Sequential Difference-of-Convex Programming

Welington de Oliveira
www.oliveira.mat.br

MINES ParisTech, CMA - Centre de Mathématiques Appliquées

École des Ponts, November 18th, 2021

Outline

A brief tutorial on DC programming

Sequential DC programming

DC programming

We are now interested in a class of Difference-of-Convex (DC) programming:

$$\min_{x \in X} f(x), \quad \text{with} \quad f(x) = f_1(x) - f_2(x) \qquad (P)$$

Assumptions

- $f_1, f_2 : \mathbb{R}^n \to \mathbb{R}$ are convex but possibly nonsmooth
- $X \subset \mathbb{R}^n$ is a closed convex set (e.g. $X = \mathbb{R}^n$)

- DC programming is a sub-field of nonlinear programming that finds many applications in engineering problems and data science
- DC programs are, in the general situation, NP-hard
- They cover a broad class of nonconvex optimization problems, but still allows the use of the convex analysis apparatus to establish optimality conditions and design algorithms

Universality of DC functions

Lower-C^2 functions are globally DC. In particular,

- finite maxima of C^2-functions are DC
- finite maxima of functions with gradient Lipschitz continuous are DC
- polynomials are DC

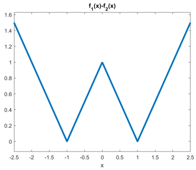
Every extended real-valued lsc function can be approximated by not only DoC, but actually piece-wise affine DC of the kind max-max

This result shows that the class of optimization problems fitting into formulation (P) is comprehensive, and hence covers almost all problems of practical interest

Note, however, that a DC decomposition of f is not always available

In many situations of practical interest, a DC decomposition can be easily obtained

Example: $f(x) = ||x| - 1|$ is DC because
$f(x) = 2\max\{|x| - 1, 0\} - [|x| - 1]$



A DC function has infinitely many DC decompositions $f_1 - f_2$

Universality of DC functions

Let $f_i = \psi_i - \phi_i$ be DC functions for all $i = 1, \ldots, m$. The following functions are DC

- $g_1(x) = \sum_{i=1}^{m} \alpha_i f_i(x)$, $\alpha \in \mathbb{R}^m$, with DC decomposition

$$g_1(x) = \left(\sum_{j \in \{i:\alpha_i > 0\}} \alpha_j \psi_j(x) - \sum_{j \in \{i:\alpha_i < 0\}} \alpha_j \phi_j(x) \right) - \left(\sum_{j \in \{i:\alpha_i > 0\}} \alpha_j \phi_j(x) - \sum_{j \in \{i:\alpha_i < 0\}} \alpha_j \psi_j(x) \right)$$

Centre de Mathématiques Appliquées

Universality of DC functions

Let $f_i = \psi_i - \phi_i$ be DC functions for all $i = 1, \ldots, m$. The following functions are DC

- $g_1(x) = \sum_{i=1}^{m} \alpha_i f_i(x)$, $\alpha \in \mathbb{R}^m$, with DC decomposition

$$g_1(x) = \left(\sum_{j \in \{i : \alpha_i > 0\}} \alpha_j \psi_j(x) - \sum_{j \in \{i : \alpha_i < 0\}} \alpha_j \phi_j(x) \right) - \left(\sum_{j \in \{i : \alpha_i > 0\}} \alpha_j \phi_j(x) - \sum_{j \in \{i : \alpha_i < 0\}} \alpha_j \psi_j(x) \right)$$

- $g_2(x) = \max\{f_1(x), \ldots, f_m(x)\}$ with DC decomposition

$$g_2(x) = \max_i \left\{ \psi_i(x) + \sum_{j \neq i}^{m} \phi_i(x) \right\} - \sum_{j=1}^{m} \phi_j(x)$$

Universality of DC functions

Let $f_i = \psi_i - \phi_i$ be DC functions for all $i = 1, \dots, m$. The following functions are DC

- $g_1(x) = \sum_{i=1}^m \alpha_i f_i(x)$, $\alpha \in \mathbb{R}^m$, with DC decomposition

$$g_1(x) = \left(\sum_{j \in \{i : \alpha_i > 0\}} \alpha_j \psi_j(x) - \sum_{j \in \{i : \alpha_i < 0\}} \alpha_j \phi_j(x) \right) - \left(\sum_{j \in \{i : \alpha_i > 0\}} \alpha_j \phi_j(x) - \sum_{j \in \{i : \alpha_i < 0\}} \alpha_j \psi_j(x) \right)$$

- $g_2(x) = \max\{f_1(x), \dots, f_m(x)\}$ with DC decomposition

$$g_2(x) = \max_i \left\{ \psi_i(x) + \sum_{j \neq i}^m \phi_i(x) \right\} - \sum_{j=1}^m \phi_j(x)$$

- $g_3(x) = \min\{f_1(x), \dots, f_m(x)\}$ with DC decomposition

$$g_3(x) = \sum_{j=1}^m \psi_j(x) - \max_i \left\{ \phi_i(x) + \sum_{j \neq i}^m \psi_j(x) \right\}$$

Let $f_i = \psi_i - \phi_i$ be DC functions for all $i = 1, \ldots, m$. The following functions are DC

- $g_1(x) = \sum_{i=1}^m \alpha_i f_i(x)$, $\alpha \in \mathbb{R}^m$, with DC decomposition

$$g_1(x) = \left(\sum_{j \in \{i : \alpha_i > 0\}} \alpha_j \psi_j(x) - \sum_{j \in \{i : \alpha_i < 0\}} \alpha_j \phi_j(x) \right) - \left(\sum_{j \in \{i : \alpha_i > 0\}} \alpha_j \phi_j(x) - \sum_{j \in \{i : \alpha_i < 0\}} \alpha_j \psi_j(x) \right)$$

- $g_2(x) = \max\{f_1(x), \ldots, f_m(x)\}$ with DC decomposition

$$g_2(x) = \max_i \left\{ \psi_i(x) + \sum_{j \neq i}^m \phi_i(x) \right\} - \sum_{j=1}^m \phi_j(x)$$

- $g_3(x) = \min\{f_1(x), \ldots, f_m(x)\}$ with DC decomposition

$$g_3(x) = \sum_{j=1}^m \psi_j(x) - \max_i \left\{ \phi_i(x) + \sum_{j \neq i}^m \psi_j(x) \right\}$$

