# The Boosted Difference of Convex functions Algorithm

**Phan Tu Vuong**
School of Mathematical Sciences
University of Southampton, UK
joint work with **Francisco J. Aragón Artacho**

**Dynamical systems and Semi-algebraic geometry: interactions with Optimization and Deep Learning**

**Da Lat University, July 17-21, 2023**

# The Boosted Difference of Convex functions Algorithm

**Phan Tu Vuong**
School of Mathematical Sciences
University of Southampton, UK

joint work with **Francisco J. Aragón Artacho**

**Dynamical systems and Semi-algebraic geometry: interactions with Optimization and Deep Learning**

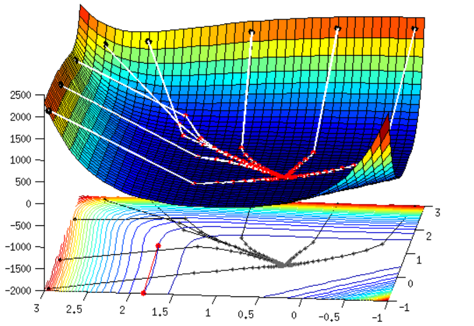
**Da Lat University, July 17-21, 2023**

## Outline

## Computational Optimisation

$$\underset{x \in C}{\text{minimise}} \quad \phi(x)$$

$$C \subset \mathbb{R}^m$$



- $$\underset{x \in \mathbb{R}^m}{\text{minimise}} \quad \phi(x) + \delta_C(x),$$

  where

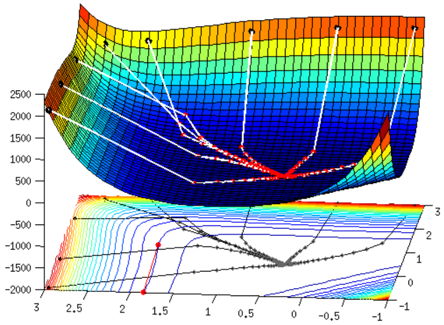
  $$\delta_C(x) = \begin{cases} 0 & x \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

- Efficient solvers for (strongly) convex cost

## Computational Optimisation

$$\underset{x \in C}{\text{minimise}} \quad \phi(x)$$

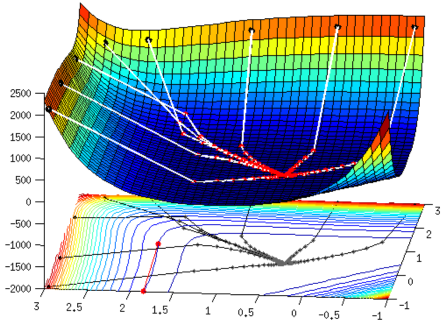
$$C \subset \mathbb{R}^m$$



## Computational Optimisation

$$\underset{x \in C}{\text{minimise}} \quad \phi(x)$$

$$C \subset \mathbb{R}^m$$



Algorithm Principles: Given $x_k$

1. Find a (good) descent direction $d_k$ at $x_k$ (expensive)

$$\phi'(x_k; d_k) < 0$$

2. Follow the descent direction as far as possible

## A nonconvex optimization problem

We will focus on the nonconvex optimization problem

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ g(x) - h(x) =: \phi(x),$$

where $g, h : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ are convex functions with

$$\inf_{x\in\mathbb{R}^m} \phi(x) > -\infty.$$

The objective function $\phi$ is a DC function, i.e., a difference of convex functions.

## A nonconvex optimization problem

We will focus on the nonconvex optimization problem

$$(\mathcal{P}) \ \underset{x \in \mathbb{R}^m}{\text{minimize}} \ \ g(x) - h(x) =: \phi(x),$$

where $g, h : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ are convex functions with

$$\inf_{x \in \mathbb{R}^m} \phi(x) > -\infty.$$

The objective function $\phi$ is a DC function, i.e., a difference of convex functions.

The following assumptions are made:

- WLOG $g$ and $h$ are strongly convex with modulus $\rho > 0$ (otherwise, take $\widetilde{g}(x) := g(x) + \frac{\rho}{2}\|x\|^2$ and $\widetilde{h}(x) := h(x) + \frac{\rho}{2}\|x\|^2$).

## A nonconvex optimization problem

We will focus on the nonconvex optimization problem

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ g(x) - h(x) =: \phi(x),$$

where $g, h : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ are convex functions with

$$\inf_{x\in\mathbb{R}^m} \phi(x) > -\infty.$$

The objective function $\phi$ is a DC function, i.e., a difference of convex functions.

The following assumptions are made:

- WLOG $g$ and $h$ are strongly convex with modulus $\rho > 0$ (otherwise, take $\widetilde{g}(x) := g(x) + \frac{\rho}{2}\|x\|^2$ and $\widetilde{h}(x) := h(x) + \frac{\rho}{2}\|x\|^2$).
- $g$ is continuously differentiable on an open set containing $\operatorname{dom} h$.

## A nonconvex optimization problem

We will focus on the nonconvex optimization problem

$$(\mathcal{P})\ \underset{x\in\mathbb{R}^m}{\text{minimize}}\ \ g(x) - h(x) =: \phi(x),$$

where $g, h : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ are convex functions with

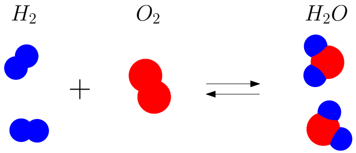
$$\inf_{x\in\mathbb{R}^m} \phi(x) > -\infty.$$

The objective function $\phi$ is a DC function, i.e., a difference of convex functions.

The following assumptions are made:

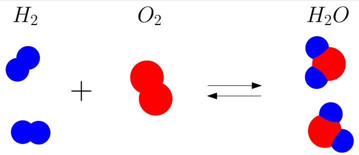
- WLOG $g$ and $h$ are strongly convex with modulus $\rho > 0$ (otherwise, take $\widetilde{g}(x) := g(x) + \frac{\rho}{2}\|x\|^2$ and $\widetilde{h}(x) := h(x) + \frac{\rho}{2}\|x\|^2$).
- $g$ is continuously differentiable on an open set containing $\operatorname{dom} h$.
- $h$ is subdifferentiable at every point in $\operatorname{dom} h$; i.e., $\partial h(x) \neq \emptyset$ for all $x \in \operatorname{dom} h$.

## An example in Biochemistry



$$2\underbrace{H_2}_{c_1} + \underbrace{O_2}_{c_2} \rightleftharpoons 2\underbrace{H_2O}_{c_3}$$

## An example in Biochemistry



$$2\ H_2 + O_2 \rightleftharpoons 2\,H_2O$$

- $(c_1, c_2, c_3) \in \mathbb{R}^3_{>0}$ denotes molecular species concentrations.
- The net reaction rate quantifies the rate of a chemical reaction:

$$\text{net reaction rate} = k_f c_1^2 c_2 - k_r c_3^2$$

$k_f, k_r \in \mathbb{R}_{\geq 0}$ are the kinetic parameters.

- The stoichiometric matrices are defined as

$$F := \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} \quad \text{and} \quad R := \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

- The dynamical equation for time evolution of molecular species is

$$\frac{dc}{dt} = (R - F)(k_f c_1^2 c_2 - k_r c_3^2)$$

## A function arising from a biochemical network

Consider a biochemical network with:

- $m$ molecular species, $n$ reversible elementary reactions;
- $F, R \in \mathbb{Z}^{m\times n}_{\geq 0}$ denote the forward and reverse stoichiometric matrices;
- We assume constant non-negative elementary kinetic parameters $k_f, k_r \in \mathbb{R}^n_{\geq 0}$;
- $c \in \mathbb{R}^m_{>0}$ is the vector of molecular species concentrations;
- The dynamical equation of molecular species concentration is

$$\frac{dc}{dt} = (R - F)\big(\exp(\ln(k_f) + F^T \ln(c)) - \exp(\ln(k_r) + R^T \ln(c))\big);$$

- If we transform the right-hand side into logarithmic scale, we get

$$f(x) := [F, R]\exp(p + [F, R]^T x) - [R, F]\exp(p + [F, R]^T x),$$

where $x := \ln(c), p := [\ln(k_f)^T, \ln(k_r)^T]^T$ and $[\cdot, \cdot]$ is the horizontal concatenation operator.

## A simple biochemical network

$$\begin{tikzcd}
& A \arrow[dl, shift right] \arrow[dr, shift left] & \\
B \arrow[ur, shift right] \arrow[rr, shift left] & & C \arrow[ul, shift left] \arrow[ll, shift left]
\end{tikzcd}$$

$$f(x) := ([F, R] - [R, F]) \exp([F, R]^T x)$$

## A simple biochemical network

$$\begin{tikzcd}
& A \arrow[dl, shift right] \arrow[dr, shift left] & \\
B \arrow[ur, shift right] \arrow[rr, shift left] & & C \arrow[ul, shift left] \arrow[ll, shift left]
\end{tikzcd}$$

$$f(x) := ([F,R]-[R,F])\exp([F,R]^T x)$$

$$F := \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad R := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

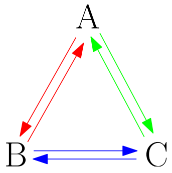
## A simple biochemical network

$$\begin{tikzcd}
 & A \arrow[dl, shift right] \arrow[dr, shift left] & \\
B \arrow[ur, shift right] \arrow[rr, shift left] & & C \arrow[ul, shift left] \arrow[ll, shift left]
\end{tikzcd}$$

$$f(x) := ([F,R]-[R,F])\exp([F,R]^T x)$$

$$F := \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad R := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

## A simple biochemical network



$$f(x) := ([F,R] - [R,F]) \exp([F,R]^T x)$$

$$F := \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad R := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

Thus, for any $x := (x_1, x_2, x_3)^T \in \mathbb{R}^3$ we have

$$f(x) = \begin{bmatrix} 2e^{x_1} - e^{x_2} - e^{x_3} \\ -e^{x_1} + 2e^{x_2} - e^{x_3} \\ -e^{x_1} - e^{x_2} + 2e^{x_3} \end{bmatrix}.$$

## A DC problem in biochemistry

We are interested in finding zeroes of

$$f(x) := [F,R]\exp(p+[F,R]^T x) - [R,F]\exp(p+[F,R]^T x).$$

Setting

$$p(x) := [F,R]\exp(p+[F,R]^T x) \quad \text{and} \quad c(x) := [R,F]\exp(p+[F,R]^T x)$$

we have an equivalent system of equations

$$p(x) = c(x) \quad x \in \mathbb{R}^m \tag{1}$$

- Solving (1) is equivalent to minimising the function

$$\|p(x)-c(x)\|^2 = 2\left(\|p(x)\|^2 + \|c(x)\|^2\right) - \|p(x)+c(x)\|^2$$

## A DC problem in biochemistry

We are interested in finding zeroes of

$$f(x) := [F,R]\exp(p+[F,R]^T x) - [R,F]\exp(p+[F,R]^T x).$$

Setting

$$p(x) := [F,R]\exp(p+[F,R]^T x) \quad \text{and} \quad c(x) := [R,F]\exp(p+[F,R]^T x)$$

we have an equivalent system of equations

$$p(x) = c(x) \quad x \in \mathbb{R}^m \tag{1}$$

- Solving (1) is equivalent to minimising the function

$$\|p(x)-c(x)\|^2 = 2\left(\|p(x)\|^2 + \|c(x)\|^2\right) - \|p(x)+c(x)\|^2$$

- All the components of $p(x)$ and $c(x)$ are nonnegative convex functions. Hence, $g(x) := 2\left(\|p(x)\|^2 + \|c(x)\|^2\right)$ and $h(x) := \|p(x)+c(x)\|^2$ are nonnegative convex functions.

## First-order necessary optimality condition and critical points

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \phi(x) := g(x) - h(x),$$ with $g$ smooth and $h$ convex

**Fact (First-order necessary optimality condition)**

*If $x^* \in dom\,\phi$ is an optimal solution of $(\mathcal{P}) \Rightarrow \partial h(x^*) = \{\nabla g(x^*)\}$.*

## First-order necessary optimality condition and critical points

$(\mathcal{P})\ \underset{x\in\mathbb{R}^m}{\text{minimize}}\ \ \phi(x) := g(x) - h(x),$ with $g$ smooth and $h$ convex

**Fact (First-order necessary optimality condition)**

*If* $x^* \in dom\,\phi$ *is an optimal solution of* $(\mathcal{P}) \Rightarrow \partial h(x^*) = \{\nabla g(x^*)\}$ .

**Definition**

We say that $\bar{x}$ is a critical point of $(\mathcal{P})$ if $\nabla g(\bar{x}) \in \partial h(\bar{x})$.

## First-order necessary optimality condition and critical points

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x), \text{ with } g \text{ smooth and } h \text{ convex}$$

**Fact (First-order necessary optimality condition)**

*If $x^* \in dom\,\phi$ is an optimal solution of $(\mathcal{P}) \Rightarrow \partial h(x^*) = \{\nabla g(x^*)\}$.*

**Definition**

We say that $\bar{x}$ is a critical point of $(\mathcal{P})$ if $\nabla g(\bar{x}) \in \partial h(\bar{x})$.

**Example**

Consider the DC function $\phi : \mathbb{R}^m \to \mathbb{R}$ defined for $x \in \mathbb{R}^m$ by

$$\phi(x) := \left(\|x\|^2 + \sum_{i=1}^{m} x_i\right) - \left(\sum_{i=1}^{m} |x_i|\right).$$

## First-order necessary optimality condition and critical points

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \phi(x) := g(x) - h(x),$$ with $g$ smooth and $h$ convex

**Fact (First-order necessary optimality condition)**

*If $x^* \in dom\, \phi$ is an optimal solution of $(\mathcal{P}) \Rightarrow \partial h(x^*) = \{\nabla g(x^*)\}$.*

**Definition**

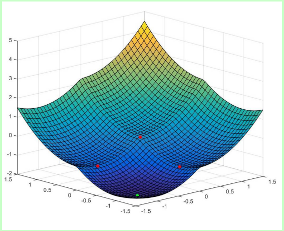
We say that $\bar{x}$ is a critical point of $(\mathcal{P})$ if $\nabla g(\bar{x}) \in \partial h(\bar{x})$.

