Locally Accelerated Conditional Gradients

Jelena Diakonikolas [1], Alejandro Carderera [2], Sebastian Pokutta [3, 4]

[1]UC Berkeley, [2]Georgia Institute of Technology, [3]Zuse Institute Berlin, [4]Technische Universität Berlin

AISTATS 2020

Goal is L-smooth μ-strongly convex optimization over polytope $\mathcal{X}$.

$$\min_{x \in \mathcal{X}} f(x)$$

Goal is L-smooth μ-strongly convex optimization over polytope $\mathcal{X}$.

$$\min_{x \in \mathcal{X}} f(x)$$

Main ingredients:

First-order (FO) oracle. Given $x \in \mathcal{X}$ and a differentiable convex function $f : \mathbb{R}^n \to \mathbb{R}$, return:

$$\nabla f(x) \in \mathbb{R}^n \text{ and } f(x) \in \mathbb{R}$$

Linear optimization (LO) oracle. Given $v \in \mathbb{R}^n$, return:

$$\operatorname*{argmin}_{x \in \mathcal{X}} \langle v, x \rangle$$

Goal is L-smooth μ-strongly convex optimization over polytope $\mathcal{X}$.

$$\min_{x \in \mathcal{X}} f(x)$$

Main ingredients:

First-order (FO) oracle. Given $x \in \mathcal{X}$ and a differentiable convex function $f : \mathbb{R}^n \to \mathbb{R}$, return:

$$\nabla f(x) \in \mathbb{R}^n \text{ and } f(x) \in \mathbb{R}$$

Linear optimization (LO) oracle. Given $v \in \mathbb{R}^n$, return:

$$\operatorname*{argmin}_{x \in \mathcal{X}} \langle v, x \rangle$$

Focus on *Conditional Gradients/Frank-Wolfe* algorithm [FW56; Pol74] and its variants such as the *Away-step Conditional Gradients/Frank-Wolfe* (AFW) algorithm [Wol70; GM86].

Away-step Conditional Gradients (AFW)

Choose direction that guarantees more progress:

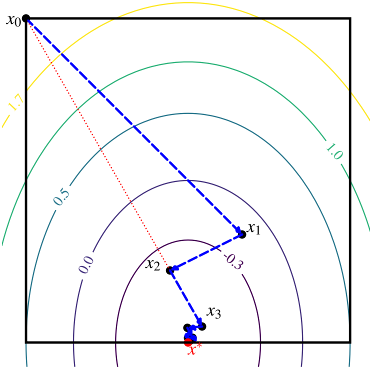


Figure: Away-step CG (AFW)

1. **Frank-Wolfe direction:**

$$\operatorname*{argmin}_{y \in \mathcal{X}} \langle \nabla f(x), y \rangle - x.$$

2. **Away-step direction:**

$$x - \operatorname*{argmax}_{y \in \mathcal{S}} \langle \nabla f(x), y \rangle ,$$

where $\mathcal{S}$ is the *active set* of x.

Convergence rate for L-smooth μ-strongly convex f

Theorem (Convergence rate of AFW)

[LJ15] Suppose that f is L-smooth μ-strongly convex over a polytope $\mathcal{X}$, the number of steps T required to reach an ϵ-optimal solution to the minimization problem satisfies,

$$T = \mathcal{O}\left(\frac{L}{\mu}\left(\frac{D}{\delta}\right)^2 \log\frac{1}{\epsilon}\right),$$

where D and δ are the diameter and pyramidal width of $\mathcal{X}$.

CG Global Acceleration

However, we know that optimal methods for this class of functions achieve an ϵ solution in $T = \mathcal{O}\left(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon}\right)$ first-order calls [NY83; Nes83].

Can CG achieve these convergence rates **globally**?

CG Global Acceleration

However, we know that optimal methods for this class of functions achieve an ϵ solution in $T = \mathcal{O}\left(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon}\right)$ first-order calls [NY83; Nes83].

Can CG achieve these convergence rates **globally**?

Dimension independent global acceleration
is not possible [Jag13; Lan13].

