A Lower Bound in ADGT

In this section, we provide the construction of the lower bounds on the minimum function value $f(\mathbf{x}^*)$ that are used in our analysis. By μ-strong convexity of f, we have that, $\forall \mathbf{x} \in \mathcal{X}$:

$$f(\mathbf{x}^*) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{x}^* - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^*\|^2. \tag{A.1}$$

Further, if $\mathbf{x}^*$ belongs to the interior of $\mathcal{X}$, then $\nabla f(\mathbf{x}^*) = 0$, and L-smoothness of f implies, $\forall \mathbf{x} \in \mathcal{X}$:

$$f(\mathbf{x}^*) \geq f(\mathbf{x}) - \frac{L}{2}\|\mathbf{x} - \mathbf{x}^*\|^2. \tag{A.2}$$

Let $\{\mathbf{x}_i\}_{i=0}^k$ be a sequence of points from some feasible set $\mathcal{X}$ and let $\{a_i\}_{i=0}^k$ be a sequence of positive numbers with $a_0 = 1$. Define $A_k \stackrel{\text{def}}{=} \sum_{i=0}^k a_i$.

Assume first that $\mathbf{x}^*$ belongs to the interior of the feasible set $\mathcal{X}$. Then, taking a convex combination of Eq. (A.2) with $\mathbf{x} = \mathbf{x}_0$ and Eq. (A.1) with $\mathbf{x} = \mathbf{x}_i$, $1 \leq i \leq k$, we get:

$$\begin{aligned}
f(\mathbf{x}^*) \geq & \frac{\sum_{i=0}^k a_i f(\mathbf{x}_i) + \sum_{i=1}^k a_i(\langle \nabla f(\mathbf{x}_i), \mathbf{x}^* - \mathbf{x}_i \rangle + \frac{\mu}{2}\|\mathbf{x}_i - \mathbf{x}^*\|^2) - \frac{L}{2}\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{A_k} \\
& + \frac{\mu}{2A_k}\|\mathbf{x}_0 - \mathbf{x}^*\|^2 - \frac{\mu}{2A_k}\|\mathbf{x}_0 - \mathbf{x}^*\|^2 \\
\geq & \frac{\sum_{i=0}^k a_i f(\mathbf{x}_i) + \min_{\mathbf{u}\in\mathbb{R}^d}\{\sum_{i=1}^k a_i(\langle \nabla f(\mathbf{x}_i), \mathbf{u} - \mathbf{x}_i \rangle + \frac{\mu}{2}\|\mathbf{x}_i - \mathbf{u}\|^2) + \frac{\mu}{2}\|\mathbf{x}_0 - \mathbf{u}\|^2\}}{A_k} \\
& - \frac{L+\mu}{2A_k}\|\mathbf{x}_0 - \mathbf{x}^*\|^2.
\end{aligned}$$

The last expression corresponds to the lower bound used in the proof of Lemma B.2.

Now assume that $\mathbf{x}^*$ is not necessarily from the interior of $\mathcal{X}$. Take a convex combination (with weights a_i/A_k) of Eq. (A.1) for $\mathbf{x} = \mathbf{x}_i$, $0 \leq i \leq k$. Let $\mathcal{C}_k$ be any convex subset of $\mathcal{X}$ that contains $\mathbf{x}^*$. Then, we have:

$$\begin{aligned}
f(\mathbf{x}^*) \geq & \frac{\sum_{i=0}^k a_i f(\mathbf{x}_i) + \sum_{i=0}^k a_i(\langle \nabla f(\mathbf{x}_i), \mathbf{x}^* - \mathbf{x}_i \rangle + \frac{\mu}{2}\|\mathbf{x}_i - \mathbf{x}^*\|^2)}{A_k} \\
& + \frac{\mu_0}{2A_k}\|\mathbf{x}_0 - \mathbf{x}^*\|^2 - \frac{\mu_0}{2A_k}\|\mathbf{x}_0 - \mathbf{x}^*\|^2 \\
\geq & \frac{\sum_{i=0}^k a_i f(\mathbf{x}_i) + \min_{\mathbf{u}\in\mathcal{C}_k}\{\sum_{i=0}^k a_i(\langle \nabla f(\mathbf{x}_i), \mathbf{u} - \mathbf{x}_i \rangle + \frac{\mu}{2}\|\mathbf{x}_i - \mathbf{u}\|^2) + \frac{\mu_0}{2}\|\mathbf{x}_0 - \mathbf{u}\|^2\}}{A_k} \\
& - \frac{\mu_0}{2A_k}\|\mathbf{x}_0 - \mathbf{x}^*\|^2.
\end{aligned}$$

The last expression corresponds to the lower bound used in the proof of Lemma 3.2.

B Omitted Proofs from Section 3

B.1 Proofs and Results for Warm-up: Optimum in the Interior of $\mathcal{X}$

Starting at point $\mathbf{x}_k$, the Frank-Wolfe step $\mathbf{x}_{k+1}^{\text{FW}}$ is defined via:

$$\begin{aligned}
\mathbf{v}_k &= \operatorname*{argmin}_{\mathbf{u}\in\mathcal{X}} \langle \nabla f(\mathbf{x}_k), \mathbf{u} \rangle, \\
\mathbf{x}_{k+1}^{\text{FW}} &= (1 - \eta_k)\mathbf{x}_k + \eta_k \mathbf{v}_k,
\end{aligned} \tag{B.1}$$

where

$$\eta_k = \operatorname*{argmin}_{\eta\in[0,1]} \left\{ f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \eta_k(\mathbf{v}_k - \mathbf{x}_k) \rangle + \frac{L}{2}{\eta_k}^2\|\mathbf{x}_k - \mathbf{v}_k\|^2 \right\}.$$

On the other hand, the accelerated step $\hat{\mathbf{x}}_{k+1}$ is defined as:

$$\begin{aligned}\mathbf{y}_{k+1} &= \frac{1}{1+\theta}\mathbf{x}_k + \frac{\theta}{1+\theta}\mathbf{w}_k,\\ \mathbf{w}_{k+1} &= (1-\theta)\mathbf{w}_k + \theta\Big(\mathbf{y}_{k+1} - \frac{1}{\mu}\nabla f(\mathbf{y}_{k+1})\Big),\\ \hat{\mathbf{x}}_{k+1} &= (1-\theta)\mathbf{x}_k + \theta\mathbf{w}_{k+1},\end{aligned} \tag{B.2}$$

where $\theta = \sqrt{\frac{\mu}{L}}$ and $\mathbf{w}_k$ and $\mathbf{x}_k$ are appropriately initialized. We now proceed to describe the algorithm.