**Example**

Consider the DC function $\phi : \mathbb{R}^m \to \mathbb{R}$ defined for $x \in \mathbb{R}^m$ by

$$\phi(x) := \left(\|x\|^2 + \sum_{i=1}^{m} x_i\right) - \left(\sum_{i=1}^{m} |x_i|\right).$$

Then, $\phi$ has $2^m$ critical points (any $x \in \{-1,0\}^m$), and only one point $x^* := (-1,\ldots,-1)$ satisfying $\partial h(x^*) = \{\nabla g(x^*)\}$, which is the global minimum of $\phi$.



## Previous works: linearizing the nonconvex part

### Fukushima–Mine'81

$$(\mathcal{P})\ \underset{x\in\mathbb{R}^m}{\text{minimize}}\ \ \phi(x) := g(x) - h(x)$$

- In 1981 Fukushima and Mine introduced two algorithms to minimize a composite function $g - h$, where $g$ is (strictly) convex (*possibly nonsmooth*) and $h$ is smooth (*possibly nonconvex*).

## Previous works: linearizing the nonconvex part

### Fukushima–Mine'81

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

- In 1981 Fukushima and Mine introduced two algorithms to minimize a composite function $g - h$, where $g$ is (strictly) convex (*possibly nonsmooth*) and $h$ is smooth (*possibly nonconvex*).
- If $x^\star$ is a local minimum $\Rightarrow \nabla h(x^\star) \in \partial g(x^\star)$ (critical point).

## Previous works: linearizing the nonconvex part

Fukushima–Mine'81

$$(\mathcal{P})\ \underset{x\in\mathbb{R}^m}{\text{minimize}}\ \ \phi(x) := g(x) - h(x)$$

- In 1981 Fukushima and Mine introduced two algorithms to minimize a composite function $g-h$, where $g$ is (strictly) convex (*possibly nonsmooth*) and $h$ is smooth (*possibly nonconvex*).
- If $x^\star$ is a local minimum $\Rightarrow \nabla h(x^\star) \in \partial g(x^\star)$ (critical point).

**ALGORITHM 1 (FM'81):** Fix some parameters $\alpha > 0$ and $0 < \beta < 1$. Let $x_0$ be any initial point and set $k := 0$.

1. Find the solution $y_k$ of

$$(\mathcal{P}_k)\ \underset{y\in\mathbb{R}^m}{\text{minimize}}\ \ g(y) - \langle \nabla h(x_k), y\rangle.$$

   and set $d_k := y_k - x_k$. If $d_k = 0 \Rightarrow$ **stop** and **return** $x_k$.

## Previous works: linearizing the nonconvex part

Fukushima–Mine'81

$$(\mathcal{P})\ \underset{x\in\mathbb{R}^m}{\text{minimize}}\ \ \phi(x) := g(x) - h(x)$$

- In 1981 Fukushima and Mine introduced two algorithms to minimize a composite function $g - h$, where $g$ is (strictly) convex (*possibly nonsmooth*) and $h$ is smooth (*possibly nonconvex*).
- If $x^\star$ is a local minimum $\Rightarrow \nabla h(x^\star) \in \partial g(x^\star)$ (critical point).

**ALGORITHM 1 (FM'81):** Fix some parameters $\alpha > 0$ and $0 < \beta < 1$. Let $x_0$ be any initial point and set $k := 0$.

1. Find the solution $y_k$ of

$$(\mathcal{P}_k)\ \underset{y\in\mathbb{R}^m}{\text{minimize}}\ \ g(y) - \langle \nabla h(x_k), y\rangle.$$

   and set $d_k := y_k - x_k$. If $d_k = 0 \Rightarrow$ **stop** and **return** $x_k$.

2. (Armijo - backtracking) Set $\lambda_k := 1$.
   **while** $\phi(x_k + \lambda_k d_k) > \phi(x_k) - \alpha\lambda_k\|d_k\|^2$ **do** $\lambda_k := \beta\lambda_k$.

## Previous works: linearizing the nonconvex part

Fukushima–Mine'81

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

- In 1981 Fukushima and Mine introduced two algorithms to minimize a composite function $g - h$, where $g$ is (strictly) convex (*possibly nonsmooth*) and $h$ is smooth (*possibly nonconvex*).
- If $x^\star$ is a local minimum $\Rightarrow \nabla h(x^\star) \in \partial g(x^\star)$ (critical point).

**ALGORITHM 1 (FM'81):** Fix some parameters $\alpha > 0$ and $0 < \beta < 1$. Let $x_0$ be any initial point and set $k := 0$.

1. Find the solution $y_k$ of
$$(\mathcal{P}_k) \ \underset{y\in\mathbb{R}^m}{\text{minimize}} \ \ g(y) - \langle \nabla h(x_k), y\rangle.$$
and set $d_k := y_k - x_k$. If $d_k = 0 \Rightarrow$ **stop** and **return** $x_k$.
2. (Armijo - backtracking) Set $\lambda_k := 1$.
**while** $\phi(x_k + \lambda_k d_k) > \phi(x_k) - \alpha\lambda_k \|d_k\|^2$ **do** $\lambda_k := \beta\lambda_k$.
3. Set $x_{k+1} := x_k + \lambda_k d_k$, $k := k+1$ and **go to** Step 1.

## Previous works: linearizing the nonconvex part

Le Thi–Pham Dinh–El Bernoussi'86: DC algorithm

$$(\mathcal{P}) \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \phi(x) := g(x) - h(x)$$

- In 1986 Pham Dinh and Souad introduced an algorithm to minimize a DC functions $g - h$, where $g$ and $h$ are both convex (*possibly nonsmooth*). This was further developed by Pham Dinh, Le Thi and their collaborators.

## Previous works: linearizing the nonconvex part

Le Thi–Pham Dinh–El Bernoussi'86: DC algorithm

$$(\mathcal{P}) \underset{x\in\mathbb{R}^m}{\text{minimize}} \;\; \phi(x) := g(x) - h(x)$$

- In 1986 Pham Dinh and Souad introduced an algorithm to minimize a DC functions $g - h$, where $g$ and $h$ are both convex (*possibly nonsmooth*). This was further developed by Pham Dinh, Le Thi and their collaborators.
- If $x^\star$ is a local minimum $\Rightarrow \partial h(x^\star) \subset \partial g(x^\star)$

## Previous works: linearizing the nonconvex part

Le Thi–Pham Dinh–El Bernoussi'86: DC algorithm

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

- In 1986 Pham Dinh and Souad introduced an algorithm to minimize a DC functions $g - h$, where $g$ and $h$ are both convex (*possibly nonsmooth*). This was further developed by Pham Dinh, Le Thi and their collaborators.
- If $x^\star$ is a local minimum $\Rightarrow \partial h(x^\star) \subset \partial g(x^\star) \Rightarrow \partial h(x^\star) \cap \partial g(x^\star) \neq \emptyset$.

## Previous works: linearizing the nonconvex part

Le Thi–Pham Dinh–El Bernoussi'86: DC algorithm

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

- In 1986 Pham Dinh and Souad introduced an algorithm to minimize a DC functions $g-h$, where $g$ and $h$ are both convex (*possibly nonsmooth*). This was further developed by Pham Dinh, Le Thi and their collaborators.
- If $x^\star$ is a local minimum $\Rightarrow \partial h(x^\star) \subset \partial g(x^\star) \Rightarrow \partial h(x^\star) \cap \partial g(x^\star) \neq \emptyset$.

**ALGORITHM 2 (DCA):** Let $x_0$ be any initial point and set $k := 0$.

1. Choose $u_k \in \partial h(x_k)$
2. Find a solution $y_k$ of
$$(\mathcal{P}_k) \ \underset{y\in\mathbb{R}^m}{\text{minimize}} \ \ g(y) - \langle y, u_k\rangle.$$
3. If $y_k = x_k \Rightarrow$ **stop**. Otherwise, set $x_{k+1} := y_k$, $k := k+1$ and **go to** Step 1.

## Previous works: linearizing the nonconvex part

Le Thi–Pham Dinh–El Bernoussi'86: DC algorithm

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

- In 1986 Pham Dinh and Souad introduced an algorithm to minimize a DC functions $g-h$, where $g$ and $h$ are both convex (*possibly nonsmooth*). This was further developed by Pham Dinh, Le Thi and their collaborators.
- If $x^\star$ is a local minimum $\Rightarrow \partial h(x^\star) \subset \partial g(x^\star) \Rightarrow \partial h(x^\star) \cap \partial g(x^\star) \neq \emptyset$.

**ALGORITHM 2 (DCA):** Let $x_0$ be any initial point and set $k := 0$.

1. Choose $u_k \in \partial h(x_k) \Leftrightarrow x_k \in (\partial h)^{-1}(u_k) = \partial h^*(u_k) \Leftrightarrow u_k$ is a solution of
$$(\mathcal{D}_k) \ \underset{u\in\mathbb{R}^m}{\text{minimize}} \ \ h^*(u) - \langle x_k, u\rangle.$$
2. Choose $y_k \in \partial g^*(u_k) \Leftrightarrow u_k \in \partial g(y_k) \Leftrightarrow y_k$ is a solution of
$$(\mathcal{P}_k) \ \underset{y\in\mathbb{R}^m}{\text{minimize}} \ \ g(y) - \langle y, u_k\rangle.$$
3. If $y_k = x_k \Rightarrow$ **stop**. Otherwise, set $x_{k+1} := y_k$, $k := k+1$ and **go to** Step 1.

## FM'81 and DCA when $h$ is smooth?

**ALGORITHM 2 (DCA):** Let $x_0$ be any initial point and set $k := 0$.

1. Find the solution $y_k$ of
$$(\mathcal{P}_k)\ \underset{y\in\mathbb{R}^m}{\text{minimize}}\ \ g(y) - \langle \nabla h(x_k), y\rangle.$$
2. If $y_k = x_k \Rightarrow$ **stop**. Otherwise, set $x_{k+1} := y_k$, $k := k+1$ and **go to** Step 1.

## FM'81 and DCA when $h$ is smooth?

```
ALGORITHM 2 (DCA): Let $x_0$ be any initial point and set $k := 0$.
  1. Find the solution $y_k$ of
         $(\mathcal{P}_k)\ \underset{y\in\mathbb{R}^m}{\text{minimize}}\ g(y) - \langle \nabla h(x_k), y\rangle.$
  2. If $y_k = x_k \Rightarrow$ stop. Otherwise, set $x_{k+1} := y_k$, $k := k+1$ and go to Step 1.
```

```
ALGORITHM 1 (FM'81): Fix some parameters $\alpha > 0$ and $0 < \beta < 1$.
Let $x_0$ be any initial point and set $k := 0$.
  1. Find the solution $y_k$ of
         $(\mathcal{P}_k)\ \underset{y\in\mathbb{R}^m}{\text{minimize}}\ g(y) - \langle \nabla h(x_k), y\rangle.$
     and set $d_k := y_k - x_k$. If $d_k = 0 \Rightarrow$ stop and return $x_k$.
  2. (Armijo - backtracking) Set $\lambda_k := 1$.
     while $\phi(x_k + \lambda_k d_k) > \phi(x_k) - \alpha\lambda_k \|d_k\|^2$ do $\lambda_k := \beta\lambda_k$.
  3. Set $x_{k+1} := x_k + \lambda_k d_k = \lambda y_k + (1-\lambda)x_k$, $k := k+1$ and go to Step 1.
```

## FM’81 and DCA when $h$ is smooth?

**ALGORITHM 2 (DCA):** Let $x_0$ be any initial point and set $k := 0$.

```
1. Find the solution $y_k$ of
       $(\mathcal{P}_k)\ \underset{y\in\mathbb{R}^m}{\text{minimize}}\ \ g(y) - \langle \nabla h(x_k), y\rangle.$
2. If $y_k = x_k \Rightarrow$ stop. Otherwise, set $x_{k+1} := y_k$, $k := k+1$ and go to Step 1.
```

**ALGORITHM 1 (FM’81):** Fix some parameters $\alpha > 0$ and $0 < \beta < 1$.
Let $x_0$ be any initial point and set $k := 0$.

```
1. Find the solution $y_k$ of
       $(\mathcal{P}_k)\ \underset{y\in\mathbb{R}^m}{\text{minimize}}\ \ g(y) - \langle \nabla h(x_k), y\rangle.$
   and set $d_k := y_k - x_k$. If $d_k = 0 \Rightarrow$ stop and return $x_k$.
2. (Armijo - backtracking) Set $\lambda_k := 1$.
   while $\phi(x_k + \lambda_k d_k) > \phi(x_k) - \alpha\lambda_k \|d_k\|^2$ do $\lambda_k := \beta\lambda_k$.
3. Set $x_{k+1} := x_k + \lambda_k d_k = \lambda y_k + (1-\lambda)x_k$, $k := k+1$ and go to Step 1.
```

**Proposition**

*If $g$ and $h$ are strongly convex with constant $\rho > 0$, then*

$$\phi(y_k) \le \phi(x_k) - \rho\|d_k\|^2 \quad \forall k \in \mathbb{N}.$$

## FM'81 and DCA when $h$ is smooth?

**ALGORITHM 2 (DCA):** Let $x_0$ be any initial point and set $k := 0$.

1. Find the solution $y_k$ of
$$(\mathcal{P}_k) \ \underset{y\in\mathbb{R}^m}{\text{minimize}} \ \ g(y) - \langle \nabla h(x_k), y\rangle.$$
2. If $y_k = x_k \Rightarrow$ **stop**. Otherwise, set $x_{k+1} := y_k$, $k := k+1$ and **go to** Step 1.

**ALGORITHM 1 (FM'81):** Fix some parameters $\alpha > 0$ and $0 < \beta < 1$.
Let $x_0$ be any initial point and set $k := 0$.

1. Find the solution $y_k$ of
$$(\mathcal{P}_k) \ \underset{y\in\mathbb{R}^m}{\text{minimize}} \ \ g(y) - \langle \nabla h(x_k), y\rangle.$$
and set $d_k := y_k - x_k$. If $d_k = 0 \Rightarrow$ **stop** and **return** $x_k$.
2. (Armijo - backtracking) Set $\lambda_k := 1$.
**while** $\phi(x_k + \lambda_k d_k) > \phi(x_k) - \alpha\lambda_k \|d_k\|^2$ **do** $\lambda_k := \beta\lambda_k$.
3. Set $x_{k+1} := x_k + \lambda_k d_k = \lambda y_k + (1-\lambda)x_k$, $k := k+1$ and **go to** Step 1.

**Proposition**

*If $g$ and $h$ are strongly convex with constant $\rho > 0$, then*
$$\phi(y_k) \leq \phi(x_k) - \rho\|d_k\|^2 \quad \forall k \in \mathbb{N}.$$

$\Rightarrow$ If $0 < \alpha \leq \rho$, the iterations of FM'81 and DCA coincide.

