Performance analysis of the (boosted) Difference-of-Convex Algorithm (DCA)

Etienne de Klerk

EUROpt 2026 summer school

Linz, July 2026

DC Optimization

DC Optimization

$$f_\star := \inf f(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \in \mathbb{R}^n \tag{DCO}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a *difference of convex (DC)* function,

$$f = f_1 - f_2,$$

and f_1, f_2 are convex functions.

Assumptions

- The functions $f_1 \in \mathcal{F}_{\mu_1,L_1}$, $f_2 \in \mathcal{F}_{\mu_2,L_2}$ for some $\mu_1, \mu_2 \in [0, \infty)$ and $L_1 \in [\mu_1, \infty)$, $L_2 \in [\mu_2, \infty)$.
- $f_\star > -\infty$ is a lower bound on f.

DC decompositions

Exercises:

1. Show that any twice continuously differentiable function on $\mathbb{R}^n$ is the difference of convex functions.
2. Show that, if $f = f_1 - f_2$ with $f_1 \in \mathcal{F}_{\mu_1, L_1}$ and $f_2 \in \mathcal{F}_{\mu_2, L_2}$, then $f \in \mathcal{F}_{\mu_1 - L_2, L_1 - \mu_2}$.

Difference of Convex Algorithm (DCA)

Algorithm 1 Unconstrained DCA

Pick $\mathbf{x}_0 \in \mathbb{R}^n$, $N \in \mathbb{N}$.

For $k = 0, 1, \dots, N-1$ perform the following step:

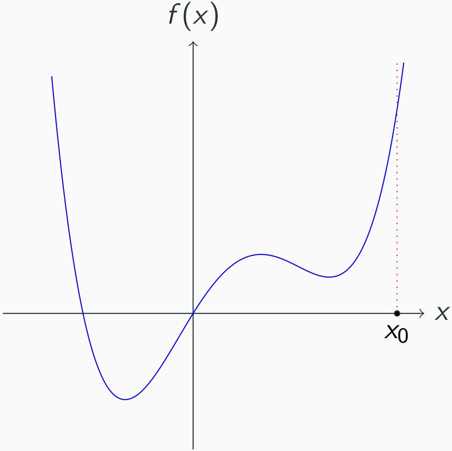
$$\mathbf{x}_{k+1} \in \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} f_1(\mathbf{x}) - \left(f_2(\mathbf{x}_k) + \langle \nabla f_2(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle\right).$$

Surveys:

- Le Thi HA, Dinh TP (2018) DC programming and DCA: thirty years of developments. *Mathematical Programming*. 169(1):5-68.
- Lipp, T., Boyd, S. Variations and extension of the convex–concave procedure. *Optim Eng* 17, 263–287 (2016).

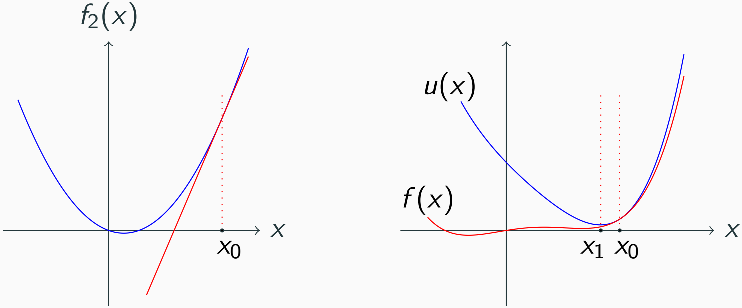
One iteration of DCA

$$\min_{x \in \mathbb{R}} f(x) := \frac{1}{4}x^4 - \frac{2}{3}x^3 - \frac{1}{2}x^2 + 2x, \quad x_0 = 3.$$



One iteration of DCA

$$\min_{x\in\mathbb{R}} u(x) := \underbrace{\left(\tfrac{1}{4}x^4 - \tfrac{2}{3}x^3 + 2x^2\right)}_{f_1} - \underbrace{(16.5 + 13(x-3))}_{f_2(x_0)+f_2'(x_0)(x-x_0)}$$



$$\min_{x\in\mathbb{R}} f(x) := \tfrac{1}{4}x^4 - \tfrac{2}{3}x^3 - \tfrac{1}{2}x^2 + 2x$$

$f(x)$

Exercise:

1. Show that, at each iteration $k = 0, \ldots, N-1$ of DCA:

$$\nabla f_1(\mathbf{x}_{k+1}) = \nabla f_2(\mathbf{x}_k).$$

2. Assume f is L-smooth and consider the D.C. decomposition:

$$f_1(\mathbf{x}) = \frac{L}{2}\|\mathbf{x}\|^2 \in \mathcal{F}_{L,L}(\mathbb{R}^n),\ f_2 = f_1 - f \in \mathcal{F}_{0,2L}(\mathbb{R}^n).$$

   Show that DCA generates the gradient descent iterates:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k).$$

Performance Estimation (PEP)