Algorithm 2 Preliminary Locally Accelerated Frank-Wolfe for $\mathbf{x}^* \in \operatorname{int}(\mathcal{X})$

Input: $\mathbf{x}_0 \in \mathcal{X}$, μ, L, $\mathcal{X}$
Initialization: $\mathbf{w}_0 = \mathbf{x}_0$, $\theta = \sqrt{\mu/L}$
for $k = 0$ to $N-1$ **do**
  Compute $\mathbf{x}_{k+1}^{\mathrm{FW}}$ based on Eq. (B.1) and $\hat{\mathbf{x}}_{k+1}$ based on Eq. (B.2)
  if $\hat{\mathbf{x}}_{k+1} \in \mathcal{X}$ **then**
    $\mathbf{x}_{k+1} = \operatorname{argmin}\{f(\mathbf{x}_{k+1}^{\mathrm{FW}}),\ f(\hat{\mathbf{x}}_{k+1})\}$
  else
    $\mathbf{x}_{k+1} = \mathbf{x}_{k+1}^{\mathrm{FW}}$
    $\mathbf{w}_{k+1} = \mathbf{x}_{k+1}$

Note that the "else" branch in Algorithm 2 effectively restarts the accelerated sequence.

Let us now argue about the convergence of the algorithm. Observe first that the algorithm makes at least as much progress as Frank-Wolfe, since, whatever the step is, $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_{k+1}^{\mathrm{FW}})$. We thus have the following simple proposition, which bounds the length of the so-called *burn-in phase*.

Proposition B.1. *Assume that $r > 0$. Then, after at most $K_0 = \lfloor \frac{LD^2}{\mu r^2} \rfloor$ steps of Algorithm 2, $f(\mathbf{x}_{K_0}) - f(\mathbf{x}^*) \leq 2\mu r^2$. Further, in every subsequent iteration $k \geq K_0$, $\|\mathbf{x}_k - \mathbf{x}^*\| \leq 2r$.*

Proof. As in every iteration the algorithm makes at least as much progress as standard Frank-Wolfe (since $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_{k+1}^{\mathrm{FW}})$), by standard Frank-Wolfe guarantees (see e.g., Jaggi (2013)), we have that after K_0 steps $f(\mathbf{x}_{K_0}) - f(\mathbf{x}^*) \leq \frac{2LD^2}{K_0+4}$, which gives the first part of the lemma. Since none of the iterations of the algorithm can increase the function value, we have that in every subsequent iteration $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq 2\mu r^2$. By strong convexity and $\nabla f(\mathbf{x}^*) = 0$, this implies $\|\mathbf{x}_k - \mathbf{x}^*\| \leq 2r$. □

We can conclude that if $r > 0$, for $k > K_0 = \lfloor \frac{LD^2}{\mu r^2} \rfloor$ Algorithm 2 never enters the else branch, as $\mathcal{B}(\mathbf{x}^*, 2r) \cap \operatorname{aff}(\mathcal{X}) \subseteq \mathcal{X}$. This is precisely what allows us to obtain accelerated convergence in the remaining iterations. This is formally established by the following lemma.

Lemma B.2. *Assume that $r > 0$ and let $K_0 = \lfloor \frac{LD^2}{\mu r^2} \rfloor$. Then, for all $k \geq K_0$:*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq 2\frac{L+\mu}{\mu} r^2 \left(1 - \sqrt{\frac{\mu}{L}}\right)^{k-K_0}.$$

Proof. Let $k_0 \leq K_0$ be the last iteration in which Algorithm 2 enters the "else" branch – as already argued, for $k > K_0$, this cannot happen. Then, from Algorithm 2, we have that $\mathbf{w}_{k_0} = \mathbf{x}_{k_0}$, and for all iterations $k \geq k_0 + 1$:

$$\begin{aligned}\mathbf{y}_k &= \frac{1}{1+\theta}\mathbf{x}_{k-1} + \frac{\theta}{1+\theta}\mathbf{w}_{k-1},\\ \mathbf{w}_k &= (1-\theta)\mathbf{w}_{k-1} + \theta\Big(\mathbf{y}_k - \frac{1}{\mu}\nabla f(\mathbf{y}_k)\Big),\\ \hat{\mathbf{x}}_k &= (1-\theta)\mathbf{x}_{k-1} + \theta\mathbf{w}_k,\\ \mathbf{x}_k &= \operatorname{argmin}\{f(\hat{\mathbf{x}}_k),\ f(\mathbf{x}_k^{\mathrm{FW}})\}.\end{aligned} \tag{B.3}$$

To analyze the convergence of (B.3), we use the approximate duality gap technique, as described in Section 2.2. Let $a_{k_0} = A_{k_0} = 1$ and $A_k = \sum_{i=k_0}^k a_i$, $\frac{a_k}{A_k} = \theta$ for $k \geq k_0 + 1$. Recall that the approximate duality gap G_k is defined as the difference between a lower bound on $f(\mathbf{x}^*)$, L_k and an upper bound on the algorithm output, U_k. Define $U_k = f(\mathbf{x}_k)$ and L_k via (see Appendix A):

$$\begin{aligned} L_k \stackrel{\text{def}}{=} & \frac{\sum_{i=k_0}^k a_i f(\mathbf{y}_i) + \min_{\mathbf{u}\in\mathbb{R}^d}\{\sum_{i=k_0+1}^k a_i(\langle \nabla f(\mathbf{y}_i), \mathbf{u} - \mathbf{y}_i\rangle + \frac{\mu}{2}\|\mathbf{u} - \mathbf{y}_i\|^2) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{x}_{k_0}\|^2\}}{A_k} \\ & - \frac{L+\mu}{2A_k}\|\mathbf{x}^* - \mathbf{x}_{k_0}\|^2. \end{aligned} \tag{B.4}$$