## FM'81 and DCA

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \phi(x) := g(x) - h(x)$$

$$u_k \in \partial h(x_k), y_k \in \partial g^*(u_k)$$

- FM'81 is based on the fact that $d_k := y_k - x_k$ is a descent direction at $x_k$: it holds $\phi'(x_k; d_k) \leq -\rho\|d_k\|^2$.
- DCA works thanks to

$$\phi(y_k) = (g-h)(y_k) \leq (h^* - g^*)(u_k) - \frac{\rho}{2}\|d_k\|^2 \leq \phi(x_k) - \rho\|d_k\|^2.$$

## FM'81 and DCA

$$(\mathcal{P})\ \underset{x\in\mathbb{R}^m}{\text{minimize}}\ \ \phi(x) := g(x) - h(x)$$

$$u_k \in \partial h(x_k), y_k \in \partial g^*(u_k)$$

- FM'81 is based on the fact that $d_k := y_k - x_k$ is a descent direction at $x_k$: it holds $\phi'(x_k; d_k) \leq -\rho\|d_k\|^2$.
- DCA works thanks to

$$\phi(y_k) = (g-h)(y_k) \leq (h^* - g^*)(u_k) - \frac{\rho}{2}\|d_k\|^2 \leq \phi(x_k) - \rho\|d_k\|^2.$$

Advantages: Simplicity, works well in practice, does not require any line search.

## FM'81 and DCA

$$(\mathcal{P}) \underset{x\in\mathbb{R}^m}{\text{minimize}} \;\; \phi(x) := g(x) - h(x)$$

$$u_k \in \partial h(x_k), y_k \in \partial g^*(u_k)$$

- FM'81 is based on the fact that $d_k := y_k - x_k$ is a descent direction at $x_k$: it holds $\phi'(x_k; d_k) \leq -\rho\|d_k\|^2$.
- DCA works thanks to

$$\phi(y_k) = (g-h)(y_k) \leq (h^* - g^*)(u_k) - \frac{\rho}{2}\|d_k\|^2 \leq \phi(x_k) - \rho\|d_k\|^2.$$

Advantages: Simplicity, works well in practice, does not require any line search.
Drawbacks: It can be very slow.

## FM'81 and DCA

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

$$u_k \in \partial h(x_k), y_k \in \partial g^*(u_k)$$

- FM'81 is based on the fact that $d_k := y_k - x_k$ is a descent direction at $x_k$: it holds $\phi'(x_k; d_k) \leq -\rho\|d_k\|^2$.
- DCA works thanks to

$$\phi(y_k) = (g-h)(y_k) \leq (h^* - g^*)(u_k) - \frac{\rho}{2}\|d_k\|^2 \leq \phi(x_k) - \rho\|d_k\|^2.$$

Advantages: Simplicity, works well in practice, does not require any line search.
Drawbacks: It can be very slow. Can it be accelerated?

## FM'81 and DCA

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

$$u_k \in \partial h(x_k), y_k \in \partial g^*(u_k)$$

- FM'81 is based on the fact that $d_k := y_k - x_k$ is a descent direction at $x_k$: it holds $\phi'(x_k; d_k) \leq -\rho\|d_k\|^2$.
- DCA works thanks to

$$\phi(y_k) = (g-h)(y_k) \leq (h^* - g^*)(u_k) - \frac{\rho}{2}\|d_k\|^2 \leq \phi(x_k) - \rho\|d_k\|^2.$$

Advantages: Simplicity, works well in practice, does not require any line search.
Drawbacks: It can be very slow. Can it be accelerated?
Yes, if $g$ is smooth, thanks to the fact that

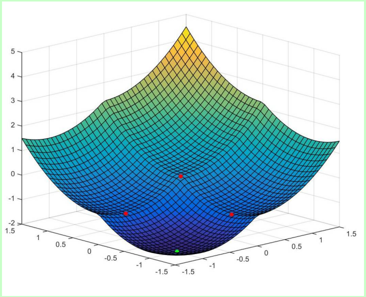
$$\boxed{\phi'(y_k; d_k) \leq -\rho\|d_k\|^2.}$$

## DCA can be slow and get easily trapped by critical points

### Example (Revisited)

Consider the DC function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined as $\phi := g - h$, where

$$g(x) = \frac{3}{2}\left(x_1^2 + x_2^2\right) + x_1 + x_2 \quad \text{and} \quad h(x) = |x_1| + |x_2| + \frac{1}{2}\left(x_1^2 + x_2^2\right).$$

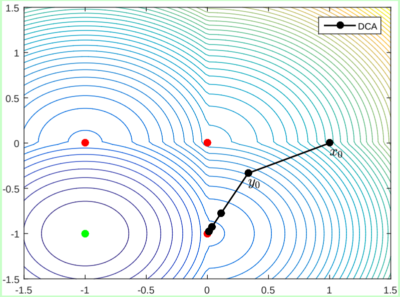


## DCA can be slow and get easily trapped by critical points

### Example (Revisited)

Consider the DC function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined as $\phi := g - h$, where

$$g(x) = \frac{3}{2}\left(x_1^2 + x_2^2\right) + x_1 + x_2 \quad \text{and} \quad h(x) = |x_1| + |x_2| + \frac{1}{2}\left(x_1^2 + x_2^2\right).$$

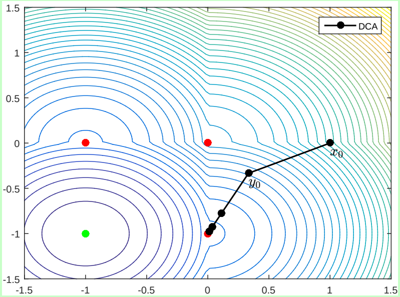


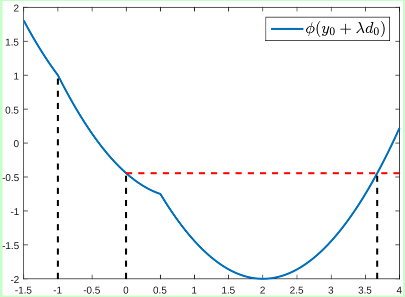
## DCA can be slow and get easily trapped by critical points

### Example (Revisited)

Consider the DC function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined as $\phi := g - h$, where

$$g(x) = \frac{3}{2}\left(x_1^2 + x_2^2\right) + x_1 + x_2 \quad \text{and} \quad h(x) = |x_1| + |x_2| + \frac{1}{2}\left(x_1^2 + x_2^2\right).$$



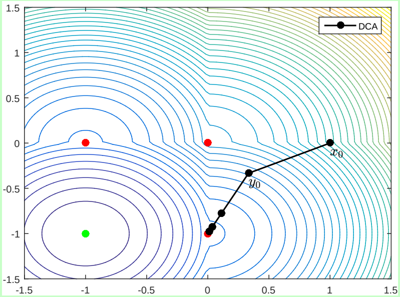


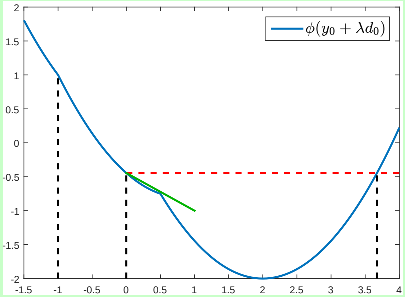
## DCA can be slow and get easily trapped by critical points

**Example (Revisited)**

Consider the DC function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined as $\phi := g - h$, where

$$g(x) = \frac{3}{2}\left(x_1^2 + x_2^2\right) + x_1 + x_2 \quad \text{and} \quad h(x) = |x_1| + |x_2| + \frac{1}{2}\left(x_1^2 + x_2^2\right).$$





$$\boxed{\phi'(y_0; d_0) \leq -\rho \|d_0\|^2}$$

## Outline

## The Boosted DC Algorithm

$$\phi(x) := g(x) - h(x), u_k \in \partial h(x_k), y_k \in \partial g^*(u_k)$$

**Proposition**

*If $g$ is differentiable, then $\phi'(y_k; d_k) \leq -\rho\|d_k\|^2$.*

## The Boosted DC Algorithm

$$\phi(x) := g(x) - h(x), u_k \in \partial h(x_k), y_k \in \partial g^*(u_k)$$

**Proposition**

*If $g$ is differentiable, then $\phi'(y_k; d_k) \leq -\rho\|d_k\|^2$.*

**Proof**

Pick any $v \in \partial h(y_k) \neq \emptyset$.

## The Boosted DC Algorithm

$$\phi(x) := g(x) - h(x), u_k \in \partial h(x_k), y_k \in \partial g^*(u_k)$$

**Proposition**

*If $g$ is differentiable, then $\phi'(y_k; d_k) \leq -\rho \|d_k\|^2$.*

**Proof**

Pick any $v \in \partial h(y_k) \neq \emptyset$. The one-sided directional derivative $\phi'(y_k; d_k)$ is given by

$$\begin{aligned}\phi'(y_k; d_k) &= \lim_{t\downarrow 0} \frac{g(y_k + td_k) - g(y_k)}{t} - \lim_{t\downarrow 0} \frac{h(y_k + td_k) - h(y_k)}{t} \\ &\leq \langle \nabla g(y_k), d_k \rangle - \langle v, d_k \rangle ,\end{aligned}$$

by convexity of $h$.

## The Boosted DC Algorithm

$$\phi(x) := g(x) - h(x), u_k \in \partial h(x_k), y_k \in \partial g^*(u_k)$$

**Proposition**

*If $g$ is differentiable, then $\phi'(y_k; d_k) \leq -\rho\|d_k\|^2$.*

**Proof**

Pick any $v \in \partial h(y_k) \neq \emptyset$. The one-sided directional derivative $\phi'(y_k; d_k)$ is given by

$$\begin{aligned}\phi'(y_k; d_k) &= \lim_{t\downarrow 0} \frac{g(y_k + td_k) - g(y_k)}{t} - \lim_{t\downarrow 0} \frac{h(y_k + td_k) - h(y_k)}{t} \\ &\leq \langle \nabla g(y_k), d_k\rangle - \langle v, d_k\rangle ,\end{aligned}$$

by convexity of $h$. As $y_k$ is the solution of $(\mathcal{P}_k)$, we have

$$\nabla g(y_k) = u_k \in \partial h(x_k).$$

## The Boosted DC Algorithm

$$\phi(x) := g(x) - h(x), u_k \in \partial h(x_k), y_k \in \partial g^*(u_k)$$

**Proposition**

*If $g$ is differentiable, then $\phi'(y_k; d_k) \leq -\rho \|d_k\|^2$.*

**Proof**

Pick any $v \in \partial h(y_k) \neq \emptyset$. The one-sided directional derivative $\phi'(y_k; d_k)$ is given by

$$\begin{aligned}\phi'(y_k; d_k) &= \lim_{t \downarrow 0} \frac{g(y_k + td_k) - g(y_k)}{t} - \lim_{t \downarrow 0} \frac{h(y_k + td_k) - h(y_k)}{t} \\ &\leq \langle \nabla g(y_k), d_k \rangle - \langle v, d_k \rangle ,\end{aligned}$$

by convexity of $h$. As $y_k$ is the solution of $(\mathcal{P}_k)$, we have

$$\nabla g(y_k) = u_k \in \partial h(x_k).$$

Since $\partial h$ is strongly monotone with constant $\rho$ and $v \in \partial h(y_k)$,

$$\langle u_k - v, x_k - y_k \rangle \geq \rho \|x_k - y_k\|^2 = \rho \|d_k\|^2.$$

## The Boosted DC Algorithm

$$\phi(x) := g(x) - h(x), u_k \in \partial h(x_k), y_k \in \partial g^*(u_k)$$

**Proposition**

*If $g$ is differentiable, then $\phi'(y_k; d_k) \leq -\rho\|d_k\|^2$.*

**Proof**

Pick any $v \in \partial h(y_k) \neq \emptyset$. The one-sided directional derivative $\phi'(y_k; d_k)$ is given by

$$\begin{aligned}\phi'(y_k; d_k) &= \lim_{t\downarrow 0} \frac{g(y_k + td_k) - g(y_k)}{t} - \lim_{t\downarrow 0} \frac{h(y_k + td_k) - h(y_k)}{t} \\ &\leq \langle \nabla g(y_k), d_k\rangle - \langle v, d_k\rangle\,,\end{aligned}$$

by convexity of $h$. As $y_k$ is the solution of $(\mathcal{P}_k)$, we have

$$\nabla g(y_k) = u_k \in \partial h(x_k).$$

Since $\partial h$ is strongly monotone with constant $\rho$ and $v \in \partial h(y_k)$,

$$\langle u_k - v, x_k - y_k\rangle \geq \rho\|x_k - y_k\|^2 = \rho\|d_k\|^2.$$

Hence

$$\phi'(y_k; d_k) \leq \langle \nabla g(y_k) - v, d_k\rangle = \langle u_k - v, y_k - x_k\rangle \leq -\rho\|d_k\|^2. \quad \blacksquare$$

## The Boosted DC Algorithm

**BDCA** (*Boosted DC Algorithm*)

Fix $\alpha > 0$ and $0 < \beta < 1$. Let $x_0$ be any initial point and set $k := 0$.

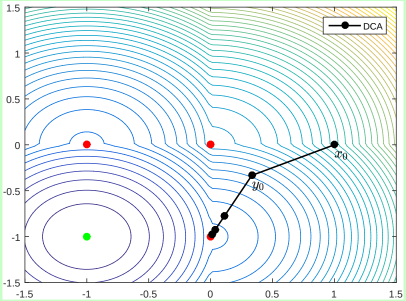
1. Select $u_k \in \partial h(x_k)$ and find the unique solution $y_k$ of the problem
$$(\mathcal{P}_k) \ \underset{x \in \mathbb{R}^m}{\text{minimize}} \ g(x) - \langle u_k, x \rangle.$$
2. Set $d_k := y_k - x_k$. If $d_k = 0 \Rightarrow$ **stop** and **return** $x_k$.
3. Choose any $\overline{\lambda}_k \geq 0$. Set $\lambda_k := \overline{\lambda}_k$.
**while** $\phi(y_k + \lambda_k d_k) > \phi(y_k) - \alpha \lambda_k^2 \|d_k\|^2$ **do** $\lambda_k := \beta \lambda_k$.
4. Set $x_{k+1} := y_k + \lambda_k d_k, k := k + 1$, and **go to** Step 1.

