BFGSobj = []
    BFGStimes = []
    BFGSniter = []

    for i in range(numsamples):
        res = dcamin(initialpoints[:,i],data,timelimit,tol,method = 'BFGS')
        BFGSobj += [res[0]]
```

```
            if res[2]!=0:
                BFGStimes += [res[2]]
                BFGSniter += [res[3]]

        asarray(BFGSobj)
        asarray(BFGStimes)
        asarray(BFGSniter)

        #======================================================================
        # Means, minimum and maximum of objective values, iterations and times
        #======================================================================

        DCAmean += [(mean(DCAobj),mean(DCAtimes),floor(mean(DCAniter)))]
        NMmean += [(mean(NMobj),mean(NMtimes),floor(mean(NMniter)))]
        CGmean += [(mean(CGobj),mean(CGtimes),floor(mean(CGniter)))]
        BFGSmean += [(mean(BFGSobj),mean(BFGStimes),floor(mean(BFGSniter)))]

        DCAmin += [(min(DCAobj),min(DCAtimes),floor(min(DCAniter)))]
        NMmin += [(min(NMobj),min(NMtimes),floor(min(NMniter)))]
        CGmin += [(min(CGobj),min(CGtimes),floor(min(CGniter)))]
        BFGSmin += [(min(BFGSobj),min(BFGStimes),floor(min(BFGSniter)))]