We claim that:

$$\begin{aligned} \mathbf{w}_k &= \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^d}\Big\{\sum_{i=k_0+1}^k a_i(\langle \nabla f(\mathbf{y}_i), \mathbf{u} - \mathbf{y}_i\rangle + \frac{\mu}{2}\|\mathbf{u} - \mathbf{y}_i\|^2) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{x}_{k_0}\|^2\Big\} \\ &= \frac{\mathbf{x}_{k_0} + \sum_{i=k_0+1}^k a_i(\mathbf{y}_i - \frac{1}{\mu}\nabla f(\mathbf{y}_i))}{A_k}. \end{aligned} \tag{B.5}$$

Indeed, Eq. (B.5) implies that $\mathbf{w}_{k_0} = \mathbf{x}_{k_0}$, while for $k > k_0$ it gives: $A_k\mathbf{w}_k = A_{k-1}\mathbf{w}_{k-1} + a_k(\mathbf{y}_k - \frac{1}{\mu}\nabla f(\mathbf{y}_k))$. As $A_k = A_{k-1} + a_k$ and $\frac{a_k}{A_k} = \theta$, (B.5) implies that $\mathbf{w}_k = (1-\theta)\mathbf{w}_{k-1} + \theta(\mathbf{y}_k - \frac{1}{\mu}\nabla f(\mathbf{y}_k))$, which is equivalent to the definition from Eq. (B.3).

Further, observe from (B.3) that $\mathbf{x}_{k-1} = (1+\theta)\mathbf{y}_k - \theta\mathbf{w}_{k-1}$, which, combined with $\mathbf{w}_k = (1-\theta)\mathbf{w}_{k-1} + \theta(\mathbf{y}_k - \frac{1}{\mu}\nabla f(\mathbf{y}_k))$ and $\theta = \sqrt{\mu/L}$, implies

$$\hat{\mathbf{x}}_k = \mathbf{y}_k - \frac{1}{L}\nabla f(\mathbf{y}_k). \tag{B.6}$$

The rest of the proof bounds the initial gap G_{k_0} and shows that for $k > k_0$, $G_k \leq (1-\theta)G_{k-1}$. Note that, by construction, $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq G_k$.

The initial gap equals $G_{k_0} = \frac{L+\mu}{2}\|\mathbf{x}^* - \mathbf{x}_{k_0}\|^2$. This follows by simply evaluating $U_{k_0} - L_{k_0}$.

Now let $k > k_0$. As $f(\mathbf{x}_k) \leq f(\hat{\mathbf{x}}_k)$ and using (B.6):

$$\begin{aligned} A_kU_k - A_{k-1}U_{k-1} &\leq A_k f(\hat{\mathbf{x}}_k) - A_{k-1}f(\mathbf{x}_{k-1}) \\ &= a_k f(\mathbf{y}_k) + A_k(f(\hat{\mathbf{x}}_k) - f(\mathbf{y}_k)) + A_{k-1}(f(\mathbf{y}_k) - f(\mathbf{x}_{k-1})) \\ &\leq a_k f(\mathbf{y}_k) - \frac{A_k}{2L}\|\nabla f(\mathbf{y}_k)\|^2 + A_{k-1}(f(\mathbf{y}_k) - f(\mathbf{x}_{k-1})). \end{aligned} \tag{B.7}$$

To bound the change in the lower bound, denote by:

$$m_k(\mathbf{u}) = \sum_{i=k_0+1}^k a_i(\langle \nabla f(\mathbf{y}_i), \mathbf{u} - \mathbf{y}_i\rangle + \frac{\mu}{2}\|\mathbf{u} - \mathbf{y}_i\|^2) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{x}_{k_0}\|^2$$

the function inside the minimum in the definition of L_k. Hence:

$$m_k(\mathbf{w}_k) = m_{k-1}(\mathbf{w}_k) + a_k\langle \nabla f(\mathbf{x}_k), \mathbf{w}_k - \mathbf{y}_k\rangle + a_k\frac{\mu}{2}\|\mathbf{w}_k - \mathbf{y}_k\|^2.$$

As $\mathbf{w}_{k-1}$ minimizes $m_{k-1}(\cdot)$, expanding $m_{k-1}(\mathbf{w}_k)$ around $\mathbf{w}_{k-1}$, we have:

$$m_{k-1}(\mathbf{w}_k) = m_{k-1}(\mathbf{w}_{k-1}) + \langle \nabla m_{k-1}(\mathbf{w}_{k-1}), \mathbf{w}_k - \mathbf{w}_{k-1}\rangle + \frac{A_{k-1}\mu}{2}\|\mathbf{w}_k - \mathbf{w}_{k-1}\|^2,$$

leading to:

$$\begin{aligned} m_k(\mathbf{w}_k) - m_{k-1}(\mathbf{w}_{k-1}) &= a_k\langle \nabla f(\mathbf{y}_k), \mathbf{w}_k - \mathbf{y}_k\rangle + a_k\frac{\mu}{2}\|\mathbf{w}_k - \mathbf{y}_k\|^2 + \frac{A_{k-1}\mu}{2}\|\mathbf{w}_k - \mathbf{w}_{k-1}\|^2 \\ &\geq a_k\langle \nabla f(\mathbf{y}_k), \mathbf{w}_k - \mathbf{y}_k\rangle + \frac{A_k\mu}{2}\Big\|\mathbf{w}_k - \frac{A_{k-1}}{A_k}\mathbf{w}_{k-1} - \frac{a_k}{A_k}\mathbf{y}_k\Big\|^2, \end{aligned}$$

where the second line is by Jensen's inequality. As $\frac{a_k}{A_k} = \theta = \sqrt{\mu/L}$, using the definition of $\mathbf{w}_k$, we have:

$$m_k(\mathbf{w}_k) - m_{k-1}(\mathbf{w}_{k-1}) \geq a_k \langle \nabla f(\mathbf{y}_k), \mathbf{w}_k - \mathbf{y}_k \rangle + \frac{A_k}{2L} \|\nabla f(\mathbf{y}_k)\|^2.$$

Combining with the definition of L_k, we thus have:

$$A_k L_k - A_{k-1} L_{k-1} \geq a_k f(\mathbf{y}_k) + a_k \langle \nabla f(\mathbf{y}_k), \mathbf{w}_k - \mathbf{y}_k \rangle + \frac{A_k}{2L} \|\nabla f(\mathbf{y}_k)\|^2. \tag{B.8}$$