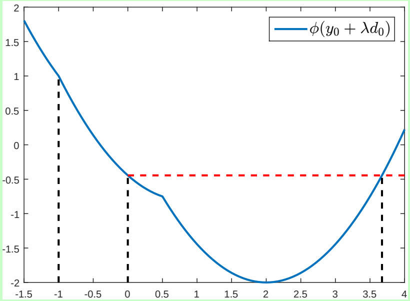
## DCA vs BDCA

### Example (Revisited)

Consider the DC function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined as $\phi := g - h$, where

$$g(x) = \frac{3}{2}\left(x_1^2 + x_2^2\right) + x_1 + x_2 \quad \text{and} \quad h(x) = |x_1| + |x_2| + \frac{1}{2}\left(x_1^2 + x_2^2\right).$$



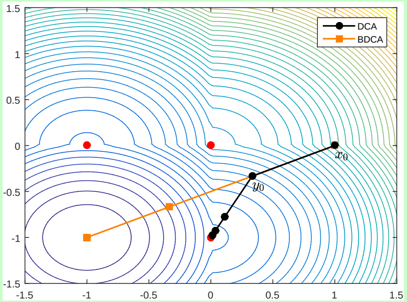


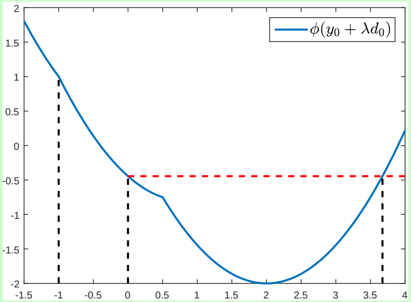
## DCA vs BDCA

### Example (Revisited)

Consider the DC function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined as $\phi := g - h$, where

$$g(x) = \frac{3}{2}\left(x_1^2 + x_2^2\right) + x_1 + x_2 \quad \text{and} \quad h(x) = |x_1| + |x_2| + \frac{1}{2}\left(x_1^2 + x_2^2\right).$$



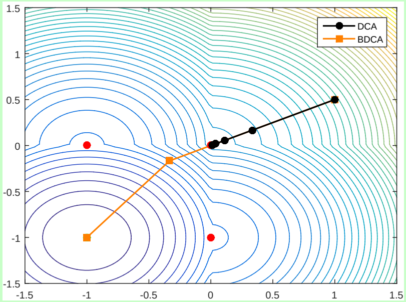


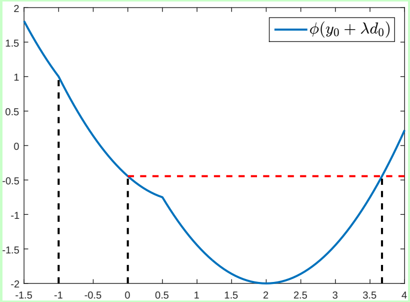
## DCA vs BDCA

### Example (Revisited)

Consider the DC function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined as $\phi := g - h$, where

$$g(x) = \frac{3}{2}\left(x_1^2 + x_2^2\right) + x_1 + x_2 \quad \text{and} \quad h(x) = |x_1| + |x_2| + \frac{1}{2}\left(x_1^2 + x_2^2\right).$$



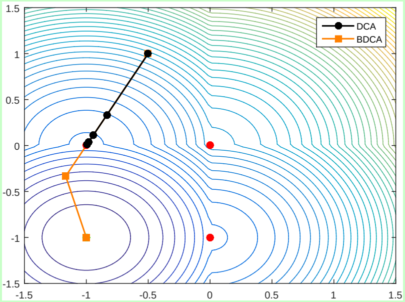


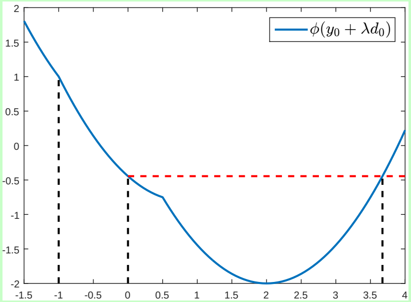
## DCA vs BDCA

### Example (Revisited)

Consider the DC function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined as $\phi := g - h$, where

$$g(x) = \frac{3}{2}\left(x_1^2 + x_2^2\right) + x_1 + x_2 \quad \text{and} \quad h(x) = |x_1| + |x_2| + \frac{1}{2}\left(x_1^2 + x_2^2\right).$$



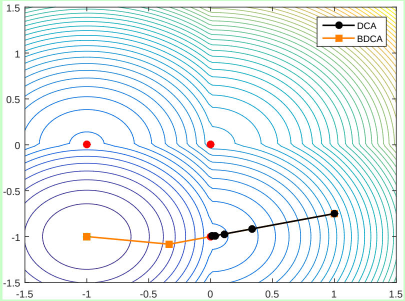


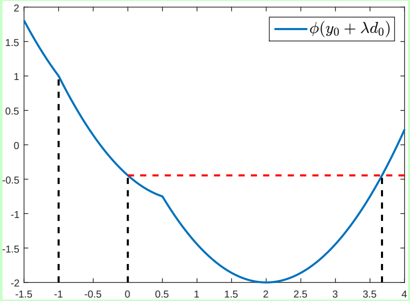
## DCA vs BDCA

### Example (Revisited)

Consider the DC function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined as $\phi := g - h$, where

$$g(x) = \frac{3}{2}\left(x_1^2 + x_2^2\right) + x_1 + x_2 \quad \text{and} \quad h(x) = |x_1| + |x_2| + \frac{1}{2}\left(x_1^2 + x_2^2\right).$$



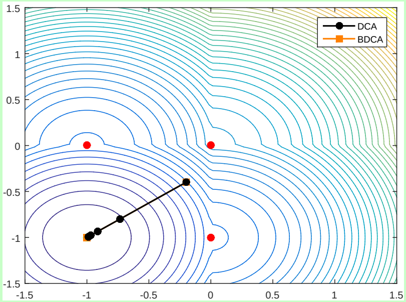


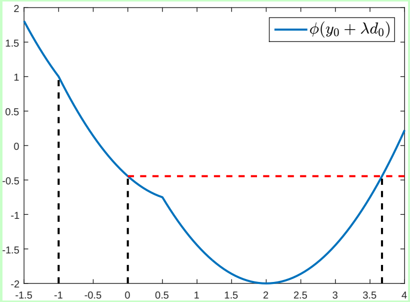
## DCA vs BDCA

### Example (Revisited)

Consider the DC function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined as $\phi := g - h$, where

$$g(x) = \frac{3}{2}\left(x_1^2 + x_2^2\right) + x_1 + x_2 \quad \text{and} \quad h(x) = |x_1| + |x_2| + \frac{1}{2}\left(x_1^2 + x_2^2\right).$$



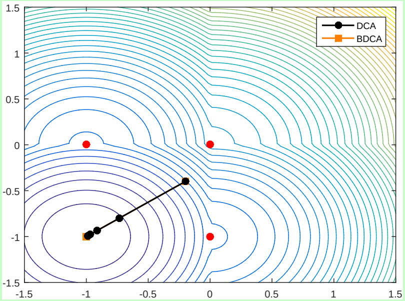


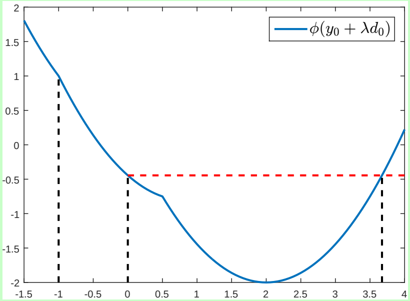
## DCA vs BDCA

### Example (Revisited)

Consider the DC function $\phi : \mathbb{R}^2 \to \mathbb{R}$ defined as $\phi := g - h$, where

$$g(x) = \frac{3}{2}\left(x_1^2 + x_2^2\right) + x_1 + x_2 \quad \text{and} \quad h(x) = |x_1| + |x_2| + \frac{1}{2}\left(x_1^2 + x_2^2\right).$$





| | $(-1,-1)$ | $(-1,0)$ | $(0,-1)$ | $(0,0)$ |
|---|---|---|---|---|
| DCA | 249,763 | 249,841 | 250,204 | 250,192 |
| BDCA | 996,104 | 1,922 | 1,974 | 0 |

Table: For one million random starting points in $[-1.5, 1.5]^2$, we count the sequences converging to each of the four stationary points.

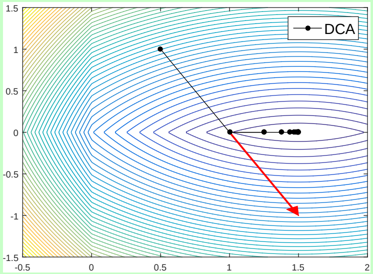
## Why to restrict to the case where $g$ is differentiable?

**Example (Failure of BDCA when $g$ is not differentiable)**

Consider the following modification of the previous example

$$g(x) = -\frac{5}{2}x_1 + x_1^2 + x_2^2 + |x_1| + |x_1| \quad \text{and} \quad h(x) = \frac{1}{2}\left(x_1^2 + x_2^2\right),$$

so $h$ is differentiable but $g$ is not. Let $x_0 = (0.5, 1)$. The point generated by DCA is $y_0 = (1, 0)$ and $d_0 = y_0 - x_0 = (0.5, -1)$ is not a descent direction for $\phi$ at $y_0$:



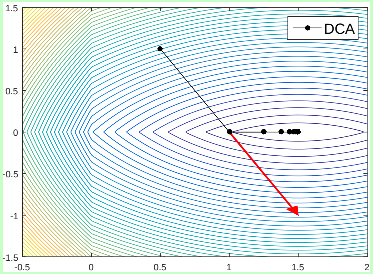
## Why to restrict to the case where $g$ is differentiable?

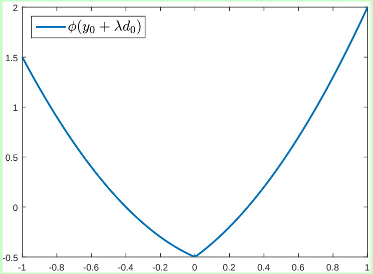
**Example (Failure of BDCA when $g$ is not differentiable)**

Consider the following modification of the previous example

$$g(x) = -\frac{5}{2}x_1 + x_1^2 + x_2^2 + |x_1| + |x_1| \quad \text{and} \quad h(x) = \frac{1}{2}\left(x_1^2 + x_2^2\right),$$

so $h$ is differentiable but $g$ is not. Let $x_0 = (0.5, 1)$. The point generated by DCA is $y_0 = (1, 0)$ and $d_0 = y_0 - x_0 = (0.5, -1)$ is not a descent direction for $\phi$ at $y_0$:





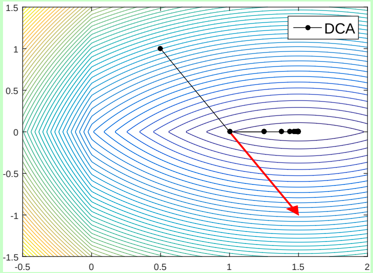
## Why to restrict to the case where $g$ is differentiable?

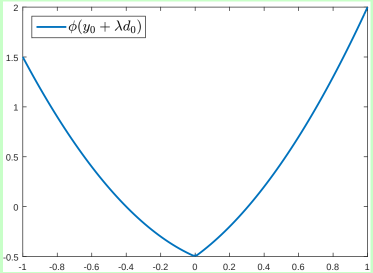
### Example (Failure of BDCA when $g$ is not differentiable)

Consider the following modification of the previous example

$$g(x) = -\frac{5}{2}x_1 + x_1^2 + x_2^2 + |x_1| + |x_1| \quad \text{and} \quad h(x) = \frac{1}{2}\left(x_1^2 + x_2^2\right),$$

so $h$ is differentiable but $g$ is not. Let $x_0 = (0.5, 1)$. The point generated by DCA is $y_0 = (1, 0)$ and $d_0 = y_0 - x_0 = (0.5, -1)$ is not a descent direction for $\phi$ at $y_0$:





### Proposition

*Let $\phi = g - h$, where $g : \mathbb{R} \to \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$ are convex and $h$ is differentiable. If $0 \notin \partial_C \phi(y_k)$, then $\phi'(y_k; y_k - x_k) < 0$.*

## Convergence of BDCA

$$(\mathcal{P})\ \underset{x\in\mathbb{R}^m}{\text{minimize}}\ \ \phi(x) := g(x) - h(x)$$

Our convergence results follow the ideas from

H. Attouch, J. Bolte: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Math. Program.* 116 (2009), 5–16.

which in turn were adapted from Łojasiewicz's original ideas.

## Convergence of BDCA

$$(\mathcal{P})\ \underset{x\in\mathbb{R}^m}{\text{minimize}}\ \ \phi(x) := g(x) - h(x)$$

Our convergence results follow the ideas from

H. Attouch, J. Bolte: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Math. Program.* 116 (2009), 5–16.

which in turn were adapted from Łojasiewicz's original ideas.

**Theorem**

For any $x_0 \in \mathbb{R}^m$, either BDCA returns a critical point of $(\mathcal{P})$ or it generates an infinite sequence such that the following holds.

1. $\phi(x_k)$ is monotonically decreasing and convergent to some $\phi^*$.

## Convergence of BDCA

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

Our convergence results follow the ideas from

H. Attouch, J. Bolte: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Math. Program.* 116 (2009), 5–16.

which in turn were adapted from Łojasiewicz's original ideas.

**Theorem**

For any $x_0 \in \mathbb{R}^m$, either BDCA returns a critical point of $(\mathcal{P})$ or it generates an infinite sequence such that the following holds.

1. $\phi(x_k)$ is monotonically decreasing and convergent to some $\phi^*$.
2. Any limit point of $\{x_k\}$ is a critical point of $(\mathcal{P})$. If in addition, $\phi$ is coercive then there exits a subsequence of $\{x_k\}$ which converges to a critical point of $(\mathcal{P})$.

## Convergence of BDCA

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

Our convergence results follow the ideas from

H. Attouch, J. Bolte: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Math. Program.* 116 (2009), 5–16.

which in turn were adapted from Łojasiewicz's original ideas.

**Theorem**

For any $x_0 \in \mathbb{R}^m$, either BDCA returns a critical point of $(\mathcal{P})$ or it generates an infinite sequence such that the following holds.