- $g(x) = |f(x)|$ with DC decomposition

$$g(x) = 2 \max\{\psi(x), \phi(x)\} - (\psi(x) + \phi(x))$$

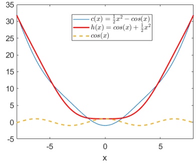
Centre de Mathématiques Appliquées

Universality of DC functions

Lemma

Let $f : \mathbb{R}^n \to \mathbb{R}$ have Lipschitz continuous gradient with modulus $L > 0$. Then f admits the DC decompositions $f(x) = \frac{L}{2}\|x\|^2 - \left[\frac{L}{2}\|x\|^2 - f(x)\right]$ *and* $f(x) = \left[f(x) + \frac{L}{2}\|x\|^2\right] - \frac{L}{2}\|x\|^2$

Example: $f(x) = \cos(x)$, that has gradient Lipschitz with constant $L = 1$



Lemma

Let $\psi : \mathbb{R}^n \to \mathbb{R}_+$ be a convex function. If $\phi : \mathbb{R}_+ \to \mathbb{R}$ is a concave and non-decreasing function such that $\phi'_+(0) < \infty$, then

$$\tau\,\psi(x) - \phi(\psi(x)) \quad \textit{is convex for all } \tau \geq \phi'_+(0)$$

Such a property is useful for inducing sparsity in certain problems

$$\boxed{\phi(\psi(x)) = \tau\psi(x) - [\tau\psi(x) - \phi(\psi(x))]}$$

Centre de Mathématiques Appliquées

Inducing sparsity

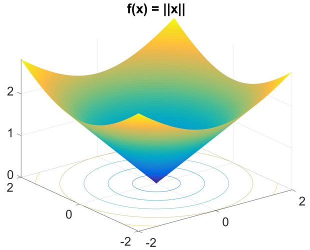


Figura: Euclidean norm

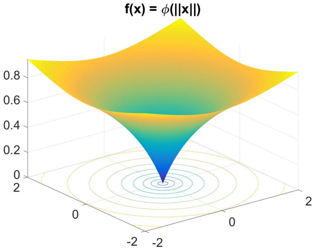
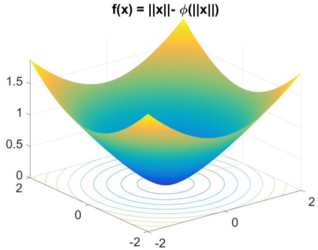


Figura: $\phi(r) = \log(1+2r)/2$



$$\phi(\|x\|) = \|x\| - [\|x\| - \phi(\|x\|)] \text{ is DC}$$

- Let $b \in \mathbb{R}^n$ be the vectorization of a corrupted grayscale image. In order to preserve edges in the process of restoring b, the following nonconvex *Total Variation* formulation is commonly employed[1]:

$$\min_{x \in \mathbb{R}^n} \frac{\mu}{2}\|x-b\|^2 + \phi(TV(x)), \quad \text{with} \quad TV(x) = \sum_{i=1}^{n} \|(\nabla x)_i\|$$

- $\phi : \mathbb{R}_+ \to \mathbb{R}$ is a penalizing function, $\mu > 0$ is a fidelity parameter and $(\nabla x)_i \in \mathbb{R}^2$ denotes the discretization of the gradient of image x at pixel i
- If ϕ is concave and non-decreasing, its right derivative $\phi'_+(r)$ is well defined for all $r \geq 0$. Then $\tau TV(x) - \phi(TV(x))$ is convex function for all $\tau \geq \phi'_+(0)$
- The nonconvex image denoising problem fits into the DC formulation

$$\min_{x \in \mathbb{R}^n} f_1(x) - f_2(x)$$

$$f_1(x) = \tfrac{\mu}{2}\|x-b\|^2 + \tau TV(x) \quad \text{and} \quad f_2(x) = \tau TV(x) - \phi(TV(x)),$$

two convex functions

[1] $\|(\nabla x)_i\| = \sqrt{(X_{l+1,j} - X_{l,j})^2 + (X_{l,j+1} - X_{l,j})^2}$ with i^{th} the coordinate of x where the pixel $X_{l,j}$ is stored

Corrupted image

Convex model: FISTA (best setting)

$$\min_{x \in \mathbb{R}^n} \frac{\mu}{2}\|x - b\|^2 + TV(x)$$

Nonconvex model: DC programming

$$\min_{x \in \mathbb{R}^n} \frac{\mu}{2}\|x - b\|^2 + \phi(TV(x))$$

- Let $a_i \in \mathbb{R}$ and $\psi_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, be convex functions. The (NP-hard) clipped optimization problem reads as

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} \min\{\psi_i(x), a_i\}$$

  and finds applications in statistics, risk minimization, clipped control, and machine learning

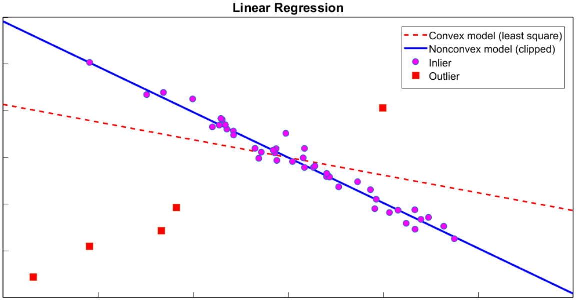
- The sum of clipped convex functions is indeed a DC function:

$$f(x) = \sum_{i=1}^{m} \min\{\psi_i(x), a_i\} = f_1(x) - f_2(x)$$

$f_1(x) = \sum_{i=1}^{m} \psi_i(x) + a_i$ and $f_2(x) = \sum_{i=1}^{m} \max\{\psi_i(x), a_i\}$

Clipped linear regression

- Convex model: $\min_{x\in\mathbb{R}} 0.2\,x^2 + \sum_{i=1}^{m}(xp_i - q_i)^2$
- Nonconvex, nonsmooth model: $\min_{x\in\mathbb{R}} 0.2\,x^2 + \sum_{i=1}^{m} \min\{(xp_i - q_i)^2, 0.5\}$



- Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be a convex function, X a convex set, and $\lambda > 0$ a given parameter
- In sparse optimization we may wish to solve

$$\min_{x \in X} \phi(x) + \lambda \|x\|_0$$

- The (difficult) zero norm counts the number of nonzeros elements of vector x
- We may approximate $\|x\|_0$ by a simpler function: given $r > 0$ small enough

$$\|x\|_0 \quad \approx \quad \sum_{i=1}^n \min\left\{\frac{|x_i|}{r}, 1\right\} \quad = \quad \sum_{i=1}^n \left(\frac{|x_i|}{r} + 1\right) - \sum_{i=1}^n \max\left\{\frac{|x_i|}{r}, 1\right\}$$