Combining (B.7) and (B.8), we have:

$$\begin{aligned} A_k G_k - A_{k-1} G_{k-1} &\leq A_{k-1}(f(\mathbf{y}_k) - f(\mathbf{x}_{k-1})) - a_k \langle \nabla f(\mathbf{y}_k), \mathbf{w}_k - \mathbf{y}_k \rangle - \frac{A_k}{L} \|\nabla f(\mathbf{y}_k)\|^2 \\ &\leq \langle \nabla f(\mathbf{y}_k), A_k \mathbf{y}_k - A_{k-1} \mathbf{x}_{k-1} - a_k \mathbf{w}_k \rangle - \frac{A_k}{L} \|\nabla f(\mathbf{y}_k)\|^2 \\ &= A_k \langle \nabla f(\mathbf{y}_k), \mathbf{y}_k - \hat{\mathbf{x}}_k \rangle - \frac{A_k}{L} \|\nabla f(\mathbf{y}_k)\|^2 \\ &= 0, \end{aligned}$$

where the second line is by convexity of f (namely, by $f(\mathbf{y}_k) - f(\mathbf{x}_k) \leq \langle \nabla f(\mathbf{y}_k), \mathbf{y}_k - \mathbf{x}_k \rangle$), the third line is by the definition of $\hat{\mathbf{x}}_k$ and $\theta = \frac{a_k}{A_k}$, and the last line is by (B.6).

As $\frac{A_{k-1}}{A_k} = 1 - \theta$, we have that $G_k \leq (1-\theta)^{k-k_0} G_{k_0} = (1-\theta)^{k-k_0} \frac{L+\mu}{2} \|\mathbf{x}^* - \mathbf{x}_{k_0}\|^2$, and, thus:

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \left(1 - \sqrt{\frac{\mu}{L}}\right)^{k-k_0} \frac{L+\mu}{2} \|\mathbf{x}^* - \mathbf{x}_{k_0}\|^2.$$

By the same arguments as in the proof of Proposition B.1, $f(\mathbf{x}_{k_0}) - f(\mathbf{x}^*) \leq \frac{2LD^2}{k_0+4}$. By strong convexity of f, this implies that also $\mu \|\mathbf{x}_{k_0} - \mathbf{x}^*\|^2 \leq \frac{4LD^2}{k_0+4}$. To complete the proof, it remains to argue that $\left(1 - \sqrt{\frac{\mu}{L}}\right)^{K_0-k_0} \mu \|\mathbf{x}_{k_0} - \mathbf{x}^*\|^2 \leq \left(1 - \sqrt{\frac{\mu}{L}}\right)^{K_0-k_0} \frac{4LD^2}{k_0+4} \leq 4r^2$. This simply follows by arguing that for the choice of k_0 from the statement of the lemma and $k_0 \leq K_0$, we have $\left(1 - \sqrt{\frac{\mu}{L}}\right)^{K_0-k_0} \frac{1}{k_0+4} \leq \frac{1}{K_0+4}$, while the rest follows from Proposition B.1. This is not hard to show and is omitted. □

Finally, we have the following bound on the convergence of Algorithm 2.

Theorem 3.1. *Let $\mathbf{x}_k$ be the solution output by Algorithm 2 (Appendix B.1) for $k \geq 1$. If:*

$$k \geq \min\left\{ \frac{2LD^2}{\epsilon}, \frac{LD^2}{\mu r^2} + \sqrt{\frac{L}{\mu}} \log\left(\frac{2(L+\mu)r^2}{\mu\epsilon}\right) \right\},$$

then $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$.

Proof. Follows directly by applying the standard convergence bound for FW, Proposition B.1, and Lemma B.2. □

Note that in the argument in Proposition B.1 we could have also used the Away-Step Frank-Wolfe algorithm achieving linear convergence for the burn-in phase. However, for the easy of exposition we used the simpler bound for the warm-up; we will use the Away-Step Frank-Wolfe algorithm in Section 3.2.

B.2 Proofs and Results for Optimum in the Relative Interior of a Face of $\mathcal{X}$ from Section 3.2

In this section we provide full technical details for the results in Section 3.2 and we also restate material from that section here once again to facilitate reading.

We will now formulate the general case that subsumes the case from above. We assume that, given points $\mathbf{x}_1, ..., \mathbf{x}_m$ and a point $\mathbf{y}$, the following problem is easily solvable:

$$\min_{\substack{\mathbf{u} = \sum_{i=1}^m \lambda_i \mathbf{x}_i, \\ \boldsymbol{\lambda} \in \Delta_m}} \frac{1}{2} \|\mathbf{u} - \mathbf{y}\|^2. \tag{3.1}$$

In other words, we assume that the projection onto the convex hull of a given set of vertices can be implemented efficiently; however, we do not require access to a membership oracle anymore. Solving this problem amounts to minimizing a quadratic function over the probability simplex. The size of the program m from Eq. (3.1) corresponds to the size of the active set of the CG-type method employed within LaCG. Note that m is never larger than the iteration count k, and is often much lower than the dimension of the original problem. Further, there exist multiple heuristics for keeping the size of the active set small in practice (see, e.g., Braun et al. (2017)). The projection from Eq. (3.1) does not require access to either the first-order oracle or the linear optimization oracle. Finally, due to Lemma 3.2 stated below, we only need to solve this problem to accuracy of the order $\frac{\epsilon}{\sqrt{\mu L}}$, where ϵ is the target accuracy of the program.

For simplicity, we illustrate the framework using AFW as the coupled CG method. However, the same ideas can be applied to other active-set-based methods such as PFW in a straightforward manner. Unlike in the previous subsection, the assumption that $\mathcal{X}$ is a polytope is crucial here, as the linear convergence for the AFW algorithm established in Lacoste-Julien and Jaggi (2015) relies on a constant, the *pyramidal width*, that is only known to be bounded away from 0 for polytopes. For completeness, we provide the pseudocode for one iteration of AFW (as stated in Lacoste-Julien and Jaggi (2015)) in Algorithm 3 below. In the following, the vector $\boldsymbol{\lambda}_k \in \Delta_m$ with $m = |\mathcal{S}_k|$ denotes the barycentric coordinates of the current iterate $\mathbf{x}_k$ over the active set $\mathcal{S}_k$.