1. $\phi(x_k)$ is monotonically decreasing and convergent to some $\phi^*$.
2. Any limit point of $\{x_k\}$ is a critical point of $(\mathcal{P})$. If in addition, $\phi$ is coercive then there exits a subsequence of $\{x_k\}$ which converges to a critical point of $(\mathcal{P})$.
3. $\sum_{k=0}^{+\infty} \|d_k\|^2 < +\infty$. Further, if there is some $\overline{\lambda}$ such that $\lambda_k \le \overline{\lambda}$ for all $k$, then $\sum_{k=0}^{+\infty} \|x_{k+1} - x_k\|^2 < +\infty$.

## Outline

## The Kurdyka–Łojasiewicz property

**Definition**

Let $f:\mathbb{R}^m\to\mathbb{R}$ be a locally Lipschitz function. We say that $f$ satisfies the strong Kurdyka–Łojasiewicz inequality at $x^*\in\mathbb{R}^m$ if there exist $\eta\in\,]0,+\infty[$, a neighborhood $U$ of $x^*$, and a concave function $\varphi:[0,\eta]\to[0,+\infty[$ such that:

1. $\varphi(0)=0$;
2. $\varphi$ is of class $\mathcal{C}^1$ on $]0,\eta[$;
3. $\varphi'>0$ on $]0,\eta[$;
4. for all $x\in U$ with $f(x^*)<f(x)<f(x^*)+\eta$ we have
$$\varphi'(f(x)-f(x^*))\,\mathrm{dist}\,(0,\partial_C f(x))\geq 1.$$

Here $\partial_C f$ stands for the Clarke subdifferential

$$\partial_C f(\bar{x})=\mathrm{co}\left\{\lim_{x\to\bar{x},\,x\notin\Omega_f}\nabla f(x)\right\},$$

where $\mathrm{co}$ stands for the convex hull and $\Omega_f$ denotes the set of Lebesgue measure zero where $f$ fails to be differentiable.

## Convergence under the Kurdyka–Łojasiewicz property

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

**Theorem (Convergence)**

Let $\{x_k\}$ be the sequence generated by the BDCA. Suppose that $\{x_k\}$ has a cluster point $x^*$, that $\nabla g$ is locally Lipschitz around $x^*$ and that $\phi$ satisfies the strong Kurdyka–Łojasiewicz inequality at $x^*$.
Then $\{x_k\}$ converges to $x^*$, which is a critical point of $(\mathcal{P})$.

## Convergence under the Kurdyka–Łojasiewicz property

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

**Theorem (Convergence)**

Let $\{x_k\}$ be the sequence generated by the BDCA. Suppose that $\{x_k\}$ has a cluster point $x^*$, that $\nabla g$ is locally Lipschitz around $x^*$ and that $\phi$ satisfies the strong Kurdyka–Łojasiewicz inequality at $x^*$.
Then $\{x_k\}$ converges to $x^*$, which is a critical point of $(\mathcal{P})$.

**Proof**

- Technical but “standard”.
- $\lambda_k$ can be zero or unbounded!
- We either need Clarke’s subdifferential or to assume that $-\phi$ satisfies the Kurdyka–Łojasiewicz inequality:

$$\nabla g(y_k) - \nabla g(x_k) \in \partial h(x_k) - \nabla g(x_k) = \boxed{\partial_C\left(-\phi(x_k)\right) = -\partial_C \phi(x_k)}$$

## Convergence under the Kurdyka–Łojasiewicz property

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

**Theorem (Convergence)**

Let $\{x_k\}$ be the sequence generated by the BDCA. Suppose that $\{x_k\}$ has a cluster point $x^*$, that $\nabla g$ is locally Lipschitz around $x^*$ and that $\phi$ satisfies the strong Kurdyka–Łojasiewicz inequality at $x^*$.
Then $\{x_k\}$ converges to $x^*$, which is a critical point of $(\mathcal{P})$.

**Theorem (Rate)**

Suppose that the sequence $\{x_k\}$ generated by the BDCA has the limit point $x^*$, that $\nabla g$ is locally Lipschitz continuous around $x^*$ and $\phi$ satisfies the strong Kurdyka–Łojasiewicz inequality at $x^*$ with $\varphi(t) = Mt^{1-\theta}$ for some $M > 0$ and $0 \leq \theta < 1$. Then:

1. if $\theta = 0$, then $\{x_k\}$ converges in a finite number of steps to $x^*$;

## Convergence under the Kurdyka–Łojasiewicz property

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

**Theorem (Convergence)**

Let $\{x_k\}$ be the sequence generated by the BDCA. Suppose that $\{x_k\}$ has a cluster point $x^*$, that $\nabla g$ is locally Lipschitz around $x^*$ and that $\phi$ satisfies the strong Kurdyka–Łojasiewicz inequality at $x^*$.
Then $\{x_k\}$ converges to $x^*$, which is a critical point of $(\mathcal{P})$.

**Theorem (Rate)**

Suppose that the sequence $\{x_k\}$ generated by the BDCA has the limit point $x^*$, that $\nabla g$ is locally Lipschitz continuous around $x^*$ and $\phi$ satisfies the strong Kurdyka–Łojasiewicz inequality at $x^*$ with $\varphi(t) = Mt^{1-\theta}$ for some $M > 0$ and $0 \leq \theta < 1$. Then:

1. if $\theta = 0$, then $\{x_k\}$ converges in a finite number of steps to $x^*$;
2. if $\theta \in \left]0, \frac{1}{2}\right]$, then $\{x_k\}$ converges linearly to $x^*$;

## Convergence under the Kurdyka–Łojasiewicz property

$$(\mathcal{P}) \ \underset{x\in\mathbb{R}^m}{\text{minimize}} \ \ \phi(x) := g(x) - h(x)$$

**Theorem (Convergence)**

Let $\{x_k\}$ be the sequence generated by the BDCA. Suppose that $\{x_k\}$ has a cluster point $x^*$, that $\nabla g$ is locally Lipschitz around $x^*$ and that $\phi$ satisfies the strong Kurdyka–Łojasiewicz inequality at $x^*$.
Then $\{x_k\}$ converges to $x^*$, which is a critical point of $(\mathcal{P})$.

**Theorem (Rate)**

Suppose that the sequence $\{x_k\}$ generated by the BDCA has the limit point $x^*$, that $\nabla g$ is locally Lipschitz continuous around $x^*$ and $\phi$ satisfies the strong Kurdyka–Łojasiewicz inequality at $x^*$ with $\varphi(t) = Mt^{1-\theta}$ for some $M > 0$ and $0 \leq \theta < 1$. Then:

1. if $\theta = 0$, then $\{x_k\}$ converges in a finite number of steps to $x^*$;
2. if $\theta \in \left]0, \frac{1}{2}\right]$, then $\{x_k\}$ converges linearly to $x^*$;
3. if $\theta \in \left]\frac{1}{2}, 1\right[$, then $\exists \eta > 0$ s.t. $\|x_k - x^*\| \leq \eta k^{-\frac{1-\theta}{2\theta-1}}$ for all large $k$.

## How to choose the trial step size $\overline{\lambda}_k$?

**BDCA** (*Boosted DC Algorithm*)

Fix $\alpha > 0$ and $0 < \beta < 1$. Let $x_0$ be any initial point and set $k := 0$.

1. Select $u_k \in \partial h(x_k)$ and $y_k \in \partial g^*(u_k)$.
2. Set $d_k := y_k - x_k$. If $d_k = 0 \Rightarrow$ **stop** and **return** $x_k$.
3. Choose any $\overline{\lambda}_k \geq 0$. Set $\lambda_k := \overline{\lambda}_k$.
   **while** $\phi(y_k + \lambda_k d_k) > \phi(y_k) - \alpha\lambda_k^2\|d_k\|^2$ **do** $\lambda_k := \beta\lambda_k$.
4. Set $x_{k+1} := y_k + \lambda_k d_k, k := k + 1$, and **go to** Step 1.

One possibility would be to set $\overline{\lambda}_k = \overline{\lambda}$ for all $k$.

## How to choose the trial step size $\overline{\lambda}_k$?

**BDCA** (*Boosted DC Algorithm*)

Fix $\alpha > 0$ and $0 < \beta < 1$. Let $x_0$ be any initial point and set $k := 0$.

1. Select $u_k \in \partial h(x_k)$ and $y_k \in \partial g^*(u_k)$.
2. Set $d_k := y_k - x_k$. If $d_k = 0 \Rightarrow$ **stop** and **return** $x_k$.
3. Choose any $\overline{\lambda}_k \geq 0$. Set $\lambda_k := \overline{\lambda}_k$.
   **while** $\phi(y_k + \lambda_k d_k) > \phi(y_k) - \alpha\lambda_k^2\|d_k\|^2$ **do** $\lambda_k := \beta\lambda_k$.
4. Set $x_{k+1} := y_k + \lambda_k d_k, k := k + 1$, and **go to** Step 1.

One possibility would be to set $\overline{\lambda}_k = \overline{\lambda}$ for all $k$. Instead, we propose:

**Self-adaptive trial step size**

Fix $\gamma > 1$. Set $\overline{\lambda}_0 := 0$. Choose some $\overline{\lambda}_1 > 0$ and obtain $\lambda_1$ by BDCA.
For any $k \geq 2$:

1. **if** $\lambda_{k-2} = \overline{\lambda}_{k-2}$ **and** $\lambda_{k-1} = \overline{\lambda}_{k-1}$ **then** set $\overline{\lambda}_k := \gamma\lambda_{k-1}$;
   **else** set $\overline{\lambda}_k := \lambda_{k-1}$.
2. Obtain $\lambda_k$ from $\overline{\lambda}_k$ by the backtracking step of BDCA.

## Outline

## Experiment 1: Finding steady states of biochemical networks

We were interested in finding a solution to the problem

$$\underset{x\in\mathbb{R}^m}{\text{minimize}} \;\; \phi(x) := \|p(x) - c(x)\|^2$$

where

$$p(x) := [F,R]\exp\left(p + [F,R]^T x\right) \quad \text{and} \quad c(x) := [R,F]\exp\left(p + [F,R]^T x\right),$$

and $F, R \in \mathbb{Z}_{\geq 0}^{m\times n}$ ($m$ molecular species, $n$ reversible elementary reactions).

## Experiment 1: Finding steady states of biochemical networks

We were interested in finding a solution to the problem

$$\underset{x\in\mathbb{R}^m}{\text{minimize}} \;\; \phi(x) := \|p(x)-c(x)\|^2 = 2\left(\|p(x)\|^2 + \|c(x)\|^2\right) - \|p(x)+c(x)\|^2,$$

where

$$p(x) := [F,R]\exp\left(p + [F,R]^T x\right) \quad \text{and} \quad c(x) := [R,F]\exp\left(p + [F,R]^T x\right),$$

and $F, R \in \mathbb{Z}_{\geq 0}^{m\times n}$ ($m$ molecular species, $n$ reversible elementary reactions).

## Experiment 1: Finding steady states of biochemical networks

We were interested in finding a solution to the problem

$$\underset{x\in\mathbb{R}^m}{\text{minimize}}\ \phi(x) := \|p(x)-c(x)\|^2 = 2\left(\|p(x)\|^2+\|c(x)\|^2\right) - \|p(x)+c(x)\|^2,$$

where

$$p(x) := [F,R]\exp\left(p+[F,R]^T x\right) \quad \text{and} \quad c(x) := [R,F]\exp\left(p+[F,R]^T x\right),$$

and $F, R \in \mathbb{Z}_{\geq 0}^{m\times n}$ ($m$ molecular species, $n$ reversible elementary reactions).

- It is not difficult to prove that $f$ is real analytic
  $\Rightarrow f$ satisfies the Łojasiewicz property with some exponent $\theta \in [0,1)$.

## Experiment 1: Finding steady states of biochemical networks

We were interested in finding a solution to the problem

$$\underset{x\in\mathbb{R}^m}{\text{minimize}}\ \phi(x) := \|p(x)-c(x)\|^2 = 2\left(\|p(x)\|^2+\|c(x)\|^2\right) - \|p(x)+c(x)\|^2,$$

where

$$p(x) := [F,R]\exp\left(p+[F,R]^T x\right) \quad \text{and} \quad c(x) := [R,F]\exp\left(p+[F,R]^T x\right),$$

and $F, R \in \mathbb{Z}_{\geq 0}^{m\times n}$ ($m$ molecular species, $n$ reversible elementary reactions).

- It is not difficult to prove that $f$ is real analytic
  $\Rightarrow f$ satisfies the Łojasiewicz property with some exponent $\theta \in [0,1)$.
- We can then apply BDCA to the functions

  $$g(x) := 2\left(\|p(x)\|^2+\|c(x)\|^2\right)+\frac{\rho}{2}\|x\|^2 \quad \text{and} \quad h(x) := \|p(x)+c(x)\|^2+\frac{\rho}{2}\|x\|^2,$$

  for any $\rho > 0$. Our results guarantee the convergence of the sequence generated by BDCA, as long as the sequence is bounded.

# Finding steady states of biochemical networks

A boring table of computational results. . .