- Note that $\lim_{r \downarrow 0} \sum_{i=1}^n \min\left\{\frac{|x_i|}{r}, 1\right\} = \|x\|_0$
- Then

$$\min_{x \in X} \phi(x) + \lambda \|x\|_0 \quad \approx \quad \min_{x \in X} \; f_1(x) - f_2(x)$$

$$f_1(x) = \phi(x) + \lambda \sum_{i=1}^n \left(\frac{|x_i|}{r} + 1\right) \quad \text{and} \quad f_2(x) = \lambda \sum_{i=1}^n \max\left\{\frac{|x_i|}{r}, 1\right\}$$

- In sparse optimization we may wish to solve

$$\begin{cases} \min\limits_{x \in X} & \phi(x) \\ \text{s.t.} & \|x\|_0 \leq k \end{cases}$$

(k is a natural number)

- By using the norm $\|x\|_{(k)} = \sum_{i=1}^{k} |x_{\sigma(i)}|$, with $\sigma(i)$ the i^{th} largest value of $\{|x_1|, \dots, |x_n|\}$, we have that

$$\begin{cases} \min\limits_{x \in X} & \phi(x) \\ \text{s.t.} & \|x\|_0 \leq k \end{cases} \equiv \begin{cases} \min\limits_{x \in X} & \phi(x) \\ \text{s.t.} & \|x\|_{(k)} - \|x\|_1 = 0 \end{cases} \approx \min_{x \in X} f_1(x) - f_2(x)$$

$$f_1(x) = \phi(x) + \lambda \|x\|_{(k)} \quad \text{and} \quad f_2(x) = \lambda \|x\|_1$$

($\lambda > 0$ is a given parameter)

- Let $\{y^1, \ldots, y^m\}$, with $y^i \in \mathbb{R}^d$, $i = 1, \ldots, m$, be the data set to be grouped
- The goal is to partition the data set into k disjoint subsets, called clusters, such that a clustering criterion is optimized
- Each cluster must be in X
- Given a distance function $d(x, y)$ (e.g. $d(x, y) = \|x - y\|$), one tries to minimize the sum of the distance of each data point to the center $x^i \in \mathbb{R}^d$ of its cluster:

$$\min_{x^1,\ldots,x^k \in X} \sum_{i=1}^{m} \min_{j=1,\ldots,k} d(x^j, y^i)$$

- The objective function can be decomposed as $f_1(x) - f_2(x)$, with $x = (x^1, \ldots, x^k) \in \mathbb{R}^n$ the vector composed of all k centers, $n = kd$,

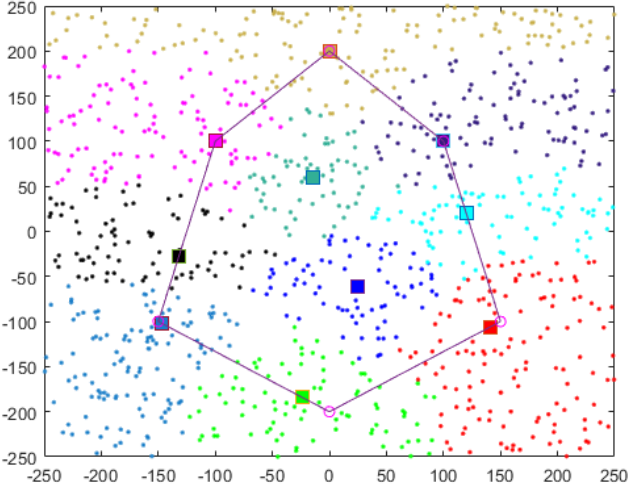
$$f_1(x) = \sum_{i=1}^{m} \sum_{j=1}^{k} d(x^j, y^i) \quad \text{and} \quad f_2(x) = \sum_{i=1}^{m} \max_{j=1,\ldots,k} \sum_{s=1, s\neq j}^{k} d(x^s, y^i)$$

- The clustering problem can thus be written as a DC program

$$\min_{x \in X} f_1(x) - f_2(x)$$

Constrained Clustering

$$\min_{x^1,\dots,x^k \in X} \sum_{i=1}^{m} \min_{j=1,\dots,k} d(x^j, y^i), \quad m = 1000,\ k = 10,\ d(x,y) = \|x-y\|_1$$



- Let $(\Xi, \mathcal{F}, \mathbb{P})$ be a probability space and $g : \mathbb{R}^n \times \Xi \to \mathbb{R}$ be convex in the first argument: $g(\cdot, \xi)$ is convex for any given $\xi \in \Xi$
- The problem of finding a point in the convex and compact set $X \subset \mathbb{R}^n$ such that the system of inequalities $g(x, \xi) \leq 0$ holds with the highest possible probability can be formulated as

$$\begin{aligned}\max_{x \in X} \mathbb{P}[g(x,\xi) \leq 0] \quad &\equiv \quad \max_{x \in X} [1 - \mathbb{P}[g(x,\xi) > 0]] \quad \equiv \quad 1 - \min_{x \in X} \mathbb{P}[g(x,\xi) > 0] \\ &\equiv \quad 1 - \min_{x \in X} \mathbb{E}[\chi(\max\{g(x,\xi), 0\}],\end{aligned}$$

where $\chi : \mathbb{R} \to \{0, 1\}$ be defined as $\chi(a) = 0$ if $a = 0$ and $\chi(a) = 1$ otherwise

- By approximating $\chi(\cdot)$ with $\min\left\{\frac{|\cdot|}{r}, 1\right\}$, and using finitely many scenarios $\{\xi^1, \cdots, \xi^S\}$ with associated probability $p_s > 0$, we get the following approximation

$$\min_{x \in X} \sum_{s=1}^{S} p_s \min\left\{\frac{\max\{g(x,\xi^s), 0\}}{r}, 1\right\}$$

- As we have already seen, this is a DC problem with available DC decompositions

CMA
Centre de Mathématiques Appliquées

What does it mean to solve a DC program?

$$\min_{x \in X} f(x), \quad \text{with} \quad f(x) = f_1(x) - f_2(x) \qquad (P)$$

Being a nonsmooth and nonconvex optimization problem, many definitions of stationarity exist for DC programs