Algorithm 3 Away-Step Frank-Wolfe Iteration: AFW($\boldsymbol{\lambda}, \mathcal{S}, \mathbf{x}$)

Set FW direction: $\mathbf{s} = \operatorname{argmin}_{\mathbf{u}\in\mathcal{X}} \langle \nabla f(\mathbf{x}), \mathbf{u}\rangle$, $\mathbf{d}^{\mathrm{FW}} = \mathbf{s} - \mathbf{x}$
Set Away direction: $\mathbf{v} = \operatorname{argmax}_{\mathbf{u}\in\mathcal{S}} \langle \nabla f(\mathbf{x}), \mathbf{u}\rangle$, $\mathbf{d}^{\mathrm{A}} = \mathbf{x} - \mathbf{v}$
if $\left\langle -\nabla f(\mathbf{x}), \mathbf{d}^{\mathrm{FW}}\right\rangle \geq \left\langle -\nabla f(\mathbf{x}), \mathbf{d}^{\mathrm{A}}\right\rangle$ **then**
    $\mathbf{d} = \mathbf{d}^{\mathrm{FW}}$, $\gamma_{\max} = 1$
else
    $\mathbf{d} = \mathbf{d}^{\mathrm{A}}$, $\gamma_{\max} = \frac{\boldsymbol{\lambda}(\mathbf{v})}{1-\boldsymbol{\lambda}(\mathbf{v})}$
$\gamma' = \operatorname{argmin}_{\gamma\in[0,\gamma_{\max}]} f(\mathbf{x} + \gamma\mathbf{d})$
$\mathbf{x}' = \mathbf{x} + \gamma'\mathbf{d}$; update $\boldsymbol{\lambda}$ (to $\boldsymbol{\lambda}'$)
$\mathcal{S}' = \{\mathbf{u} \in \mathcal{S} \cup \{\mathbf{s}\} : \boldsymbol{\lambda}'(\mathbf{u}) > 0\}$
return $\mathbf{x}'$, $\mathcal{S}'$, $\boldsymbol{\lambda}'$

We will need the following fact that establishes the existence of a radius r (and hence iteration K_r) from which onwards all active sets $\mathcal{S}_k$ maintained by our algorithm ensure that $\mathbf{x}^* \in \mathrm{co}(\mathcal{S}_k)$ for all $k \geq K_r$.

Fact B.3 (Active set convergence)**.** *There exists $r > 0$ such that for any subset $\mathcal{S} \subseteq \mathrm{vert}(\mathcal{X})$ and point $\mathbf{x} \in \mathcal{X}$ with $\mathbf{x} \in \mathrm{co}(\mathcal{S})$ and $\|\mathbf{x} - \mathbf{x}^*\| \leq r$ it follows $\mathbf{x}^* \in \mathrm{co}(\mathcal{S})$.*

Proof. Let $\mathcal{S} \subseteq \mathrm{vert}(\mathcal{X})$ be an arbitrary subset of vertices, so that $\mathbf{x}^* \notin \mathrm{co}(\mathcal{S})$. As $\mathcal{S}$ is closed, there exists $0 < r_{\mathcal{S}} \stackrel{\mathrm{def}}{=} \min_{\mathbf{x}\in\mathcal{S}} \|\mathbf{x} - \mathbf{x}^*\|$. Let $2r$ be the minimum over all such $\mathcal{S}$, which is bounded away from 0 as there are only finitely many such subsets. It follows that if $\|\mathbf{x} - \mathbf{x}^*\| \leq r$ then $\mathbf{x}^* \in \mathrm{co}(\mathcal{S}_k)$. □

Let r_0 denote the *critical radius* from Fact B.3 and K_0 the *critical iteration* so that $\|\mathbf{x}^* - \mathbf{x}_k\| \leq r_0$ is ensured for all $k \geq K_0$. The next proposition bounds the magnitude of K_0.

Proposition B.4 (Finite burn-in with linear rate)**.** *Denote by δ the pyramidal width of the polytope $\mathcal{X}$, as defined in Lacoste-Julien and Jaggi (2015). Then for all $k \geq K_0$ it holds $\mathbf{x}^* \in \mathrm{co}(\mathcal{S}_k)$ and for any algorithm that makes in each iteration at least as much progress as the Away-Step Frank-Wolfe Algorithm, we have the bound*

$$K_0 \leq \frac{8L}{\mu}\left(\frac{D}{\delta}\right)^2 \log\left(\frac{2(f(\mathbf{x}_0) - f(\mathbf{x}^*))}{\mu {r_0}^2}\right).$$

Proof. Since the algorithm makes at least as much progress as the Away-Step Frank-Wolfe algorithm, we can use the convergence rate of Lacoste-Julien and Jaggi (2015) to bound the primal gap at step k. Using the μ-strong convexity of f, we have that $f(\mathbf{x}_k) - f(\mathbf{x}^*) \geq \mu/2\|\mathbf{x}_k - \mathbf{x}^*\|$, allowing us to relate the primal gap to the distance to the optimum. □

To achieve local acceleration, we couple the AFW steps with a modification of the μAGD+ algorithm (Cohen et al., 2018) that we introduce here. Unlike its original version (Cohen et al., 2018), the version provided here (Lemma 3.2) allows coupling of the method with an arbitrary sequence of points from the feasible set, it supports inexact minimization oracles, and it supports changes in the convex set (which correspond to active sets from AFW) on which projections are performed. These modifications are crucial to being able to achieve local acceleration without any additional knowledge about the polytope or the position of the minimizer $\mathbf{x}^*$. Further, we are not aware of any other methods that allow changes to the feasible set as described here, and, thus, the result from Lemma 3.2 may be of independent interest.