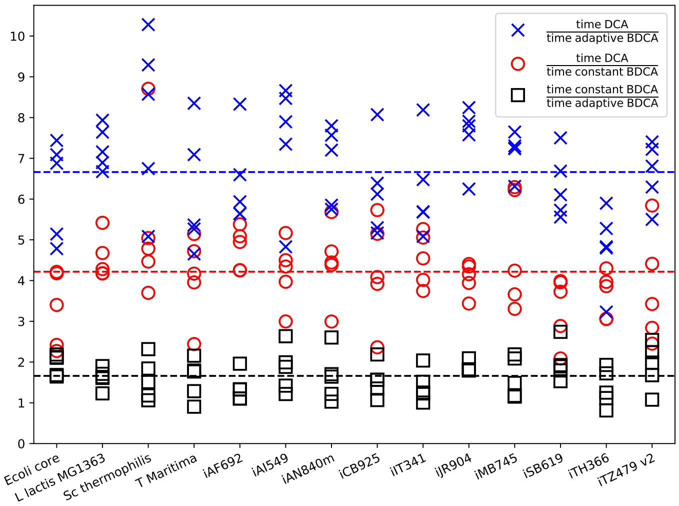
Taking $\rho := 100$, $\overline{\lambda}_k := 50$ (constant), $\beta := 0.5$ and $\alpha := 0.4$, we obtain:

| DATA | | | ALGORITHMS: Time Spent (seconds) | | | | | | RATIO (avg.) DCA/BDCA | |
|---|---|---|---|---|---|---|---|---|---|---|
| Model Name | m | n | BDCA | | | DCA | | | | |
| | | | min. | max. | avg. | min. | max. | avg. | iter. | time |
| Ecoli core | 72 | 94 | 16 | 26 | 20 | 68 | 105 | 87 | 4.9 | 4.4 |
| L lactis MG1363 | 486 | 615 | 2926 | 4029 | 3424 | 14522 | 18212 | 16670 | 5.2 | 4.9 |
| Sc thermophilis | 349 | 444 | 291 | 553 | 358 | 1302 | 2004 | 1611 | 4.9 | 4.5 |
| T Maritima | 434 | 554 | 1333 | 2623 | 1920 | 5476 | 12559 | 8517 | 4.7 | 4.4 |
| iAF692 | 466 | 546 | 1677 | 2275 | 1967 | 8337 | 11187 | 9466 | 5.3 | 4.8 |
| iAI549 | 307 | 355 | 177 | 254 | 209 | 665 | 1078 | 913 | 4.9 | 4.4 |
| iAN840m | 549 | 840 | 3229 | 6939 | 4721 | 16473 | 28957 | 21413 | 5.0 | 4.5 |
| iCB925 | 416 | 584 | 1831 | 2450 | 2133 | 7358 | 11465 | 9887 | 5.0 | 4.6 |
| iIT341 | 425 | 504 | 1925 | 2883 | 2302 | 9434 | 20310 | 12262 | 5.7 | 5.3 |
| iJR904 | 597 | 915 | 6363 | 9836 | 7623 | 24988 | 43640 | 33621 | 4.4 | 4.8 |
| iMB745 | 528 | 652 | 2629 | 5091 | 4252 | 16438 | 25172 | 20269 | 5.0 | 4.8 |
| iSB619 | 462 | 598 | 2407 | 5972 | 3323 | 8346 | 25468 | 13967 | 4.3 | 4.2 |
| iTH366 | 587 | 713 | 3310 | 5707 | 4464 | 13613 | 30044 | 20715 | 5.0 | 4.6 |
| iTZ479 v2 | 435 | 560 | 1211 | 2656 | 2216 | 7368 | 12592 | 10120 | 4.9 | 4.6 |

For each model, we selected a random kinetic parameter $p \in [-1, 1]^{2n}$ and 10 initial random points $x_0 \in [-2, 2]^m$. BDCA was run 1000 iterations, DCA was run until it reached the same value obtained by BDCA.

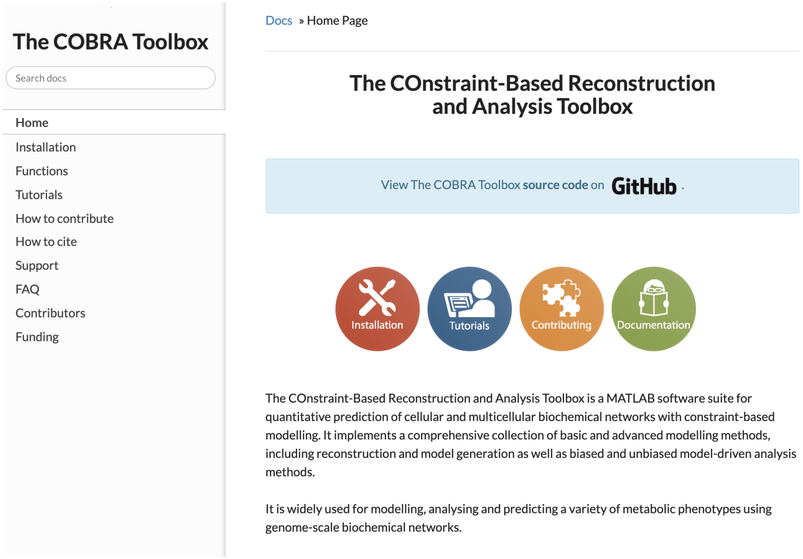
# Finding steady states of biochemical networks

## Comparison of the constant and self-adaptive trial step size strategy for BDCA



## Applications to Biochemistry

Nature Protocols (2019)



The COBRA Toolbox

Search docs

Home
Installation
Functions
Tutorials
How to contribute
How to cite
Support
FAQ
Contributors
Funding

Docs » Home Page

The COnstraint-Based Reconstruction and Analysis Toolbox

View The COBRA Toolbox **source code** on GitHub.

The COnstraint-Based Reconstruction and Analysis Toolbox is a MATLAB software suite for quantitative prediction of cellular and multicellular biochemical networks with constraint-based modelling. It implements a comprehensive collection of basic and advanced modelling methods, including reconstruction and model generation as well as biased and unbiased model-driven analysis methods.

It is widely used for modelling, analysing and predicting a variety of metabolic phenotypes using genome-scale biochemical networks.

## Outline

## The Minimum Sum-of-Squares Clustering Problem

Clustering is an unsupervised technique for data analysis whose objective is to group a collection of objects into clusters based on similarity.

- Let $A = \{a^1, \ldots, a^n\}$ be a finite set of points in $\mathbb{R}^m$, which represent the data points to be grouped.
- The goal is to partition $A$ into $k$ disjoint subsets $A^1, \ldots, A^k$, called clusters, such that a clustering criterion is optimized.

## The Minimum Sum-of-Squares Clustering Problem

Clustering is an unsupervised technique for data analysis whose objective is to group a collection of objects into clusters based on similarity.

- Let $A = \{a^1, \ldots, a^n\}$ be a finite set of points in $\mathbb{R}^m$, which represent the data points to be grouped.
- The goal is to partition $A$ into $k$ disjoint subsets $A^1, \ldots, A^k$, called clusters, such that a clustering criterion is optimized.

The *Minimum Sum-of-Squares Clustering* criterion, one tries to minimize the Euclidean distance of each data point to the centroid of its clusters, denoted by $X := (x^1, \ldots, x^k) \in \mathbb{R}^{m\times k}$:

$$\underset{X\in\mathbb{R}^{m\times k}}{\text{minimize}}\ \phi(X) := \frac{1}{n}\sum_{i=1}^{n} \min_{j=1,\ldots,k} \|x^j - a^i\|^2$$

## The Minimum Sum-of-Squares Clustering Problem

Clustering is an unsupervised technique for data analysis whose objective is to group a collection of objects into clusters based on similarity.

- Let $A = \{a^1, \ldots, a^n\}$ be a finite set of points in $\mathbb{R}^m$, which represent the data points to be grouped.
- The goal is to partition $A$ into $k$ disjoint subsets $A^1, \ldots, A^k$, called clusters, such that a clustering criterion is optimized.

The *Minimum Sum-of-Squares Clustering* criterion, one tries to minimize the Euclidean distance of each data point to the centroid of its clusters, denoted by $X := (x^1, \ldots, x^k) \in \mathbb{R}^{m\times k}$:

$$\underset{X\in\mathbb{R}^{m\times k}}{\text{minimize}}\ \phi(X) := \frac{1}{n}\sum_{i=1}^{n}\min_{j=1,\ldots,k}\|x^j - a^i\|^2 = g(X) - h(X),$$

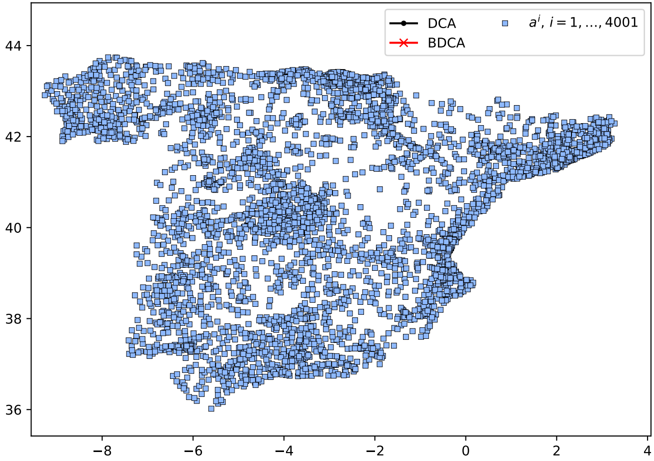
where

$$g(X) := \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\left\|x^j - a^i\right\|^2 + \frac{\rho}{2}\|X\|^2, \quad \text{(strongly convex and smooth)}$$

$$h(X) := \frac{1}{n}\sum_{i=1}^{n}\max_{j=1,\ldots,k}\sum_{t=1,t\neq j}^{k}\left\|x^t - a^i\right\|^2 + \frac{\rho}{2}\|X\|^2. \quad \text{(strongly convex but nonsmooth).}$$

## Experiment 2: Clustering the Spanish cities in the peninsula

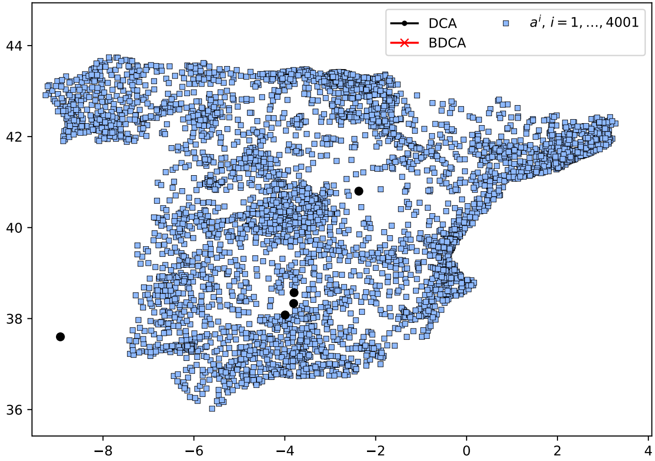
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1]The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

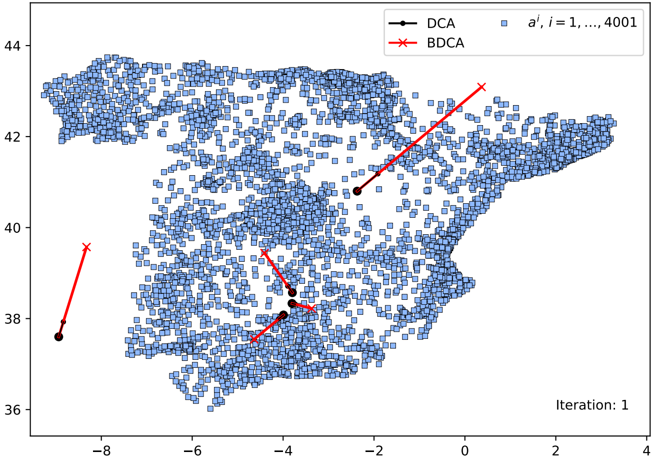
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1]The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

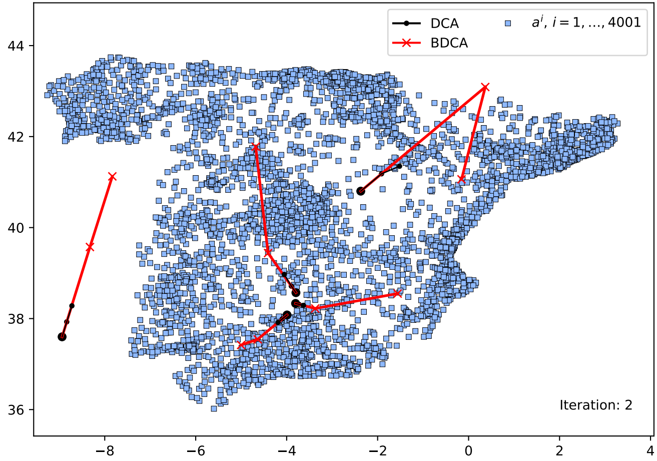
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## **Experiment 2:** Clustering the Spanish cities in the peninsula

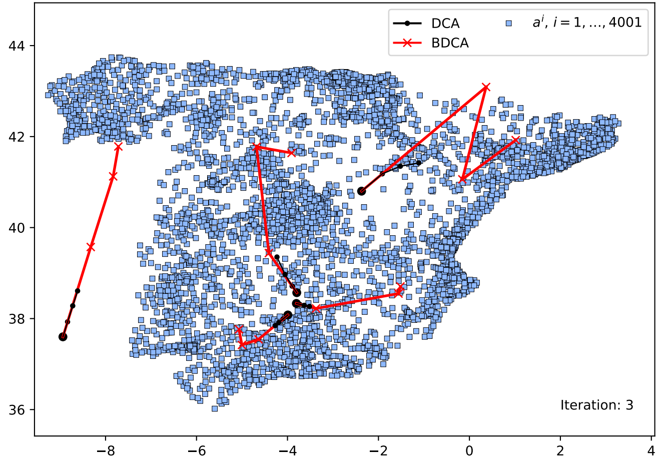
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## **Experiment 2:** Clustering the Spanish cities in the peninsula

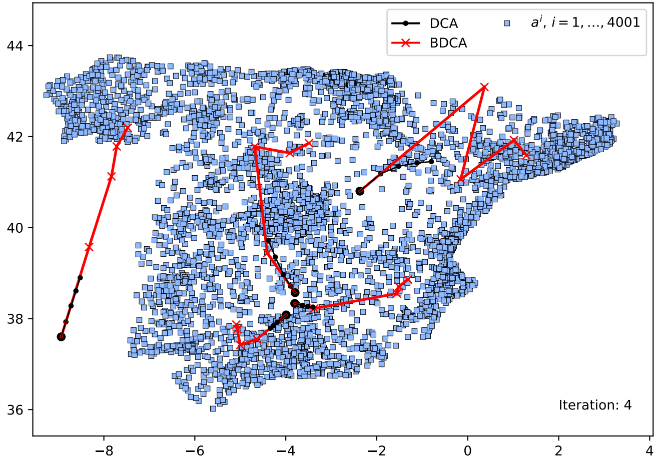
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## **Experiment 2:** Clustering the Spanish cities in the peninsula

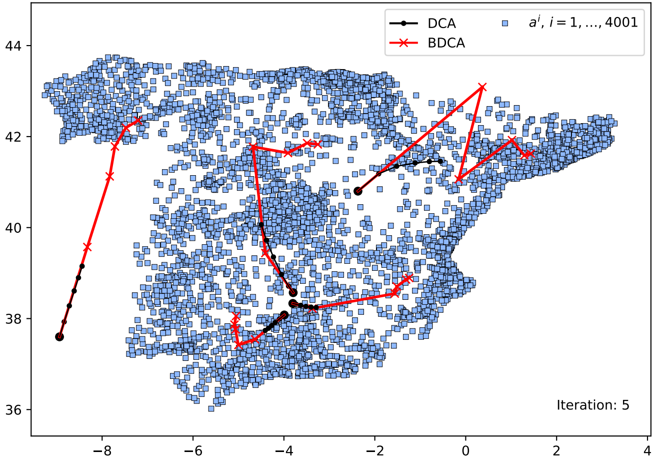
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