A point $\bar{x} \in X$ is a

- global solution to problem (P) iff

$$\partial_\epsilon f_2(\bar{x}) \subset \partial_\epsilon [f_1(\bar{x}) + \mathbf{i}_X(\bar{x})] \quad \forall \epsilon \geq 0$$

- d(irectional)-stationary point to problem (P) if

$$\partial f_2(\bar{x}) \subset \partial [f_1(\bar{x}) + \mathbf{i}_X(\bar{x})]$$

- critical point to problem (P) if

$$\partial f_2(\bar{x}) \cap \partial [f_1(\bar{x}) + \mathbf{i}_X(\bar{x})] \neq \emptyset$$

Local algorithms for nonsmooth DC problems are only ensured to provide critical points (except when a special structure is assumed)

Centre de Mathématiques Appliquées

Criticality from the dual point of view

$$\min_{x \in X} f_1(x) - f_2(x) \qquad (P)$$

Theorem (Toland duality)

Let $\bar{f}_1(x) = f_1(x) + \mathbf{i}_X(x)$ *and* $\bar{f} = \bar{f}_1 - f_2$. *Then*

$$\bar{f}^*(g) = \sup_{s \in \mathbb{R}^n} \bar{f}_1^*(g+s) - f_2^*(s)$$

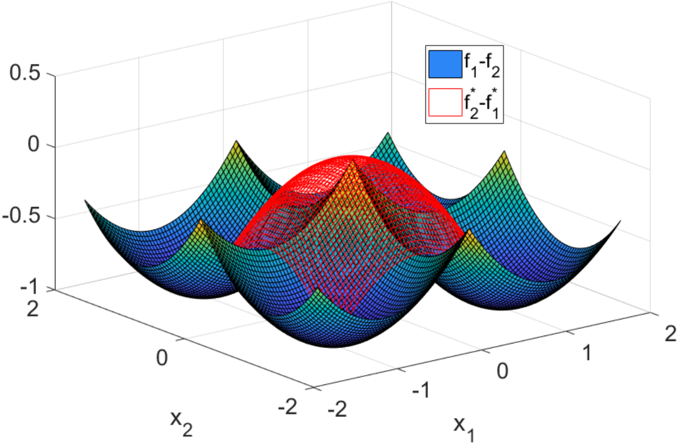
- Note that $f^*(0)$ is a DC problem itself: $\bar{f}^*(0) = \sup_{s \in \mathbb{R}^n} \bar{f}_1^*(s) - f_2^*(s)$
- With a little abuse of notation, we denote by *dual problem* the following one, with converse signal:

$$-\bar{f}^*(0) = \inf_{s \in \mathbb{R}^n} f_2^*(s) - \bar{f}_1^*(s) \qquad (D)$$

Theorem

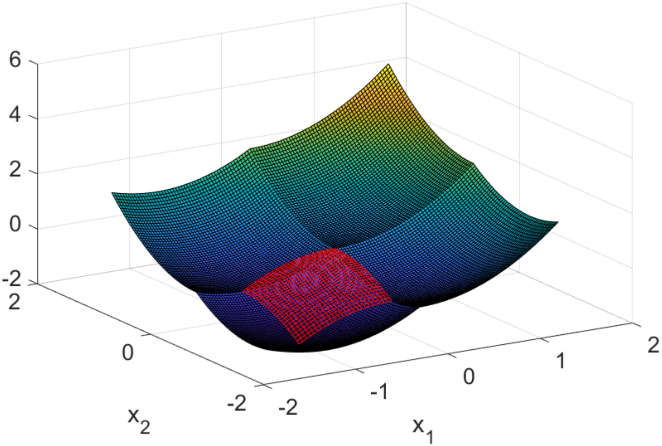
$$\bar{f}_1(\bar{x}) - f_2(\bar{x}) = f_2^*(s) - \bar{f}_1^*(s) \quad \textit{for all} \;\; s \in \partial f_2(\bar{x}) \cap \partial \bar{f}_1(\bar{x})$$

$$f(x) = \frac{1}{2}\|x\|^2 - \|x\|_1$$



The dual curves were obtained by plotting $f_2^*(\nabla f_1(x)) - f_1^*(\nabla f_1(x))$ with $x \in [-1,1] \times [-1,1]$. Critical points are the ones where f and f^* coincide

$$f(x) = (\|x\|^2 + \sum_{i=1}^{2} x_i) - \|x\|_1$$



The dual curves were obtained by plotting $f_2^*(\nabla f_1(x)) - f_1^*(\nabla f_1(x))$ with $x \in [-1, 1] \times [-1, 1]$. Critical points are the ones where f and f^* coincide

What does it mean to solve a DC program?

$$\min_{x \in X} f(x), \quad \text{with} \quad f(x) = f_1(x) - f_2(x) \qquad (P)$$

A point $\bar{x} \in X$ is a

- d(irectional)-stationary point to problem (P) if

$$\partial f_2(\bar{x}) \subset \partial[f_1(\bar{x}) + \mathbf{i}_X(\bar{x})]$$

  which is equivalent to say that $\bar{x}$ solves

$$\min_{x \in X} f_1(x) - [f_2(\bar{x}) + \langle g_2, x - \bar{x} \rangle] \quad \text{for all} \quad g_2 \in \partial f_2(\bar{x})$$

- critical point to problem (P) if

$$\partial f_2(\bar{x}) \cap \partial[f_1(\bar{x}) + \mathbf{i}_X(\bar{x})] \neq \emptyset$$

  which is equivalent to say that $\bar{x}$ solves

$$\min_{x \in X} f_1(x) - [f_2(\bar{x}) + \langle g_2, x - \bar{x} \rangle] \quad \text{for at least one} \quad g_2 \in \partial f_2(\bar{x})$$

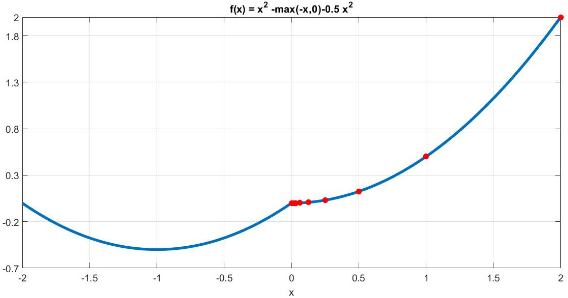
Note that the concepts of criticality and d-stationarity coincide if f_2 is differentiable at $\bar{x}$