Lemma 3.2. *(Convergence of the modified μAGD+) Let $f : \mathcal{X} \to \mathbb{R}$ be L-smooth and μ-strongly convex, and let $\mathcal{X}$ be a closed convex set. Let $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{u}\in\mathcal{X}} f(\mathbf{x})$, and let $\{\mathcal{C}_i\}_{i=0}^k$ be a sequence of convex subsets of $\mathcal{X}$ such that $\mathcal{C}_i \subseteq \mathcal{C}_{i-1}$ for all i and $\mathbf{x}^* \in \bigcap_{i=0}^k \mathcal{C}_i$. Let $\{\tilde{\mathbf{x}}_i\}_{i=0}^k$ be any (fixed) sequence of points from $\mathcal{X}$. Let $a_0 = 1$, $\frac{a_k}{A_k} = \theta$ for $k \geq 1$, where $A_k = \sum_{i=0}^k a_i$ and $\theta = \sqrt{\frac{\mu}{2L}}$. Let $\mathbf{y}_0 \in \mathcal{X}$, $\mathbf{x}_0 = \mathbf{w}_0$, and $\mathbf{z}_0 = L\mathbf{y}_0 - \nabla f(\mathbf{y}_0)$. For $k \geq 1$, define iterates $\mathbf{x}_k$ by:*

$$\begin{aligned} \mathbf{y}_k &= \frac{1}{1+\theta}\mathbf{x}_{k-1} + \frac{\theta}{1+\theta}\mathbf{w}_{k-1}, \\ \mathbf{z}_k &= \mathbf{z}_{k-1} - a_k\nabla f(\mathbf{y}_k) + \mu a_k \mathbf{y}_k, \\ \hat{\mathbf{x}}_k &= (1-\theta)\mathbf{x}_{k-1} + \theta\mathbf{w}_k, \\ \mathbf{x}_k &= \operatorname{argmin}\{f(\hat{\mathbf{x}}_k),\, f(\tilde{\mathbf{x}}_k)\} \end{aligned} \tag{3.2}$$

where, for all $k \geq 0$, $\mathbf{w}_k$ is defined as an ϵ_k^m-approximate solution of:

$$\min_{\mathbf{u}\in\mathcal{C}_k} \Big\{ -\langle \mathbf{z}_k, \mathbf{u}\rangle + \frac{\mu A_k + \mu_0}{2}\|\mathbf{u}\|^2 \Big\}, \tag{3.3}$$

with $\mu_0 \stackrel{\text{def}}{=} L - \mu$. Then, for all $k \geq 0$, $\mathbf{x}_k \in \mathcal{X}$ and:

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq (1-\theta)^k \frac{(L-\mu)\|\mathbf{x}^* - \mathbf{y}_0\|^2}{2} \\ + \frac{2\sum_{i=0}^{k-1}\epsilon_i^m + \epsilon_k^m}{A_k}. \end{aligned}$$

Proof. We first show by induction on k that $\mathbf{x}_k \in \mathcal{X}$. The claim is true initially, by the statement of the lemma. Assume that the claim is true for the iterates up to $k-1$. Then, $\hat{\mathbf{x}}_k$ must be from $\mathcal{X}$, as it is a convex combination of $\mathbf{x}_{k-1} \in \mathcal{X}$ (by the inductive hypothesis) and $\mathbf{w}_k \in \mathcal{C}_k \subseteq \mathcal{X}$. By assumption, $\tilde{\mathbf{x}}_k \in \mathcal{X}$, for all k. Thus, it must be $\mathbf{x}_k \in \mathcal{X}$.

The rest of the proof relies on showing that $A_kG_k \leq A_{k-1}G_{k-1} + \epsilon_k^m + \epsilon_{k-1}^m$ and on bounding A_0G_0, where G_k is an approximate duality gap defined as $G_k = U_k - L_k$. Here, the upper bound is defined as $U_k = f(\mathbf{x}_k)$, while the lower bound on $L_k \geq f(\mathbf{x}^*)$ can be defined as (see Appendix A):

$$L_k \stackrel{\text{def}}{=} \frac{\sum_{i=0}^k a_i f(\mathbf{y}_i) + \min_{\mathbf{u}\in\mathcal{C}_k} m_k(\mathbf{u}) - \frac{\mu_0}{2}\|\mathbf{x}^* - \mathbf{y}_0\|^2}{A_k},$$

where $\mu_0 = L - \mu$ and

$$m_k(\mathbf{u}) \stackrel{\text{def}}{=} \sum_{i=0}^k a_i \langle \nabla f(\mathbf{y}_i), \mathbf{u} - \mathbf{y}_i\rangle + \sum_{i=0}^k a_i \frac{\mu}{2}\|\mathbf{u} - \mathbf{y}_i\|^2 + \frac{\mu_0}{2}\|\mathbf{u} - \mathbf{y}_0\|^2.$$

It is not hard to verify that:

$$\operatorname*{argmin}_{\mathbf{u}\in\mathcal{C}_k} \Big\{ -\langle \mathbf{z}_k, \mathbf{u}\rangle + \frac{\mu A_k + \mu_0}{2}\|\mathbf{u}\|^2 \Big\} = \operatorname*{argmin}_{\mathbf{u}\in\mathcal{C}_k} m_k(\mathbf{u}), \quad \forall k.$$

Let us start by bounding A_0G_0. Recall that $a_0 = A_0 = 1$ and $\mathbf{x}_0 = \mathbf{w}_0$. By smoothness of f,

$$U_0 = f(\mathbf{x}_0) = f(\mathbf{w}_0) \leq f(\mathbf{y}_0) + \langle \nabla f(\mathbf{y}_0), \mathbf{w}_0 - \mathbf{y}_0\rangle + \frac{L}{2}\|\mathbf{w}_0 - \mathbf{y}_0\|^2. \tag{B.9}$$

On the other hand, as $\mu_0 = L - \mu$ and $\mathbf{w}_0$ is an ϵ_0^m-approximate minimizer of $\operatorname{argmin}_{\mathbf{u}\in\mathcal{C}} m_0(\mathbf{u}_0)$, we have:

$$\min_{\mathbf{u}\in\mathcal{C}_0} m_0(\mathbf{u}) \geq m_0(\mathbf{w}_0) - \epsilon_0^m = \langle \nabla f(\mathbf{y}_0), \mathbf{w}_0 - \mathbf{y}_0\rangle + \frac{L}{2}\|\mathbf{w}_0 - \mathbf{y}_0\|^2 - \epsilon_0^m. \tag{B.10}$$

Combining Eqs. (B.9) and (B.10) with the definition of L_k, we have that:

$$A_0 G_0 \leq \frac{\mu_0\|\mathbf{x}^* - \mathbf{y}_0\|^2}{2} + \epsilon_0^m = \frac{(L-\mu)\|\mathbf{x}^* - \mathbf{y}_0\|^2}{2} + \epsilon_0^m.$$