Objectives:

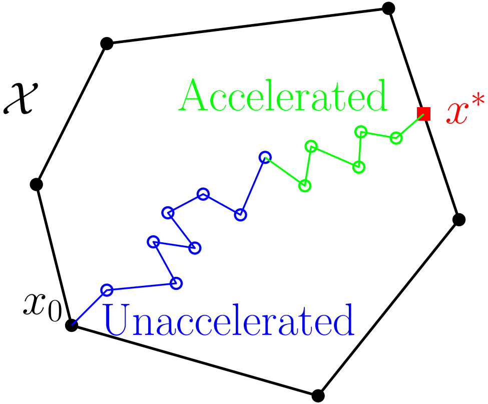
- Dimension independent global acceleration.

Objectives:

- ~~Dimension independent global acceleration.~~
- Dimension independent local acceleration.

Locally Accelerated Conditional Gradients (LaCG)

What do we mean by **local acceleration**?

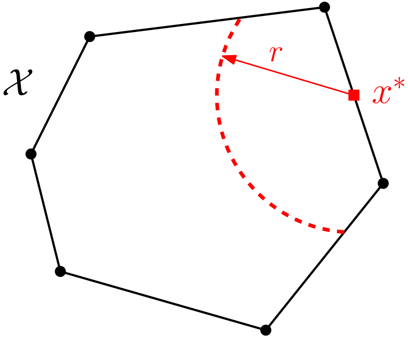


After a constant number of iterations that does not depend on ϵ, accelerate the convergence.

Let $\mathcal{S}_t$ denote the CG active set at iteration t.

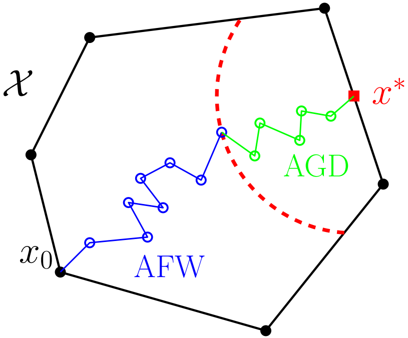
What we know:

$\exists r > 0$ s.t. if $\|x^* - x_T\| \leq r \Rightarrow x^* \in \operatorname{conv}(\mathcal{S}_T)$.

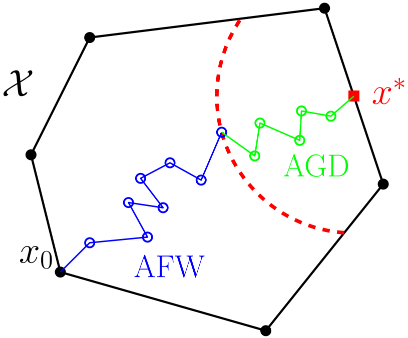


Naive Idea: Run an accelerated first-order method (AGD) on $\operatorname{conv}(\mathcal{S}_T)$.

We would want the following:

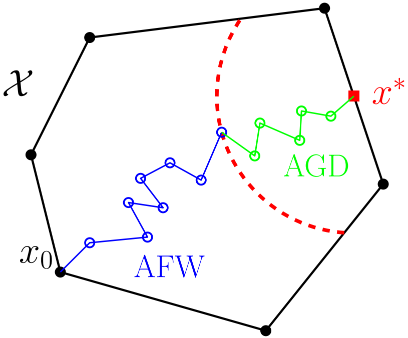


We would want the following:



Problem: The value of r is not known, we don't know when to switch from AFW to AGD.

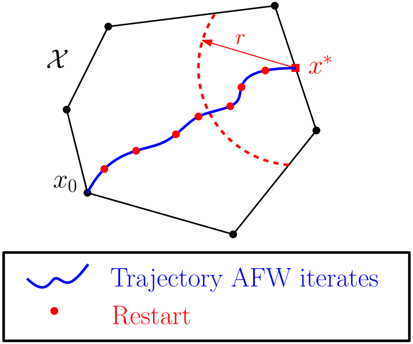
We would want the following:



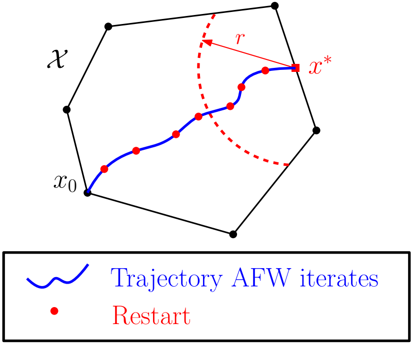
Problem: The value of r is not known, we don't know when to switch from AFW to AGD.