DC ALGORITHM - DCA

FOR ALL $k = 1, 2, \ldots,$ COMPUTE $g_2^k \in \partial f_2(x^k)$ AND $x^{k+1} \in \arg\min_{x \in \mathbb{R}^n} f_1(x) - \langle g_2^k, x\rangle$

Consider the unidimensional problem

$$\min_x \; x^2 - (\max\{-x, 0\} + 0.5x^2)$$



If we start the iterative process with $x^0 > 0$, then the DCA[2] defines $x^{k+1} = x^k/2$

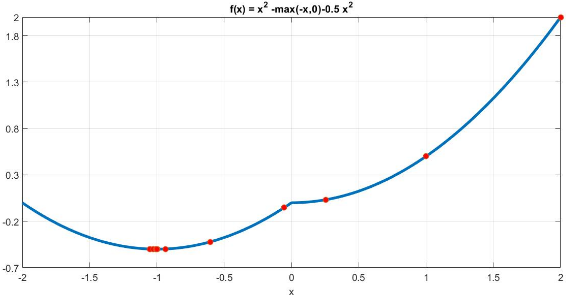
Hence, $x^k \to 0$ and $\bar{x} = 0$ is critical but not a d-stationary point: $\partial f_2(0) = [-1, 0]$ and $\nabla f_1(0) = 0$

[2] Tao, P.D., Le Thi, H.A.: Convex analysis approach to DC programming: theory, algorithms and applications. Acta Mathematica Vietnamica, 1997

$x^{k+1} \in \arg\min_{x \in \mathbb{R}^n} f_1(x) - \langle g_2^k + \beta(x^k - x^{k-1}), x\rangle, \qquad \beta = 0.49$

One manner to get points of better quality is to insert some inertial[3] to DCA

$$\min_x \; x^2 - (\max\{-x, 0\} + 0.5x^2)$$

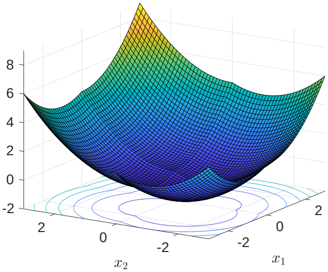


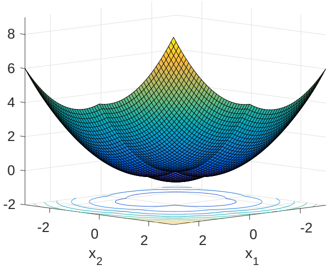
[3] W. de Oliveira and M. Tcheou. An inertial algorithm for DC programming. SVAA, 2019

A bi-dimensional example

$f(x) = f_1(x) - f_2(x)$

$f_1(x) = \|x\|^2$ and $f_2(x) = \max(-x_1, 0) + \max(-x_2, 0) + 0.5\|x\|^2$

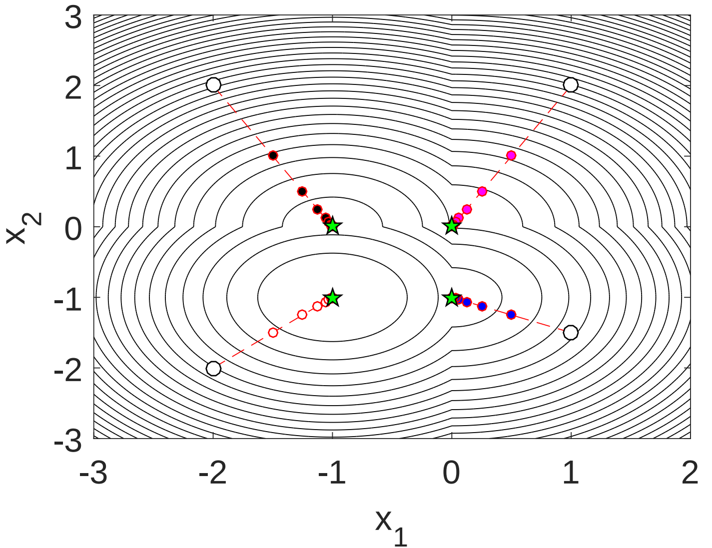




IDCA WITH $\beta = 0$ (DCA)

$x^{k+1} \in \arg\min_{x \in \mathbb{R}^n} f_1(x) - \langle g_2^k, x\rangle$

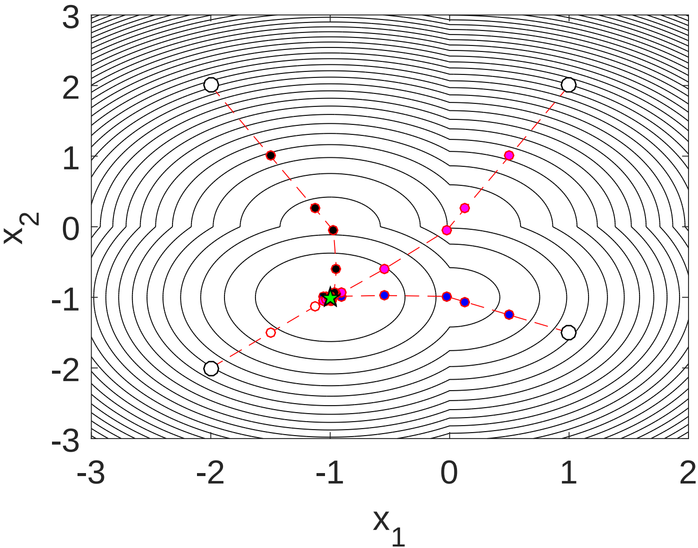
$f_1(x) = \|x\|^2$ and $f_2(x) = \max(-x_1, 0) + \max(-x_2, 0) + 0.5\|x\|^2$



IDCA WITH $\beta = 0.49$

$x^{k+1} \in \arg\min_{x\in\mathbb{R}^n} f_1(x) - \langle g_2^k + \beta(x^k - x^{k-1}), x\rangle$

$f_1(x) = \|x\|^2$ and $f_2(x) = \max(-x_1, 0) + \max(-x_2, 0) + 0.5\|x\|^2$



Sequential DC programming

- DCA has a successful history of more than 30 years: if one thinks of DC programming, one thinks of DCA
- The algorithm shows it full strength when the convex subproblem is simple

$$x^{k+1} \in \arg\min_{x \in X} \; f_1(x) - [f(x^k) + \langle g^k, x - x^k \rangle] \qquad (\texttt{Sbpm})$$