        DCAmax += [(max(DCAobj),max(DCAtimes),floor(max(DCAniter)))]
        NMmax += [(max(NMobj),max(NMtimes),floor(max(NMniter)))]
        CGmax += [(max(CGobj),max(CGtimes),floor(max(CGniter)))]
        BFGSmax += [(max(BFGSobj),max(BFGStimes),floor(max(BFGSniter)))]
```

This code provides the following tables:

Objective values

		DCA			NM			CG			BFGS		
p	n	Min	Ave	Max	Min	Ave	Max	Min	Ave	Max	Min	Ave	Max
10	1	20.47	142.6	202.7	202.7	1.36e+6	6.86e+6	20.47	7919	5.99e+5	22.93	4.16e+4	4.02e+5
10	2	74.85	327.6	695.4	170.6	1.76e+6	9.16e+6	74.84	294.6	695.347	156.6	4265	1.67e+5
10	5	30.04	293.5	645.1	63.05	3.05e+6	1.45e+7	30.05	276.3	642.30	44.44	671.2	2.86e+4
10	10	66.78	325.9	870	186.6	5.29e+6	1.87e+7	65.09	318.2	941.432	78.09	363.66	968.9
50	1	98.82	255.1	350.6	350.7	1.51e+6	7.22e+6	98.82	272.9	623.823	350.6	5.01e+4	9.3e+5
50	2	93.50	370.3	459.2	399.8	1.78e+6	8.76e+6	93.5	374.7	454.274	265.6	2095	5.87e+4
50	5	65.52	375.7	554.4	256.3	3.68e+6	1.25e+7	65.51	367.2	554.793	65.77	389.4	555
50	10	49.05	380.9	533.5	248.1	5.01e+6	1.76e+7	49.02	383.8	547.507	239.8	3561	2.54e+5
100	1	95.24	331.2	405.8	406.3	1.02e+6	7.23e+6	95.24	4980	4.63e+5	95.6	3.22e+4	5.27e+5
100	2	94.80	394.7	486.7	411.3	1.67e+6	8.96e+6	94.79	396.5	485.520	122.1	7.42e+4	1.84e+6
100	5	84.08	385.7	509.6	309.2	3.44e+6	1.25e+7	84.07	384.5	509.606	84.1	448.6	5895
100	10	296.6	418.3	510	295.6	5.60e+6	2.04e+7	291.5	412.5	504.670	296.5	428.1	844.6
200	1	102.7	329.9	409.7	409.7	1.02e+6	5.79e+6	102.7	372	1857.803	409.7	1.66e+4	3.87e+5
200	2	109.1	362.5	422.1	408.1	1.53e+6	9.39e+6	109.1	369.6	422.140	109.2	6351	3.91e+5
200	5	87.28	417.7	474.9	400.6	3.68e+6	1.20e+7	87.27	414.3	473.241	87.37	441.8	2024
200	10	294.7	403.8	528.4	344.9	5.92e+6	2.03e+7	328.8	401.1	531.285	331.1	407.6	531.7

Execution times

p	n	DCA Min	DCA Ave	DCA Max	NM Min	NM Ave	NM Max	CG Min	CG Ave	CG Max	BFGS Min	BFGS Ave	BFGS Max
10	1	3e-4	0.003	0.03	4e-4	0.014	0.03	0.001	0.01	0.03	0.001	0.028	0.03
10	2	4e-4	0.055	0.06	0.001	0.015	0.054	0.004	0.056	0.06	0.055	0.059	0.06
10	5	0.06	0.148	0.15	0.001	0.044	0.15	0.107	0.147	0.15	0.14	0.148	0.15
10	10	0.3	0.3	0.3	0.005	0.078	0.237	0.288	0.297	0.3	0.292	0.298	0.3
50	1	0.002	0.019	0.124	0.002	0.532	0.148	0.011	0.049	0.149	0.109	0.143	0.15
50	2	0.002	0.117	0.3	0.002	0.079	0.298	0.017	0.254	0.3	0.139	0.291	0.3
50	5	0.165	0.732	0.75	0.004	0.145	0.745	0.177	0.729	0.75	0.733	0.745	0.75
50	10	0.655	1.49	1.5	0.025	0.29	0.983	1.382	1.483	1.5	1.407	1.488	1.5
100	1	0.004	0.026	0.154	0.004	0.16	0.3	0.021	0.087	0.297	0.112	0.283	0.3
100	2	0.003	0.285	0.6	0.012	0.166	0.526	0.033	0.498	0.6	0.435	0.572	0.6
100	5	0.008	1.402	1.5	0.04	0.289	1.344	0.06	1.435	1.5	1.271	1.486	1.5
100	10	0.009	2.966	3.000	0.152	0.667	2.239	2.857	2.969	2.999	2.906	2.981	3
200	1	0.006	0.06	0.338	0.023	0.254	0.597	0.05	0.185	0.595	0.225	0.567	0.6
200	2	0.006	0.226	1.2	0.009	0.259	1.195	0.055	0.856	1.199	0.622	1.124	1.199
200	5	0.007	1.773	2.999	0.059	0.551	1.97	0.118	2.768	3	1.252	2.829	2.999
200	10	0.016	5.52	6.000	0.293	1.196	3.85	5.737	5.944	6	3.934	5.946	6

Iterations

p	n	DCA Min	DCA Ave	DCA Max	NM Min	NM Ave	NM Max	CG Min	CG Ave	CG Max	BFGS Min	BFGS Ave	BFGS Max
10	1	2	5	15	2	21	45	2	4	11	2	11	18
10	2	2	99	112	2	19	66	2	23	37	8	21	33
10	5	105	252	265	2	30	124	16	39	63	14	34	57
10	10	297	455	483	4	40	114	35	56	80	25	47	70
50	1	2	9	55	2	20	55	2	5	14	6	12	20
50	2	2	42	121	2	21	84	2	20	41	12	22	38
50	5	65	277	292	2	28	139	10	48	69	30	46	63
50	10	250	552	574	4	33	109	37	54	88	8	47	69
100	1	2	5	26	2	23	46	2	4	11	8	12	19
100	2	2	50	125	3	23	79	2	21	42	4	22	42
100	5	3	255	308	5	27	130	2	46	74	13	47	73
100	10	3	539	618	11	40	134	57	57	95	22	59	81
200	1	2	7	36	3	21	55	2	4	12	7	13	21
200	2	2	22	123	2	19	89	2	16	37	8	22	30
200	5	2	148	301	4	24	88	2	39	73	22	39	70
200	10	3	479	626	7	33	80	30	52	90	32	63	79

These results show that, indeed, the DCA is a promising strategy against benchmark competitors.

Bibliography

[1] E. Asplund. Differentiability of the metric projection in finite dimensional euclidean space. *Proceedings of the American Mathematical Society*, 38(1):218–219, 1973.

[2] Rafael Blanquero and Emilio Carrizosa. Optimization of the norm of a vector-valued dc function and applications. *Journal of Optimization Theory and Applications*, 107(2):245–260, 2000.

[3] Mirjam Dür. Conditions characterizing minima of the difference of functions. *Monatshefte für Mathematik*, 134:295–303, 2002.

[4] Philip Hartman. On functions representable as a difference of convex functions. *Pacific Journal of Mathematics*, 9(3):707–713, 1959.

[5] R. Horst and N.V. Thoai. Dc programming: Overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, 1999.

[6] Panos M. Pardalos and Reiner Horst. *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1st edition, 1995.

[7] Rockafellar R.T. *Convex Analysis*. Princeton University Press, New Jersey, 1970.

[8] Pham Dinh Tao and Le Thi Hoai An. Convex analysis approach to dc programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355, 1997.