Challenge: Create algorithm that accelerates without knowledge of r.

Run AFW and restart AGD by running it over a new $\mathrm{conv}(\mathcal{S}_t)$ every H iterations.

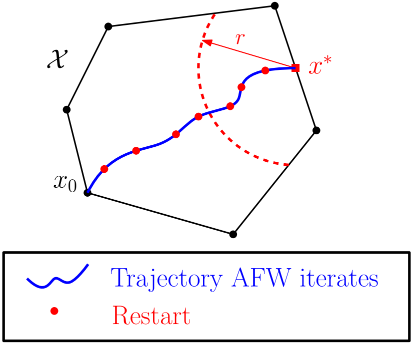


Run AFW and restart AGD by running it over a new $\mathrm{conv}(\mathcal{S}_t)$ every H iterations.



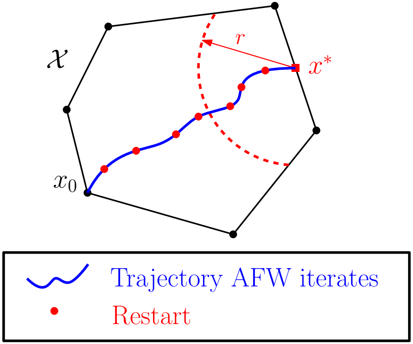
- Every H iterations restart AGD and run it over $\mathrm{conv}(\mathcal{S}_t)$.

Run AFW and restart AGD by running it over a new $\text{conv}(\mathcal{S}_t)$ every H iterations.



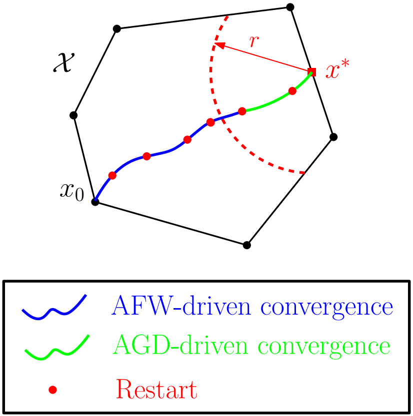
- Every H iterations restart AGD and run it over $\text{conv}(\mathcal{S}_t)$.
- Have AGD and AFW compete for progress at each iteration between restarts.

Run AFW and restart AGD by running it over a new $\operatorname{conv}(\mathcal{S}_t)$ every H iterations.



- Every H iterations restart AGD and run it over $\operatorname{conv}(\mathcal{S}_t)$.
- Have AGD and AFW compete for progress at each iteration between restarts.
- Space out restarts so that you only loose a factor of 2 in the AGD convergence rate.

What we will obtain:



Locally Accelerated Conditional Gradients (LaCG)

Algorithm 1 Locally Accelerated Conditional Gradients

Initialize $\mathcal{C}_0 = \mathcal{S}_0$, $x_0 = x_0^{AFW} = x_0^{AGD}$, $H = \mathcal{O}\left(\sqrt{\frac{L}{\mu}} \log \frac{L}{\mu}\right)$