Performance Estimation Problem (PEP)

$$
\begin{aligned}
\max \ & \left(\min_{0 \le k \le N} \|\nabla f_1(\mathbf{x}_k) - \nabla f_2(\mathbf{x}_k)\|^2 \right) \\
& f_1 \in \mathcal{F}_{\mu_1, L_1}, f_2 \in \mathcal{F}_{\mu_2, L_2} \\
& f_1(\mathbf{x}) - f_2(\mathbf{x}) \ge f_\star \quad \forall \mathbf{x} \in \mathbb{R}^n \\
& f_1(\mathbf{x}_0) - f_2(\mathbf{x}_0) - f_\star \le \Delta \\
& \mathbf{x}_N, \dots, \mathbf{x}_1 \text{ are generated by DCA w.r.t. } f_1, f_2, \mathbf{x}_0 \\
& \mathbf{x}_0 \in \mathbb{R}^n.
\end{aligned}
$$

- *Decision variables:* f_1, f_2 and $\mathbf{x}_0$.
- *Fixed parameters:* $\Delta, \mu_1, L_1, \mu_2, L_2, N$.

Performance Estimation Problem: reformulation

- As before, we replace the conditions $f_1 \in \mathcal{F}_{\mu_1,L_1}, f_2 \in \mathcal{F}_{\mu_2,L_2}$ by a finite set of inequalities using the interpolation conditions....
- ... and introduce (independent) variables $\mathbf{g}_{1,k}$ to correspond to $\nabla f_1(\mathbf{x}_k)$, ...
- ... and $f_{1,k}$ to correspond to $f_1(\mathbf{x}_k)$, etc.
- We also use that $f_1 - f_2$ is $(L_1 - \mu_2)$-smooth.

SDP reformulation of PEP

$$
\begin{aligned}
\max \quad & \left(\min_{0 \le k \le N} \|\mathbf{g}_{1,k} - \mathbf{g}_{2,k}\|^2 \right) \\
\text{s.t.} \quad & \tfrac{1}{2(1-\frac{\mu_\ell}{L_\ell})} \left(\tfrac{1}{L_\ell} \|\mathbf{g}_{\ell,i} - \mathbf{g}_{\ell,j}\|^2 + \mu_\ell \|\mathbf{x}_i - \mathbf{x}_j\|^2 - \tfrac{2\mu_\ell}{L_\ell} \langle \mathbf{g}_{\ell,j} - \mathbf{g}_{\ell,i}, \mathbf{x}_j - \mathbf{x}_i \rangle \right) \le \\
& \qquad f_{\ell,i} - f_{\ell,j} - \langle \mathbf{g}_{\ell,j}, \mathbf{x}_i - \mathbf{x}_j \rangle \quad \forall u, v \in S,\ \ell \in \{1,2\} \\
& \color{red}{\mathbf{g}_{1,k+1} = \mathbf{g}_{2,k}} \quad k \in \{1, \ldots, N\} \\
& f_{1,i} - f_{2,i} - \frac{1}{2(L_1 - \mu_2)} \|\mathbf{g}_{1,i} - \mathbf{g}_{2,i}\|^2 \ge f_\star \quad \forall i \in S \\
& f_{1,0} - f_{2,0} - f_\star \le \Delta,
\end{aligned}
$$

where $S := \{0, \ldots, N\}$, $\mathbf{g}_{\ell,i},\ f_{\ell,i}$ $(\ell \in \{1,2\},\ i \in S)$ are decision variables, and where $\Delta, \mu_1, L_1, \mu_2, L_2, N$ are fixed parameters.

Covergence rate of DCA

Convergence rate: example PEP results

Theorem

Suppose that $f_1 \in \mathcal{F}_{\mu_1,L_1}$ *and* $f_2 \in \mathcal{F}_{\mu_2,L_2}$. *If* $\mu_1 = \mu_2 = 0$, *then after* N *iterations of DCA, one has:*

$$\min_{0\le k\le N} \|\nabla f(\mathbf{x}_k)\|^2 \le \frac{2L_1L_2\left(f(\mathbf{x}_0) - f_\star\right)}{(L_1+L_2)N + L_2}.$$

This bound is tight (computer session).

Case where $\mu_1 = \mu_2 > 0$ and $L_1 = L_2$

Another special case for later comparison:

Theorem

Let $f_1 \in \mathcal{F}_{\mu,L}$ and $f_2 \in \mathcal{F}_{\mu,L}$, $0 \leq \mu < L < \infty$. After N iterations of DCA, one has, for $\kappa = \mu/L \in [0, 1)$:

$$\min_{0 \leq k \leq N} \frac{\|\nabla f(\mathbf{x}_k)\|^2}{L} \leq \frac{f(\mathbf{x}_0) - f_\star}{N + \frac{1}{2(1-\kappa)}}.$$

These results are special cases of Theorem 3.1 in:

H. Abbaszadehpeivasti, E. de Klerk, M. Zamani. On the Rate of Convergence of the Difference-of-Convex Algorithm (DCA). *JOTA* 202, 475–496 (2024).