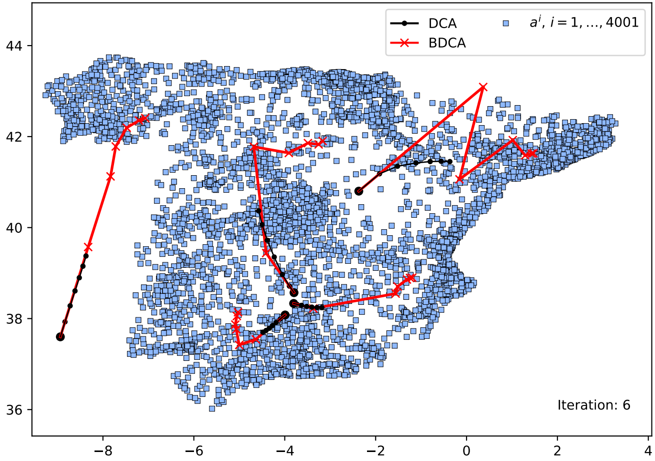
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

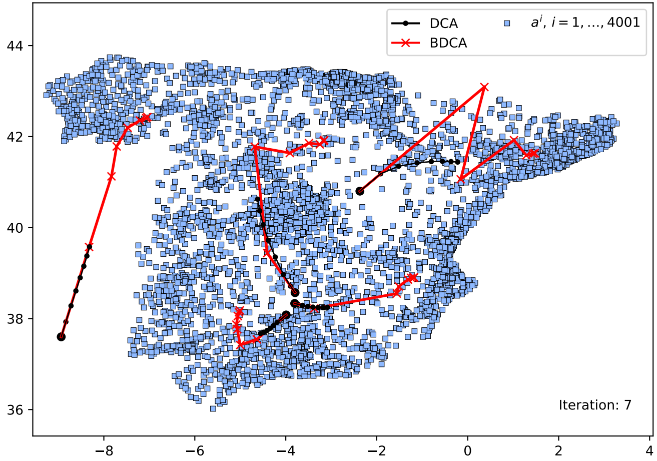
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

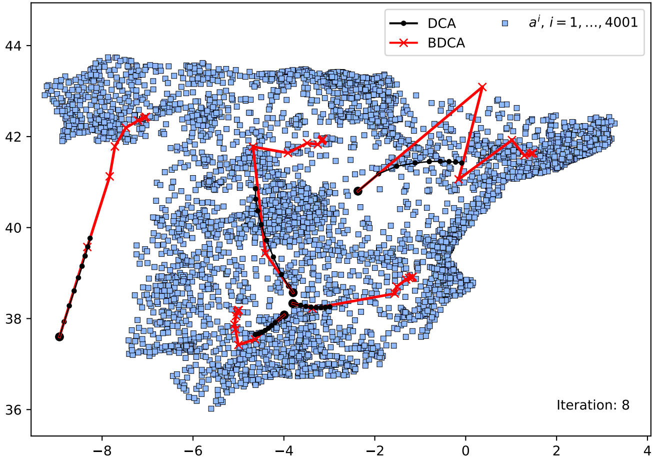
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

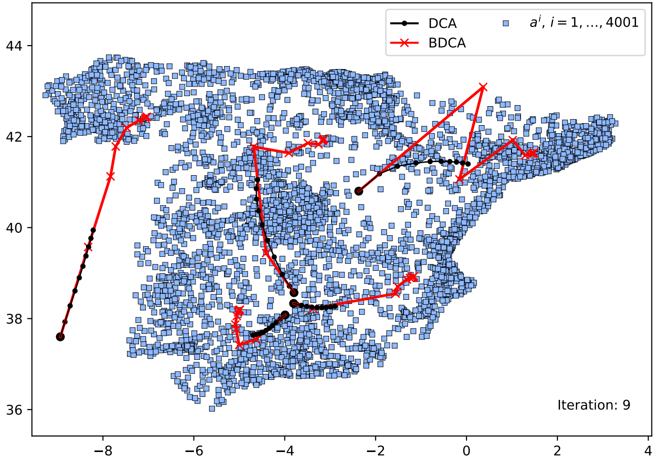
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## **Experiment 2:** Clustering the Spanish cities in the peninsula

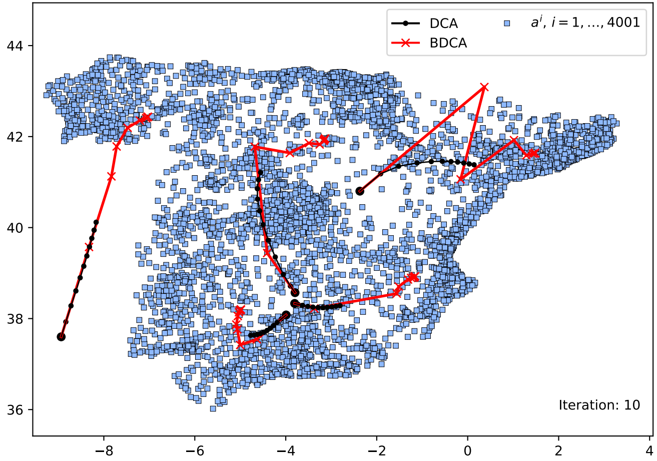
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1]The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

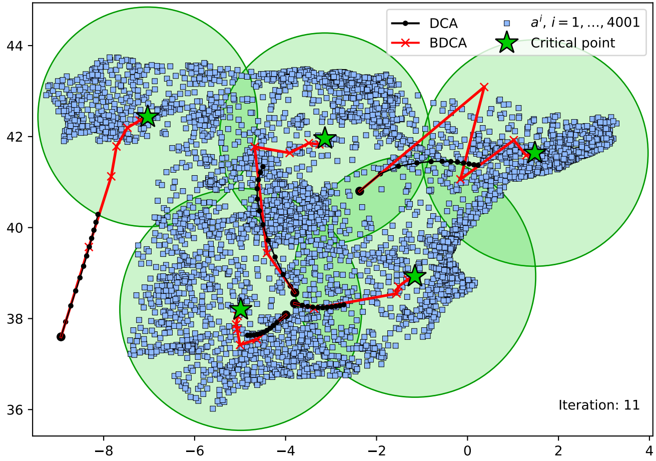
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

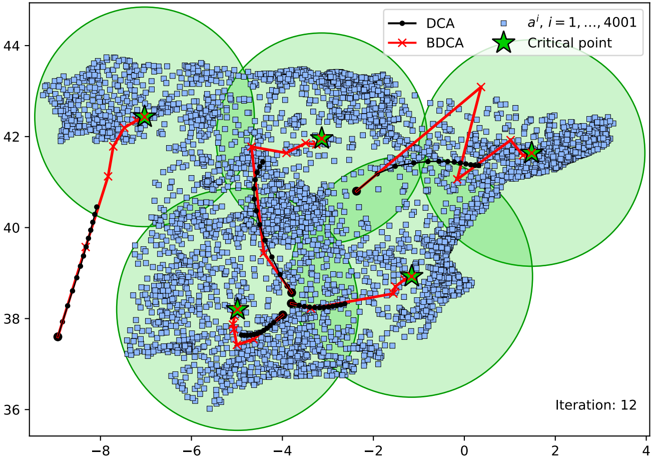
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

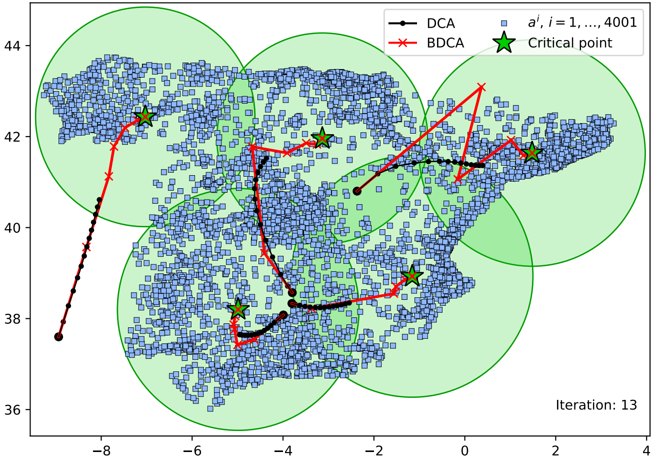
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

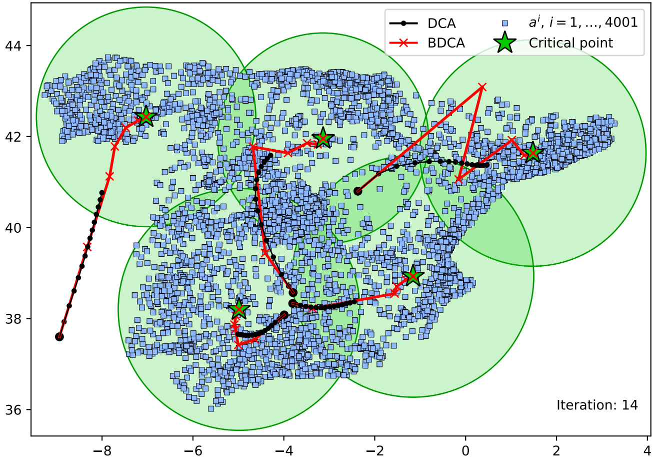
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

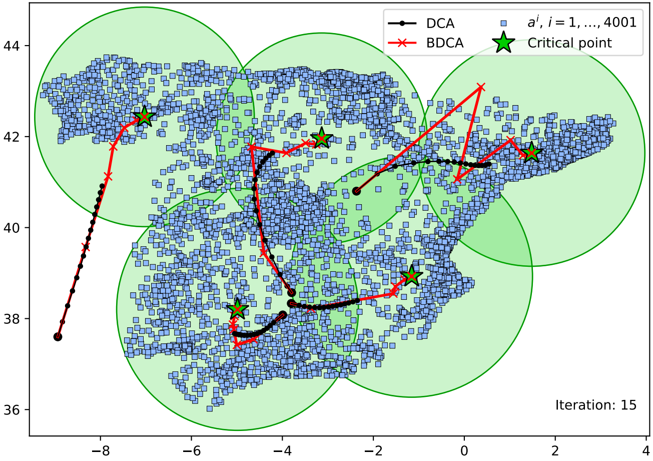
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

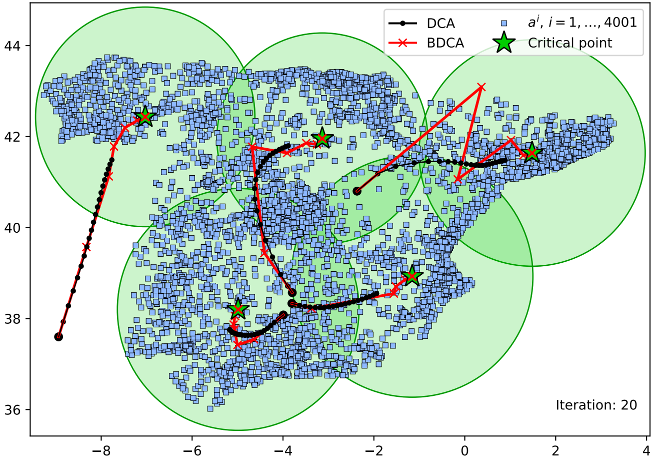
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

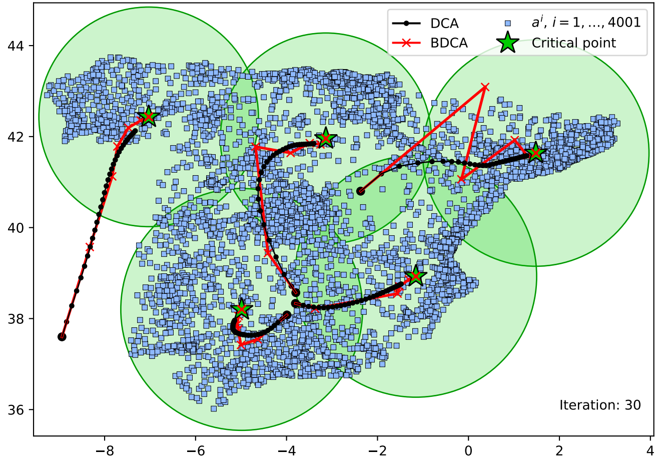
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

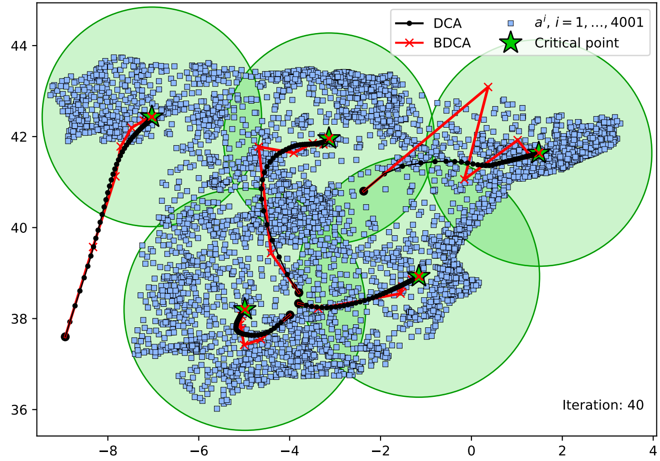
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

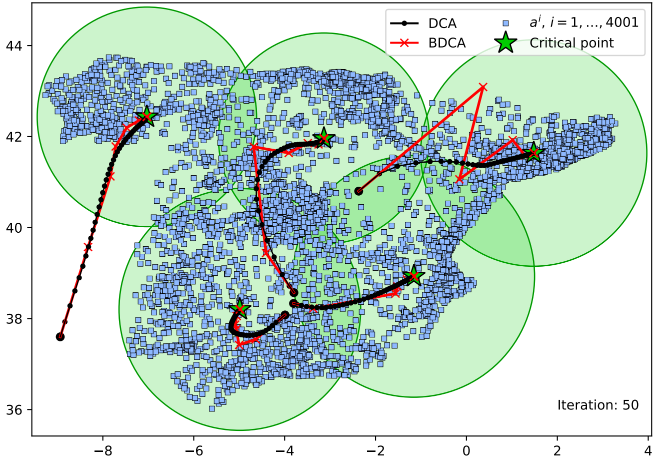
**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

**PROBLEM:** Find a partition into 5 clusters of the 4001 Spanish cities in the peninsula with more than 500 residents[1].