for $t = 1$ to T **do**

- $x_{t+1}^{AFW}, \mathcal{S}_{t+1} \leftarrow AFW(x_t^{AFW}, \mathcal{S}_t)$ ▷ AFW step
- **if** Vertex has been added to $\mathcal{S}$ since restart **then**
  - **if** $t = Hn$ for some $n \in \mathbb{N}$ **then**
    - $x_{t+1}^{AGD} \leftarrow \operatorname{argmin}_{x \in \{x_t^{AFW}, x_t^{AGD}\}} f(x)$ ▷ Restart AGD
    - $\mathcal{C}_{t+1} \leftarrow$ Update based on previous line.
  - **else**
    - $x_{t+1}^{AGD} \leftarrow AGD(x_t^{AGD}, \mathcal{C}_t)$ ▷ Run AGD decoupled from AFW
    - $\mathcal{C}_{t+1} \leftarrow \mathcal{C}_t$
  - **end if**
- **else**
  - $x_{t+1}^{AGD} \leftarrow AGD(x_t, \mathcal{C}_t)$ ▷ Run AGD coupled with AFW
  - $\mathcal{C}_{t+1} \leftarrow \operatorname{conv}(\mathcal{S}_{t+1})$
- **end if**
- $x_{t+1} \leftarrow \operatorname{argmin}_{x \in \{x_{t+1}^{AFW}, x_{t+1}^{AGD}, x_t\}} f(x)$ ▷ Monotonicity

end for

Analysis relies on the *Approximate Duality Gap* technique [DO19] and the AGD algorithm used is a *Modified μAGD+* algorithm [CDO18; DCP19].

Theorem (Convergence rate of μAGD+.)

Let f be L-smooth and μ-strongly convex and let $\{\mathcal{C}_i\}_{i=0}^{t}$ be a sequence of convex subsets of $\mathcal{X}$ such that $\mathcal{C}_i \subseteq \mathcal{C}_{i-1}$ for all i and $x^* \in \cap_{i=0}^{t} \mathcal{C}_i$, then the μAGD+ achieves an ϵ-optimal solution in a number of iterations T that satisfies:

$$T = \mathcal{O}\left(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon}\right)$$

Convergence rate of LaCG

Theorem (Convergence rate of LaCG)

Let f be L-smooth and μ-strongly convex and let r be the critical radius. The number of steps T required to reach an ϵ-optimal solution to the minimization problem satisfies:

$$t = min\left\{\mathcal{O}\left(\frac{L}{\mu}\left(\frac{D}{\delta}\right)^2 \log\frac{1}{\epsilon}\right), K + \mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\epsilon}\right)\right\},$$

where $K = \frac{8L}{\mu}\left(\frac{D}{\delta}\right)^2 \log\left(\frac{2(f(x_0) - f^*)}{\mu r^2}\right)$.

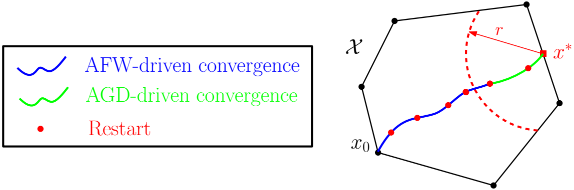
Convergence rate of LaCG

Theorem (Convergence rate of LaCG)

Let f be L-smooth and μ-strongly convex and let r be the critical radius. The number of steps T required to reach an ϵ-optimal solution to the minimization problem satisfies:

$$t = min\left\{\mathcal{O}\left(\frac{L}{\mu}\left(\frac{D}{\delta}\right)^2 \log\frac{1}{\epsilon}\right), K + \mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\epsilon}\right)\right\},$$

where $K = \frac{8L}{\mu}\left(\frac{D}{\delta}\right)^2 \log\left(\frac{2(f(x_0)-f^*)}{\mu r^2}\right)$.



Computational Results

Despite the faster convergence rate after the burn-in phase, how does LaCG perform with respect to other projection-free algorithms?

Simplex in $\mathbb{R}^{1500}$ with $L/\mu = 1000$

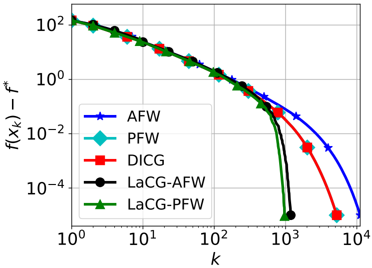


Figure: Primal gap vs. iteration

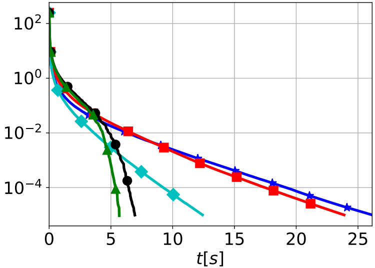


Figure: Primal gap vs. time

When close enough to x^* (after burn-in phase), there is a significant speedup in the convergence rate.