To complete the proof, it remains to show that $G_k \leq \frac{A_{k-1}}{A_k} G_{k-1} = (1-\theta)G_{k-1}$. Observe first, as $f(\mathbf{x}_k) \leq f(\hat{\mathbf{x}}_k)$, that we can bound the change in the upper bound as:

$$\begin{aligned} A_k U_k - A_{k-1}U_{k-1} &= A_k f(\mathbf{x}_k) - A_{k-1} f(\mathbf{x}_{k-1}) \\ &\leq a_k f(\mathbf{y}_k) + A_k(f(\hat{\mathbf{x}}_k) - f(\mathbf{y}_k)) + A_{k-1}(f(\mathbf{y}_k) - f(\mathbf{x}_{k-1})). \end{aligned}$$

Using smoothness and convexity of f, we further have:

$$A_k U_k - A_{k-1}U_{k-1} \leq a_k f(\mathbf{y}_k) + \langle \nabla f(\mathbf{y}_k), A_k\hat{\mathbf{x}}_k - A_{k-1}\mathbf{x}_{k-1} - a_k\mathbf{y}_k\rangle + \frac{A_k L}{2}\|\hat{\mathbf{x}}_k - \mathbf{y}_k\|^2. \tag{B.11}$$

By the definition of L_k, the change in the lower bound is:

$$A_k L_k - A_{k-1}L_{k-1} = a_k f(\mathbf{y}_k) + m_k(\mathbf{w}_k^*) - m_{k-1}(\mathbf{w}_{k-1}^*), \tag{B.12}$$

where $\mathbf{w}_k^* = \operatorname{argmin}_{\mathbf{u}\in\mathcal{C}_k} m_k(\mathbf{u})$.

To bound $m_k(\mathbf{w}_k^*) - m_{k-1}(\mathbf{w}_{k-1}^*)$, observe first that:

$$m_k(\mathbf{w}_k^*) - m_{k-1}(\mathbf{w}_{k-1}^*) \geq m_k(\mathbf{w}_k) - m_{k-1}(\mathbf{w}_{k-1}^*) - \epsilon_k^m. \tag{B.13}$$

as $\mathbf{w}_k \in \mathcal{C}_k$ is an ϵ_k^m-approximate minimizer of m_k. Further, observe that $m_k(\mathbf{u}) = m_{k-1}(\mathbf{u}) + a_k\langle \nabla f(\mathbf{y}_k), \mathbf{u} - \mathbf{y}_k\rangle + a_k\frac{\mu}{2}\|\mathbf{u} - \mathbf{y}_k\|^2$. Hence, we have:

$$\begin{aligned} &m_k(\mathbf{w}_k) - m_{k-1}(\mathbf{w}_{k-1}^*) \\ &\quad = a_k\langle \nabla f(\mathbf{y}_k), \mathbf{w}_k - \mathbf{y}_k\rangle + a_k\frac{\mu}{2}\|\mathbf{w}_k - \mathbf{y}_k\|^2 + m_{k-1}(\mathbf{w}_k) - m_{k-1}(\mathbf{w}_{k-1}^*). \end{aligned} \tag{B.14}$$

As $m_k(\mathbf{u})$ can be expressed as the sum of $\frac{\mu A_k + \mu_0}{2}\|\mathbf{u}\|^2$ and terms that are linear in $\mathbf{u}$, it is $(\mu_0 + \mu A_k)$-strongly convex. Observe that, as $\mathbf{w}_{k-1}^*$ minimizes m_{k-1} over $\mathcal{C}_{k-1}$ and $\mathbf{w}_k \in \mathcal{C}_k \subseteq \mathcal{C}_{k-1}$, by the first-order optimality condition, we have $\left\langle \nabla m_{k-1}(\mathbf{w}_{k-1}^*), \mathbf{w}_k - \mathbf{w}_{k-1}^*\right\rangle \geq 0$. Thus, it further follows that:

$$m_{k-1}(\mathbf{w}_k) \geq m_{k-1}(\mathbf{w}_{k-1}^*) + \frac{\mu_0 + \mu A_{k-1}}{2}\|\mathbf{w}_k - \mathbf{w}_{k-1}^*\|^2. \tag{B.15}$$

Next, observe that, as m_{k-1} is $(\mu_0 + \mu A_{k-1})$-strongly convex, $\mathbf{w}_{k-1}^*$ minimizes m_{k-1}, and $\mathbf{w}_{k-1}$ is an approximate minimizer, we have:

$$\frac{\mu_0 + \mu A_{k-1}}{2}\|\mathbf{w}_{k-1} - \mathbf{w}_{k-1}^*\|^2 \leq m_{k-1}(\mathbf{w}_{k-1}) - m_{k-1}(\mathbf{w}_{k-1}^*) \leq \epsilon_{k-1}^m. \tag{B.16}$$

Using Young's inequality ($(a+b)^2 \leq 2a^2 + 2b^2$ and so $a^2 \geq \frac{(a+b)^2}{2} - b^2$), we have, using Eq. (B.16), that:

$$\begin{aligned} \frac{\mu_0 + \mu A_{k-1}}{2}\|\mathbf{w}_k - \mathbf{w}_{k-1}^*\|^2 &\geq \frac{\mu_0 + \mu A_{k-1}}{4}\|\mathbf{w}_k - \mathbf{w}_{k-1}\|^2 - \frac{\mu_0 + \mu A_{k-1}}{2}\|\mathbf{w}_{k-1} - \mathbf{w}_{k-1}^*\|^2 \\ &\geq \frac{\mu_0 + \mu A_{k-1}}{4}\|\mathbf{w}_k - \mathbf{w}_{k-1}\|^2 - \epsilon_{k-1}^m. \end{aligned}$$

Combining the last inequality with Eqs. (B.13)–(B.15), we have:

$$\begin{aligned} m_k(\mathbf{w}_k^*) - m_{k-1}(\mathbf{w}_{k-1}^*) \geq& a_k\langle \nabla f(\mathbf{y}_k), \mathbf{w}_k - \mathbf{y}_k\rangle + a_k\frac{\mu}{2}\|\mathbf{w}_k - \mathbf{y}_k\|^2 \\ &+ \frac{\mu_0 + \mu A_{k-1}}{4}\|\mathbf{w}_k - \mathbf{w}_{k-1}\|^2 - \epsilon_{k-1}^m - \epsilon_k^m. \end{aligned}$$