- By "simple subproblem" we mean that a solution x^{k+1} can be computed in algebraic or computationally cheap ways
- If **(Sbpm)** is difficult (e.g. f_1 is only via an oracle/black box), then DCA can be too time consuming depending on the application
- Furthermore, DCA does not treat f_1 and f_2 equally: f_2 is approximated by a single linearization, whereas f_1 is treated as is

Sequential DC programming - SDCP

$$\min_{x \in X} f(x), \quad \text{with} \quad f(x) = f_1(x) - f_2(x) \qquad (P)$$

- DCA approximates (P) iteratively with (possibly difficult) convex subproblems

DCA

for all $k = 0, 1, 2, \ldots$, compute $g_2^k \in \partial f_2(x^k)$ and let

x^{k+1} be a solution of $\min_{x \in X} \; f_1(x) - [f_2(x^k) + \langle g_2^k, x - x^k \rangle]$

- SDCP approximates (P) iteratively with easy DC subproblems

SDCP

for all $k = 0, 1, 2, \ldots$, update a convex model $\mathfrak{f}_{\mathcal{M}}^k$ and let

x^{k+1} be an approximate critical point of $\min_{x \in X} \; \mathfrak{f}_{\mathcal{M}}^k(x) - f_2(x)$

The idea is to choose $\mathfrak{f}_{\mathcal{M}}^k$ s.t. $\min_{x \in X} \; \mathfrak{f}_{\mathcal{M}}^k(x) - f_2(x)$ is simple

$$\min_{x \in X} f(x), \quad \text{with} \quad f(x) = f_1(x) - f_2(x) \qquad (P)$$

How to choose a convex model $\mathfrak{f}^k_{\mathcal{M}}$ for f_1 such that computing a critical point of $\min_{x \in X} \mathfrak{f}^k_{\mathcal{M}}(x) - f_2(x)$ is simple?

- We define a class of models satisfying

$$\mathfrak{f}^k_{\mathcal{M}}(x) := \mathfrak{f}^k_{\texttt{low}}(x) + \frac{1}{2}\langle M^k(x - x^k), (x - x^k)\rangle \quad \text{for all } x \in \mathbb{R}^n \text{ and } k = 0, 1, 2, \ldots$$

with $\mathfrak{f}^k_{\texttt{low}}(x)$ a lower model for f_1:

$$\mathfrak{f}^k_{\texttt{low}}(x) \le f_1(x) \qquad \text{for all } x \in \mathbb{R}^n$$

and $M^k \in \mathbb{R}^{n \times n}$ is a symmetric PSD matrix (e.g. $M^k = 0$)

- Depending on M^k, we can have lower models, upper models, and even second-order Taylor models

$$\mathfrak{f}^k_{\mathcal{M}}(x) = \mathfrak{f}^k_{\mathtt{low}}(x) + \frac{1}{2}\langle M^k(x - x^k), (x - x^k)\rangle$$

A lower model can be defined by setting

- $\mathfrak{f}^k_{\mathtt{low}}(x) = \check{f}^k_1(x)$ for all k, with

$$\check{f}^k_1(x) := \max_{j=0,1,\dots,k} \{f_1(x^j) + \langle g^j_1, x - x^j\rangle\} \leq f_1(x) \quad \text{for all } x \in \mathbb{R}^n$$

- $M^k = 0 \in \mathbb{R}^{n\times n}$ for all k

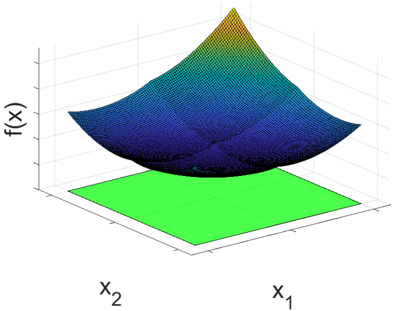
In this case, the SDCP reads as[4]

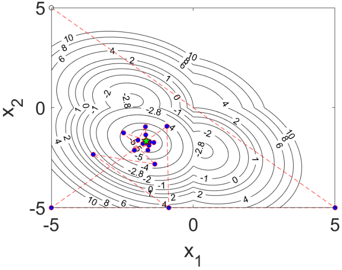
for all $k = 0, 1, 2, \dots$ let

x^{k+1} be an approximate critical point of $\min_{x\in X} \; \check{f}^k_1(x) - f_2(x)$

[4] If X is polyhedral, then the DCA applied to this subpoblem requires solving a LP per (inner) iteration

$f_1(x) = \sum_{i=1}^{2} \sqrt{1 + x_i^2} + \langle Ax, x \rangle$ (the matrix A is given by $A_{11} = A_{22} = 0.1$, $A_{12} = 0.3$ and $A_{21} = 0.2$), $f_2(x) = 5 \sum_{i=1}^{2} \max\{-x_i, 0\}$, and $X = \{-5 \le x_i \le 5,\ i = 1, 2\}$





Preliminary numerical experiments have shown that this SDCP variant

- almost always computes a global solution
- but is unstable and very slow...

Sequential DC programming with upper models

$$\mathfrak{f}^k_{\mathcal{M}}(x) = \mathfrak{f}^k_{\mathtt{low}}(x) + \frac{1}{2}\langle M^k(x-x^k),(x-x^k)\rangle$$

- Suppose that

$$f_1(x) = \psi(x) + \max_{i=1,\ldots,m} \phi_i(x)$$

  with $\psi : \mathbb{R}^n \to \mathbb{R}$ a simple convex function (piecewise linear or quadratic), and $\phi_i : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable convex functions having gradient $\nabla \phi_i$ Lipschitz continuous with modulus $L_{\phi_i} > 0$, $i = 1, \ldots, m$

- If a constant $L \geq L_{\phi_i}$, $i = 1, \ldots, m$, is known we may take
  - $\mathfrak{f}^k_{\mathtt{low}}(x) = \psi(x) + \max_{i=1,\ldots,m} \{\phi_i(x^k) + \langle \nabla \phi_i(x^k), x - x^k\rangle\}$ for all k
  - $M^k = (L+\gamma)I \in \mathbb{R}^{n\times n}$ for all k (for a $\gamma > 0$)

$$\boxed{f_1(x) \leq \mathfrak{f}^k_{\mathcal{M}}(x) \quad \text{for all } x}$$

Sequential DC programming with upper models

$$\mathfrak{f}^k_{\mathcal{M}}(x) = \mathfrak{f}^k_{\mathtt{low}}(x) + \frac{1}{2}\langle M^k(x-x^k), (x-x^k)\rangle$$