Birkhoff polytope in $\mathbb{R}^{400x400}$ with $L/\mu = 100$

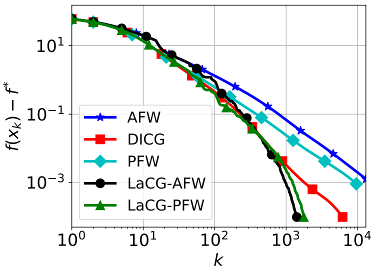


Figure: Primal gap vs. iteration

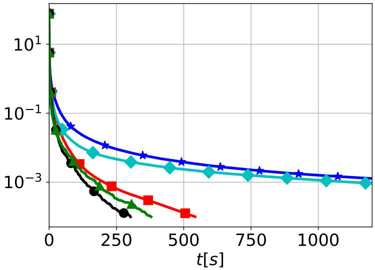


Figure: Primal gap vs. time

Structured Regression over MIPLIB Polytope (ran14x18-disj-8)

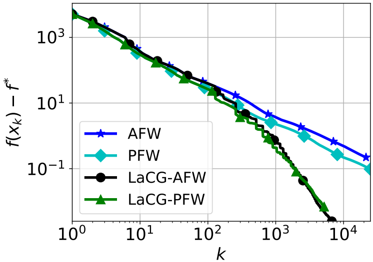


Figure: Primal gap vs. iteration

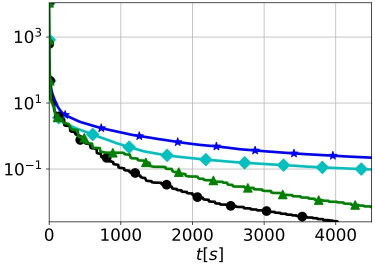


Figure: Primal gap vs. time

Thank you
for your attention.

References I

[FW56] Marguerite Frank and Philip Wolfe. "An algorithm for quadratic programming". In: *Naval research logistics quarterly* 3.1-2 (1956), pp. 95–110.

[Pol74] Boris Teodorovich Polyak. "Minimization methods in the presence of constraints". In: *Itogi Nauki i Tekhniki. Seriya" Matematicheskii Analiz"* 12 (1974), pp. 147–197.

[Wol70] Philip Wolfe. "Convergence theory in nonlinear programming". In: *Integer and nonlinear programming* (1970), pp. 1–36.

[GM86] Jacques Guélat and Patrice Marcotte. "Some comments on Wolfe's 'away step'". In: *Mathematical Programming* 35.1 (1986), pp. 110–119.

References II

[LJ15] Simon Lacoste-Julien and Martin Jaggi. "On the Global Linear Convergence of Frank-Wolfe Optimization Variants". In: *Advances in Neural Information Processing Systems 28*. 2015, pp. 496–504.

[NY83] Arkadii Semenovich Nemirovsky and David Borisovich Yudin. "Problem complexity and method efficiency in optimization". In: *Wiley-Interscience Series in Discrete Mathematics* 15 (1983).

[Nes83] Y Nesterov. "A method of solving a convex programming problem with convergence rate $O(\frac{1}{k^2})$". In: *Soviet Math. Dokl*. Vol. 27. 1983.

[Jag13] Martin Jaggi. "Revisiting Frank-Wolfe: Projection-free sparse convex optimization.". In: *ICML (1)*. 2013, pp. 427–435.

References III

[Lan13] G Lan. "The complexity of large-scale convex programming under a linear optimization oracle". In: *Technical report, Department of Industrial and Systems Engineering, University of Florida.* (2013).

[DO19] Jelena Diakonikolas and Lorenzo Orecchia. "The approximate duality gap technique: A unified theory of first-order methods". In: *SIAM Journal on Optimization* 29.1 (2019), pp. 660–689.

[CDO18] Michael B Cohen, Jelena Diakonikolas, and Lorenzo Orecchia. "On acceleration with noise-corrupted gradients". In: *35nd International Conference on Machine Learning, ICML 2018* (2018).