[1] The data can be retrieved from the Spanish National Center of Geographic Information at http://centrodedescargas.cnig.es

## Experiment 2: Clustering the Spanish cities in the peninsula

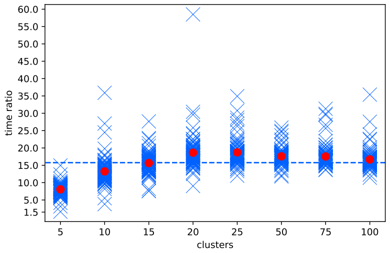
**PROBLEM:** Find a partition into $k$ clusters of the 4001 Spanish cities in the peninsula with more than 500 residents, with $k \in \{5, 10, 15, 20, 25, 50, 75, 100\}$.

For 100 random starting points, BDCA was stopped when the relative error of $\phi$ was smaller than $10^{-3}$. Then, DCA was run from the same starting point until the same value of the objective function was reached (which did not happen in 31 instances).

## Experiment 2: Clustering the Spanish cities in the peninsula

**PROBLEM:** Find a partition into $k$ clusters of the 4001 Spanish cities in the peninsula with more than 500 residents, with $k \in \{5, 10, 15, 20, 25, 50, 75, 100\}$.

For 100 random starting points, BDCA was stopped when the relative error of $\phi$ was smaller than $10^{-3}$. Then, DCA was run from the same starting point until the same value of the objective function was reached (which did not happen in 31 instances).



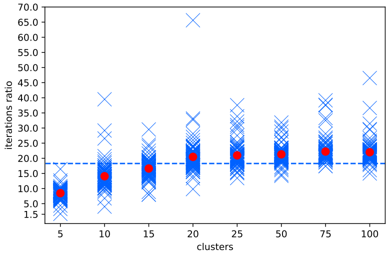


Figure: Comparison between DCA and BDCA. We represent the ratios of running time (left) and number of iterations (right) between DCA and BDCA.

## Experiment 3: Clustering random points in an $m$-dimensional box

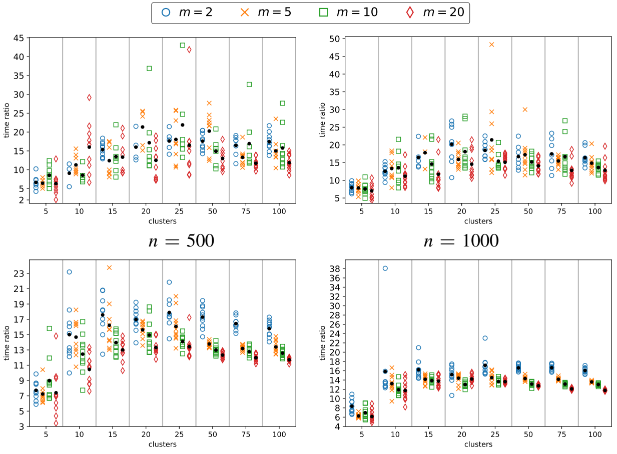
We took $n$ random points in $\mathbb{R}^m$ (normal distribution), with $n \in \{500, 1000, 5000, 10000\}$ and $m \in \{2, 5, 10, 20\}$. For each $(n, m)$, 10 random starting points were chosen. Then:

- BDCA was run to solve the $k$-clustering, with $k \in \{5, 10, 15, 20, 25, 50, 75, 100\}$.
- DCA was run until the same value of $\phi$ was reached (it failed in 123 instances).

## Experiment 3: Clustering random points in an $m$-dimensional box

We took $n$ random points in $\mathbb{R}^m$ (normal distribution), with $n \in \{500, 1000, 5000, 10000\}$ and $m \in \{2, 5, 10, 20\}$. For each $(n, m)$, 10 random starting points were chosen. Then:

- BDCA was run to solve the $k$-clustering, with $k \in \{5, 10, 15, 20, 25, 50, 75, 100\}$.
- DCA was run until the same value of $\phi$ was reached (it failed in 123 instances).



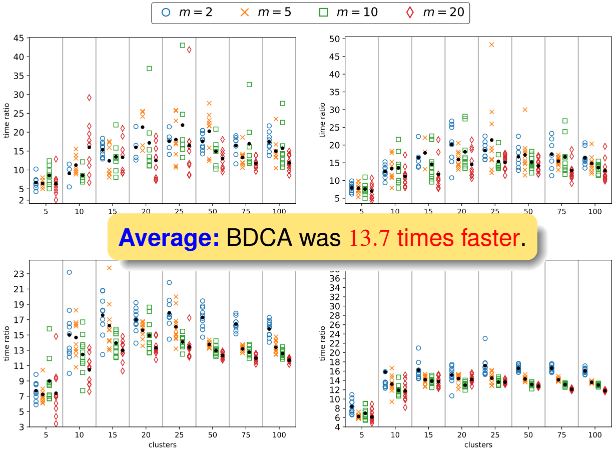
$n = 500$ $n = 1000$

$n = 5000$ $n = 10000$

## Experiment 3: Clustering random points in an $m$-dimensional box

We took $n$ random points in $\mathbb{R}^m$ (normal distribution), with $n \in \{500, 1000, 5000, 10000\}$ and $m \in \{2, 5, 10, 20\}$. For each $(n, m)$, 10 random starting points were chosen. Then:

- BDCA was run to solve the $k$-clustering, with $k \in \{5, 10, 15, 20, 25, 50, 75, 100\}$.
- DCA was run until the same value of $\phi$ was reached (it failed in 123 instances).



**Average:** BDCA was 13.7 times faster.

$n = 5000$ $n = 10000$

## Outline

## The Multidimensional Scaling Problem

Suppose that we have a table of distances between some objects, known as the dissimilarity matrix. If the objects are $n$ points $x^1, x^2, \dots, x^n$ in $\mathbb{R}^q$, the dissimilarity matrix can be defined by the Euclidean distance between them:

$$\delta_{ij} = \|x^i - x^j\| := \mathsf{d}_{ij}(X),$$

where we denote by $X$ the $n \times q$ matrix whose rows are $x^1, x^2, \dots, x^n$.

## The Multidimensional Scaling Problem

Suppose that we have a table of distances between some objects, known as the dissimilarity matrix. If the objects are $n$ points $x^1, x^2, \dots, x^n$ in $\mathbb{R}^q$, the dissimilarity matrix can be defined by the Euclidean distance between them:

$$\delta_{ij} = \|x^i - x^j\| := \mathsf{d}_{ij}(X),$$

where we denote by $X$ the $n \times q$ matrix whose rows are $x^1, x^2, \dots, x^n$.
Given a target dimension $p \leq q$, the MDS problem consists in finding $n$ points in $\mathbb{R}^p$, which are represented by an $n \times p$ matrix $X^*$, such that

$$Stress(X^*) := \sum_{i<j} w_{ij} \left(\mathsf{d}_{ij}(X^*) - \delta_{ij}\right)^2$$

is smallest, where $w_{ij} \geq 0$ are weights ($w_{ij} = 0$ if $\delta_{ij}$ is missing).

## The Multidimensional Scaling Problem

Suppose that we have a table of distances between some objects, known as the dissimilarity matrix. If the objects are $n$ points $x^1, x^2, \dots, x^n$ in $\mathbb{R}^q$, the dissimilarity matrix can be defined by the Euclidean distance between them:

$$\delta_{ij} = \|x^i - x^j\| := \mathsf{d}_{ij}(X),$$

where we denote by $X$ the $n \times q$ matrix whose rows are $x^1, x^2, \dots, x^n$.
Given a target dimension $p \leq q$, the MDS problem consists in finding $n$ points in $\mathbb{R}^p$, which are represented by an $n \times p$ matrix $X^*$, such that

$$Stress(X^*) := \sum_{i<j} w_{ij} \left(\mathsf{d}_{ij}(X^*) - \delta_{ij}\right)^2$$

is smallest, where $w_{ij} \geq 0$ are weights ($w_{ij} = 0$ if $\delta_{ij}$ is missing).
This can be equivalently formulated as a DC problem by setting

$$g(X) := \frac{1}{2} \sum_{i<j} w_{ij} \mathsf{d}_{ij}^2(X) + \frac{\rho}{2}\|X\|^2,$$

$$h(X) := \sum_{i<j} w_{ij} \delta_{ij} \mathsf{d}_{ij}(X) + \frac{\rho}{2}\|X\|^2,$$

for some $\rho \geq 0$. Moreover, it is clear that $g$ is differentiable while $h$ is not, but $\partial h$ can be explicitly computed.

## **Experiment 4:** MDS for Spanish cities

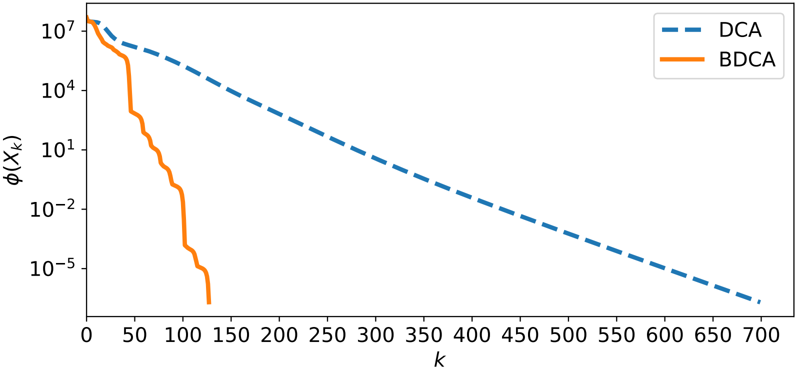
Consider the dissimilarity matrix defined by the distances between the 4155 Spanish cities with more than 500 residents, including this time those outside the peninsula. $\Rightarrow$ The optimal value of this MDS problem is zero.

## **Experiment 4:** MDS for Spanish cities

Consider the dissimilarity matrix defined by the distances between the 4155 Spanish cities with more than 500 residents, including this time those outside the peninsula. $\Rightarrow$ The optimal value of this MDS problem is zero.

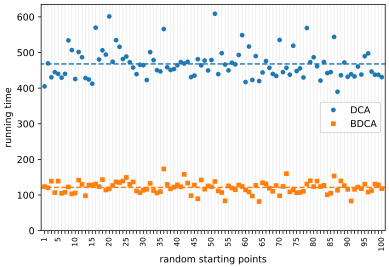
## Experiment 4: MDS for Spanish cities

Consider the dissimilarity matrix defined by the distances between the 4155 Spanish cities with more than 500 residents, including this time those outside the peninsula. ⇒ The optimal value of this MDS problem is zero.



## **Experiment 4:** MDS for Spanish cities

Consider the dissimilarity matrix defined by the distances between the 4155 Spanish cities with more than 500 residents, including this time those outside the peninsula. $\Rightarrow$ The optimal value of this MDS problem is zero.



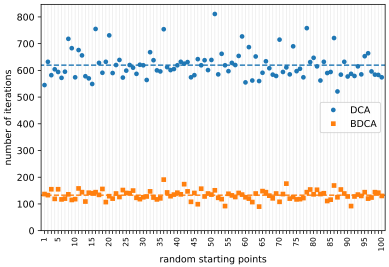
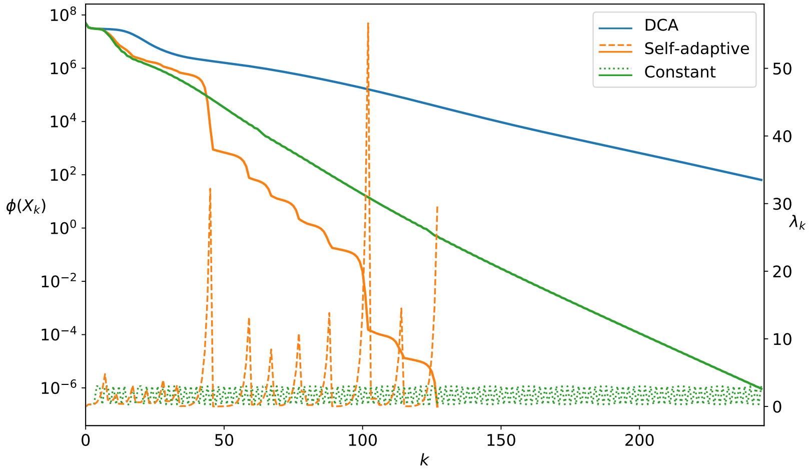


Figure: Running time (left) and number of iterations (right) of DCA and BDCA for 100 random instances.

## **Experiment 4:** MDS for Spanish cities

Consider the dissimilarity matrix defined by the distances between the 4155 Spanish cities with more than 500 residents, including this time those outside the peninsula. $\Rightarrow$ The optimal value of this MDS problem is zero.



## Main References

- H. ATTOUCH, J. BOLTE: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Math. Program.* 116 (2009), 5–16.
- **F.J. ARAGÓN ARTACHO, R.M.T. FLEMING, P.T. VUONG: Accelerating the DC algorithm for smooth functions. *Math. Program. 169B* (2018), 95–118.**
- **F.J. ARAGÓN ARTACHO, P.T. VUONG: The Boosted DC Algorithm for nonsmooth functions. *SIAM J. Optim.* 30 (2020), 980–1006**
- **F.J. ARAGÓN ARTACHO, R. CAMPOY, P.T. VUONG: The boosted dc algorithm for linearly constrained dc programming. *Set-Valued and Variational Analysis* 30 (2022), 1265–1289**
- J. BOLTE, A. DANIILIDIS, A. LEWIS, AND M. SHIOTA, Clarke subgradients of stratifiable functions, *SIAM J. Optim.* 18 (2007), 556–572.
- M. FUKUSHIMA, H. MINE: A generalized proximal point algorithm for certain non-convex minimization problems. *Int. J. Syst. Sci.* 12 (1981), 989–1000.
- H.A. LE THI, T. PHAM DINH: D.C. programing approach to the multidimensional scaling problem, in *From Local to Global Optimization*, P. Pardalos and P. Varbrand, eds, Kluwer, Dodrecht, 2001, 231–276.
- B. ORDIN, A.M. BAGIROV: A heuristic algorithm for solving the minimum sum-of-squares clustering problems, *J. Glob. Optim.* 61 (2015), 341–361.
- T. PHAM DINH, E.B. SOUAD: Algorithms for solving a class of nonconvex optimization problems. Methods of subgradients. In J.-B. Hiriart-Urruty, editor, *FERMAT Days 85: Mathematics for Optimization*, volume 129 of *North-Holland Mathematics Studies*, pp. 249–271. Elsevier (1986)