- Suppose that $f_1(x) = \sum_{i=1}^m \psi_i(x_i) + \phi(x)$ with $\psi_i : \mathbb{R}^{n_i} \to \mathbb{R}$ simple convex functions, and $\phi : \mathbb{R}^n \to \mathbb{R}$ having gradient $\nabla\phi$ Lipschitz continuous with modulus $L_\phi > 0$
- The feasible set is decomposable: $X = \Pi_i^m X_i$, with $X_i \subset \mathbb{R}^{n_i}$
- If a constant $L \geq L_\phi$, $i = 1, \ldots, m$, is known we may take
  - $\mathfrak{f}^k_{\mathtt{low}}(x) = \sum_{i=1}^m \psi_i(x_i) + \phi(x^k) + \langle \nabla\phi(x^k), x - x^k\rangle$ for all k
  - $M^k = (L+\gamma)I \in \mathbb{R}^{n\times n}$ for all k (for some $\gamma > 0$)

[5] Parallel computing can be employed to speed up calculations in SDCP (although f_1 is not decomposable)

Sequential DC programming with upper models

$$\mathfrak{f}^k_{\mathcal{M}}(x) = \mathfrak{f}^k_{\texttt{low}}(x) + \frac{1}{2}\langle M^k(x - x^k), (x - x^k)\rangle$$

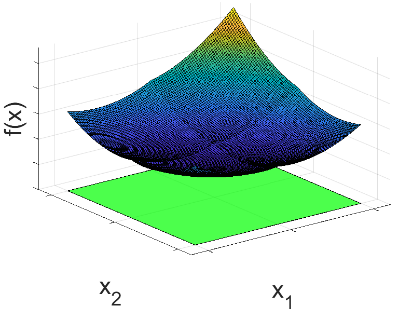
- Suppose that $f_1(x) = \sum_{i=1}^m \psi_i(x_i) + \phi(x)$ with $\psi_i : \mathbb{R}^{n_i} \to \mathbb{R}$ simple convex functions, and $\phi : \mathbb{R}^n \to \mathbb{R}$ having gradient $\nabla\phi$ Lipschitz continuous with modulus $L_\phi > 0$
- The feasible set is decomposable: $X = \Pi_i^m X_i$, with $X_i \subset \mathbb{R}^{n_i}$
- If a constant $L \geq L_\phi$, $i = 1, \ldots, m$, is known we may take
  - $\mathfrak{f}^k_{\texttt{low}}(x) = \sum_{i=1}^m \psi_i(x_i) + \phi(x^k) + \langle\nabla\phi(x^k), x - x^k\rangle$ for all k
  - $M^k = (L + \gamma)I \in \mathbb{R}^{n\times n}$ for all k (for some $\gamma > 0$)

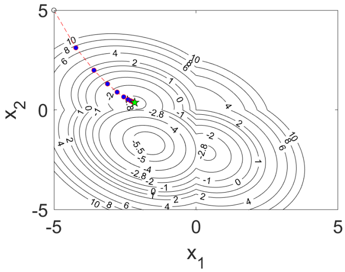
If $\min_{x\in X} \mathfrak{f}^k_{\mathcal{M}}(x) - f_2(x)$ is handled by DCA, then its convex subproblem $\min_{x\in X} \mathfrak{f}^k_{\mathcal{M}}(y) - \langle g_2^\ell, y\rangle$ can be decomposed[5]

$$y_i^{\ell+1} \in \arg\min_{y_i\in X_i} \psi_i(y_i) + \frac{L+\gamma}{2}\|y_i - {x^k}_i\|^2 + \langle\nabla_{x_i}\phi(x^k) - g_{2,i}^\ell, y_i\rangle, \quad i = 1, \ldots, m$$

[5] Parallel computing can be employed to speed up calculations in SDCP (although f_1 is not decomposable)

$f_1(x) = \sum_{i=1}^{2} \sqrt{1 + x_i^2} + \langle Ax, x \rangle$ (the matrix A is given by $A_{11} = A_{22} = 0.1$, $A_{12} = 0.3$ and $A_{21} = 0.2$), $f_2(x) = 5 \sum_{i=1}^{2} \max\{-x_i, 0\}$, and $X = \{-5 \le x_i \le 5,\ i = 1, 2\}$





Preliminary numerical experiments have shown that this SDCP variant

- is stable and reasonably fast
- solution quality depends strongly on initialization

Sequential DC programming with 2nd-order Taylor models

$$\min_{x \in X} f(x), \quad \text{with} \quad f(x) = f_1(x) - f_2(x) \qquad (P)$$

Suppose that (P) satisfies the following assumptions

$X = \mathbb{R}^n$, $f_1(x) = \phi(x) + \psi(x)$ with
$\psi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ a simple convex function (not necessarily smooth) and
$\phi : \mathbb{R}^n \to \mathbb{R}$ convex and twice-continuously differentiable

- ψ can be the indicator function of a closed convex set $C \subset \mathbb{R}^n$: $\psi(x) = \mathbf{i}_C(x)$
- The model $\mathfrak{f}^k_{\mathcal{M}}$ is the sum of ψ with the 2nd-order Taylor's expansion of ϕ

$$\mathfrak{f}^k_{\mathcal{M}}(x) := \underbrace{\psi(x) + \phi(x^k) + \langle \nabla \phi(x^k), x - x^k \rangle}_{\mathfrak{f}^k_{\mathtt{low}}(x)} + \frac{1}{2} \langle \nabla^2 \phi(x^k)(x - x^k), (x - x^k) \rangle$$

Centre de Mathématiques Appliquées

Sequential DC programming with 2nd-order Taylor models

$$\min_{x\in X} f(x), \quad \text{with} \quad f(x) = f_1(x) - f_2(x) \qquad (P)$$