[DCP19] Jelena Diakonikolas, Alejandro Carderera, and Sebastian Pokutta. "Locally Accelerated Conditional Gradients". In: *arXiv preprint arXiv:1906.07867* (2019).

References IV

[LZ16] Guanghui Lan and Yi Zhou. "Conditional gradient sliding for convex optimization". In: *SIAM Journal on Optimization* 26.2 (2016), pp. 1379–1409.

[LMH15] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. "A universal catalyst for first-order optimization". In: *Advances in neural information processing systems*. 2015, pp. 3384–3392.

Lower bound on number of iterations.

Can CG achieve these convergence rates **globally**?

Example ([Lan13; Jag13] $f(x) = \|x\|^2$ over unit simplex in $\mathbb{R}^n$.)

We know the optimal solution is given by $x^* = \mathbb{1}/n$. CG can incorporate at most one vertex in each iteration, if we start from a vertex x_0, in iteration $t < n$ we have that:

$$f(x_t) - f(x^*) \geq \frac{1}{t} - \frac{1}{n}.$$

Considering iterations such that $t \leq \lfloor n/2 \rfloor$ and rearranging into a linear convergence contraction we have:

$$T = \Omega\left(\frac{1}{r}\log\frac{1}{\epsilon}\right),$$

where $r \leq 2\frac{\log 2t}{2t}$.

Considering iterations such that $t \leq \lfloor n/2 \rfloor$ and rearranging into a linear convergence contraction we have:

$$T = \Omega\left(\frac{1}{r}\log\frac{1}{\epsilon}\right),$$

where $r \leq 2\frac{\log 2t}{2t}$.

Convergence rate of the CG variants for this problem instance: $r = \frac{1}{4t}$.

At best a global logarithmic improvement in the convergence rate, therefore **global acceleration in Nesterov's sense is not possible**.

Other Acceleration Approaches

Conditional Gradient Sliding (CGS): Run Nesterov's Accelerated Gradient Descent, use CG to solve the projection subproblems approximately [LZ16].

Other Acceleration Approaches

Conditional Gradient Sliding (CGS): Run Nesterov's Accelerated Gradient Descent, use CG to solve the projection subproblems approximately [LZ16].

Catalyst Augmented AFW: Run Accelerated Proximal Method and solve proximal problems with a linearly convergent CG [LMH15].

Other Acceleration Approaches

Conditional Gradient Sliding (CGS): Run Nesterov's Accelerated Gradient Descent, use CG to solve the projection subproblems approximately [LZ16].

Catalyst Augmented AFW: Run Accelerated Proximal Method and solve proximal problems with a linearly convergent CG [LMH15].

Complexity for L-smooth μ-strongly convex f.

Algorithm	LO Calls	FO Calls
CGS	$\mathcal{O}\left(\frac{LD^2}{\epsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\epsilon}\right)$
Catalyst	$\mathcal{O}\left(\sqrt{\frac{L-\mu}{\mu}}\left(\frac{D}{\delta}\right)^2\log\frac{1}{\epsilon}\right)$	$\mathcal{O}\left(\sqrt{\frac{L-\mu}{\mu}}\left(\frac{D}{\delta}\right)^2\log\frac{1}{\epsilon}\right)$

Additional Examples

Congestion Balancing in Traffic Networks

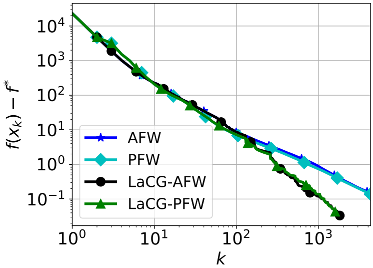


Figure: Primal gap vs. iteration

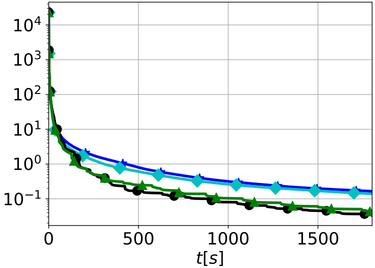


Figure: Primal gap vs. time