Boosted DCA

Boosted DCA: basic idea and motivation

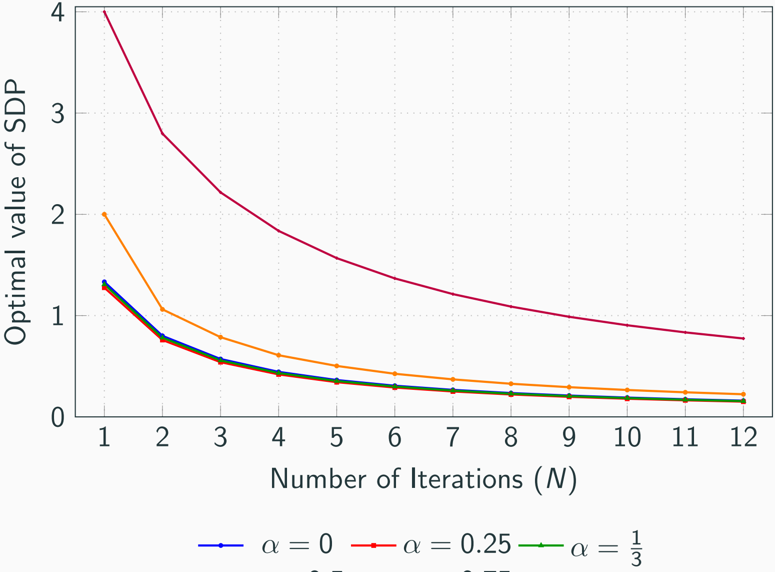
- The idea of boosted DCA is to take an additional step in the DCA direction.
- If $\mathbf{y}_k$ is obtained from $\mathbf{x}_k$ by DCA, then set $\mathbf{x}_{k+1} = \mathbf{y}_k + \alpha(\mathbf{y}_k - \mathbf{x}_k)$ for some fixed $\alpha > 0$.
- Motivation: if f is smooth, then $\mathbf{d} := \mathbf{y}_k - \mathbf{x}_k$ is a descent direction for f at $\mathbf{y}_k$.

cf. Proposition 4 in:

F.J. Aragón Artacho, R.M.T. Fleming, and P.T. Vuong. Accelerating the DC algorithm for smooth functions. *Math. Program.* 169, 95–118 (2018).

- Aragón Artacho et al. also showed improved computational performance on a class of DC problems.
- The boosted DCA may be analysed by PEP in the same way as DCA.

SDP optimal value when $L_1 = L_2 = 2$, and $\mu_1 = 1,\ \mu_2 = 0$.



New results: PEP for boosted DCA

Boosted DCA: performance estimation result

Theorem

Let $f_1 \in \mathcal{F}_{\mu,L}(\mathbb{R}^n)$ and $f_2 \in \mathcal{F}_{\mu,L}(\mathbb{R}^n)$, $0 \leq \mu < L < \infty$. After N iterations of boosted DCA, one has,

$$\min_{0 \leq k \leq N} \frac{\|\nabla f(\mathbf{x}_k)\|^2}{L} \leq \frac{(f(\mathbf{x}_0) - f_\star)}{(1 + \kappa\alpha)N + \frac{1}{2(1-\kappa)}},$$

where $\kappa := \mu/L \in [0,1)$, and where $0 \leq \alpha \leq \min\{1, 2\kappa\}$.

- When $\alpha = 0$ or $\kappa = 0$, this result reduces to the known (tight) convergence result for DCA shown earlier.
- Optimal choice of boosted step: $\alpha = \min\{1, 2\kappa\}$.
- This gives an improvement on DCA if $\mu > 0$.

Remarks

References

The SDP performance estimation results for DCA are from:

Abbaszadehpeivasti, H., de Klerk, E. & Zamani, M. On the Rate of Convergence of the Difference-of-Convex Algorithm (DCA). J Optim Theory Appl 202, 475–496 (2024)

... and for boosted DCA:

Abbaszadehpeivasti, H., de Klerk, E. & Taylor, A. On the convergence rate of the boosted difference-of-convex algorithm (DCA). Optim Lett (2026).

Related work:

Rotaru, Teodor and Patrinos, Panagiotis and Glineur, François (2025) Tight Analysis of Difference-of-Convex Algorithm (DCA) Improves Convergence Rates for Proximal Gradient Descent. arXiv:2503.04486.

Rotaru et al. extend the DCA PEP analysis to allow for negative μ_2.

Computer session

Computer session

- Use Chrome to open (or scan the QR code below): `https://github.com/PerformanceEstimation/Tutorial-Europt-2026`
- Click on Part 3 in left-hand-side menu.
- Click on `Europt2026_Tutorial_Part3.ipynb`
- Click on Open in Colab

The End: Thank you to the organisers!