With the model

$$\mathfrak{f}^k_{\mathcal{M}}(x) := \psi(x) + \phi(x^k) + \langle \nabla\phi(x^k), x - x^k\rangle + \frac{1}{2}\langle \nabla^2\phi(x^k)(x - x^k), (x - x^k)\rangle,$$

if x^{k+1} is a critical point for the DC subproblem

$$\min_{x\in X} \mathfrak{f}^k_{\mathcal{M}}(x) - f_2(x)$$

then

$$\emptyset \neq \partial\mathfrak{f}^k_{\mathcal{M}}(x^{k+1}) \cap \partial f_2(x^{k+1}) \quad \Rightarrow \quad 0 \in \partial\mathfrak{f}^k_{\mathcal{M}}(x^{k+1}) - \partial f_2(x^{k+1}),$$

i.e., x^{k+1} solves the generalized equation (GE)

$$0 \in \nabla\phi(x^k) + \nabla^2\phi(x^k)(x - x^k) + \partial\psi(x) - \partial f_2(x)$$

$$\boxed{\text{This is a Newton-type iteration!}}$$

The Josephy-Newton method

$$\min_{x \in X} f(x), \quad \text{with} \quad f(x) = f_1(x) - f_2(x) \qquad (P)$$

- If $\bar{x}$ is a critical point of (P), then $0 \in \nabla\phi(\bar{x}) + \partial\psi(\bar{x}) - \partial f_2(\bar{x})$
- In this case, computing a critical point of (P) is equivalent to

  find $\bar{x} \in \mathbb{R}^n$ such that $\quad 0 \in \Phi(\bar{x}) + \mathcal{N}(\bar{x}), \quad$ where $\begin{cases} \Phi(x) := \nabla\phi(x) \\ \mathcal{N}(x) := \partial\psi(x) - \partial f_2(x) \end{cases}$

- The Josephy-Newton method applied to the above GE works as follows: given x^k, the next iterate x^{k+1} is computed as a solution of the partially linearized GE

$$0 \in \Phi(x^k) + \nabla\Phi(x^k)(x - x^k) + \mathcal{N}(x)$$

  i.e.,

$$0 \in \nabla\phi(x^k) + \nabla^2\phi(x^k)(x - x^k) + \partial\psi(x) - \partial f_2(x)$$

Sequential DC programming with 2nd-order Taylor models

$$\min_{x \in X} f(x), \quad \text{with} \quad f(x) = f_1(x) - f_2(x) \qquad (P)$$

The SDCP algorithm

for all $k = 0, 1, 2, \dots$ let x^{k+1} be a critical point of

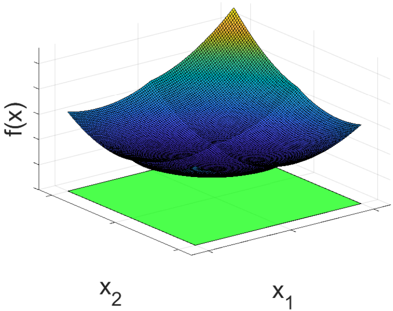
$$\min_{x \in X} \psi(x) + \phi(x^k) + \langle \nabla \phi(x^k), x - x^k \rangle + \frac{1}{2} \langle \nabla^2 \phi(x^k)(x - x^k), (x - x^k) \rangle - f_2(x)$$

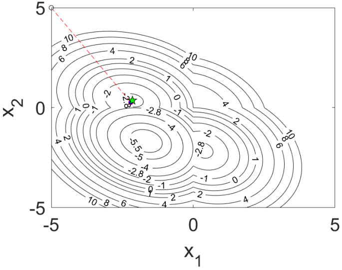
is an implementation of the Josephy-Newton method applied to the GE

$$\text{find } \bar{x} \in \mathbb{R}^n \text{ such that} \quad 0 \in \Phi(\bar{x}) + \mathcal{N}(\bar{x}), \quad \text{where} \quad \begin{cases} \Phi(x) = \nabla \phi(x) \\ \mathcal{N}(x) = \partial \psi(x) - \partial f_2(x) \end{cases}$$

SDCP converges quadratically under certain conditions

$f_1(x) = \sum_{i=1}^{2} \sqrt{1 + x_i^2} + \langle Ax, x \rangle$ (the matrix A is given by $A_{11} = A_{22} = 0.1$, $A_{12} = 0.3$ and $A_{21} = 0.2$), $f_2(x) = 5 \sum_{i=1}^{2} \max\{-x_i, 0\}$, and $X = \{-5 \leq x_i \leq 5,\ i = 1, 2\}$





Preliminary numerical experiments have shown that this SDCP variant

- is very fast
- converges if initialized near to a critical point

Concluding remarks

- The universality of DC functions implies that almost all real-life optimization problems can be recast into the DC programming framework
- However, a DC decomposition of the underlying function is not always available
- When a DC decomposition is accessible, it permits employing the convex analysis apparatus
- The availability of a DC decomposition does not imply by itself that DC tools should be employed in place of standard algorithms
- The use of DC algorithms should be supported by one of the following reasons: the absence of a nonconvex black-box for f but the availability of convex black-boxes for f_1 and f_2; nonsmoothess; the existence of analytic formulas or computationally cheap procedures for defining iterates; decomposability; modeling advantages; or others
- Without special structure, algorithms for DC programming can only compute critical points
- Critical points can be of poor quality. Ex: $\bar{x} = 0$ is critical for $\min_{x \geq -10} f(x)$, with $f(x) = \frac{|x|}{2} - \max\{-x, 0\}$

Concluding remarks

- The universality of DC functions implies that almost all real-life optimization problems can be recast into the DC programming framework
- However, a DC decomposition of the underlying function is not always available
- When a DC decomposition is accessible, it permits employing the convex analysis apparatus
- The availability of a DC decomposition does not imply by itself that DC tools should be employed in place of standard algorithms
- The use of DC algorithms should be supported by one of the following reasons: the absence of a nonconvex black-box for f but the availability of convex black-boxes for f_1 and f_2; nonsmoothess; the existence of analytic formulas or computationally cheap procedures for defining iterates; decomposability; modeling advantages; or others
- Without special structure, algorithms for DC programming can only compute critical points
- Critical points can be of poor quality. Ex: $\bar{x} = 0$ is critical for $\min_{x \geq -10} f(x)$, with $f(x) = \frac{|x|}{2} - \max\{-x, 0\} = x/2$!

Thank you!

References

- Le Thi, H.A., Tao, P.D. DC programming and DCA thirty years of developments. Math. Program. 169(1), 5-68 (2018)
- W. de Oliveira. Sequential Difference-of-Convex Programming. Journal of Optimization Theory and Applications, v.186 (3), pp. 936-959, 2020
- W. de Oliveira. The ABC of DC Programming. Set-Valued and Variational Analysis, v. 28, pp. 679-706, 2020

Contact:

welington.oliveira@mines-paristech.fr

www.oliveira.mat.br

We would like to acknowledge the financial support of PGMO through the project

Advanced nonsmooth optimization methods for stochastic programming