Using that $\mu_0 \geq 0$, $\theta = \frac{a_k}{A_k}$, and applying Jensen's inequality to the last expression,

$$\begin{aligned} &m_k(\mathbf{w}_k^*) - m_{k-1}(\mathbf{w}_{k-1}^*) \\ &\qquad \geq a_k \langle \nabla f(\mathbf{y}_k), \mathbf{w}_k - \mathbf{y}_k \rangle + \frac{\mu A_k}{4} \|\mathbf{w}_k - (1-\theta)\mathbf{w}_{k-1} - \theta \mathbf{y}_k\|^2 - \epsilon_k^m - \epsilon_{k-1}^m. \end{aligned}$$

It is not hard to verify that $\hat{\mathbf{x}}_k - \mathbf{y}_k = \theta(\mathbf{w}_k - (1-\theta)\mathbf{w}_{k-1} - \theta \mathbf{y}_k)$. Hence, combining the last inequality with Eq. (B.12):

$$A_k L_k - A_{k-1} L_{k-1} \geq a_k f(\mathbf{y}_k) + a_k \langle \nabla f(\mathbf{y}_k), \mathbf{w}_k - \mathbf{y}_k \rangle + \frac{\mu A_k}{4\theta^2} \|\hat{\mathbf{x}}_k - \mathbf{y}_k\|^2 - \epsilon_k^m - \epsilon_{k-1}^m. \tag{B.17}$$

Finally, combining Eqs. (B.11) and (B.17), we have:

$$\begin{aligned} A_k G_k - A_{k-1} G_{k-1} \leq &\langle \nabla f(\mathbf{y}_k), A_k \hat{\mathbf{x}}_k - A_{k-1}\mathbf{x}_{k-1} - a_k \mathbf{w}_k \rangle + \frac{A_k}{2}\Big(L - \frac{\mu}{2\theta^2}\Big)\|\hat{\mathbf{x}}_k - \mathbf{y}_k\|^2 \\ &+ \epsilon_k^m + \epsilon_{k-1}^m \\ \leq &\epsilon_k^m + \epsilon_{k-1}^m, \end{aligned}$$

as $\hat{\mathbf{x}}_k = \frac{A_{k-1}}{A_k}\mathbf{x}_{k-1} + \frac{a_k}{A_k}\mathbf{w}_k$ and $\theta = \sqrt{\frac{\mu}{2L}}$, completing the proof. □

A simple corollary of Lemma 3.2 that will be useful for our analysis is as follows. It shows that if the algorithm from Lemma 3.2 is not restarted too often, we do not lose more than a constant factor (two) in the final bound on the iteration count.

Corollary B.5. *Define a restart of the method from Lemma 3.2 as setting $a_k = A_k = 1$, $\mathbf{y}_k = \mathbf{x}_{k-1}$, $\mathbf{w}_k = \mathbf{y}_k$, and $\mathbf{z}_k = L\mathbf{y}_k - \nabla f(\mathbf{y}_k)$. Let $\epsilon_i^m = \frac{a_i}{2}\bar{\epsilon}$, for some $\epsilon^m \geq 0$. If the method is restarted no more frequently than every $\frac{2}{\theta}\log(1/(2\theta^2) - 1)$ iterations, where $\theta = \sqrt{\mu/(2L)}$, then:*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{L-\mu}{\mu}\big(1-\theta\big)^{k/2}(f(\mathbf{x}_0) - f(\mathbf{x}^*)) + 2\bar{\epsilon}.$$

Proof. Denote $H = \frac{2}{\theta}\log(1/(2\theta^2) - 1)$. Let the iterations at which the restarts happen be denoted as $k_0 = 0$, k_1, $k_2, ...$, and note that, by assumption, $k_i \geq k_{i-1} + H$, for all $i \geq 1$. Assume w.l.o.g. that each k_i is even. We first claim that we have the following contraction between the successive restarts:

$$f(\mathbf{x}_{k_i}) - f(\mathbf{x}^*) \leq (1-\theta)^{(k_i - k_{i-1})/2}(f(\mathbf{x}_{k_{i-1}}) - f(\mathbf{x}^*)) + \bar{\epsilon}. \tag{B.18}$$

To prove the claim, observe first using $k_i - k_{i-1} \geq H$ that:

$$\frac{L-\mu}{\mu}(1-\theta)^{k_i - k_{i-1}} \leq \Big(\frac{1}{2\theta^2} - 1\Big)(1-\theta)^{\frac{1}{\theta}\log(\frac{1}{2\theta^2}-1)}(1-\theta)^{\frac{k_i - k_{i-1}}{2}} \leq (1-\theta)^{\frac{k_i - k_{i-1}}{2}}. \tag{B.19}$$

Applying Lemma 3.2 with $\mathbf{x}_{k_{i-1}}$ as an initial point and using strong convexity of f (which implies $f(\mathbf{x}_{k_{i-1}}) - f(\mathbf{x}^*) \geq \frac{\mu}{2}\|\mathbf{x}_{k_{i-1}} - \mathbf{x}^*\|^2$), we have:

$$f(\mathbf{x}_{k_i}) - f(\mathbf{x}^*) \leq \frac{L-\mu}{\mu}(1-\theta)^{k_i - k_{i-1}}(f(\mathbf{x}_{k_{i-1}}) - f(\mathbf{x}^*)) + \bar{\epsilon}.$$

Thus, combining the last inequality with (B.19), inequality (B.18) follows.

Applying Eq. (B.18) recursively and using that $k_i - k_{i-1} \geq H$, we further have:

$$\begin{aligned} f(\mathbf{x}_{k_i}) - f(\mathbf{x}^*) &\leq (1-\theta)^{k_i/2}(f(\mathbf{x}_0) - f(\mathbf{x}^*)) + \bar{\epsilon}\sum_{j=0}^{i}(1-\theta)^{jH/2} \\ &\leq (1-\theta)^{k_i/2}(f(\mathbf{x}_0) - f(\mathbf{x}^*)) + 2\theta^2\bar{\epsilon}. \end{aligned} \tag{B.20}$$

To complete the proof, fix an iteration k and let k_i be the last iteration up to k in which a restart happened. Applying Lemma 3.2 with k_i as the initial point, we get:

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}^*) &\leq \frac{L-\mu}{\mu}\big(1-\theta\big)^{k-k_i}(f(\mathbf{x}_{k_i}) - f(\mathbf{x}^*)) + \bar{\epsilon} \\ &\leq \frac{L-\mu}{\mu}\big(1-\theta\big)^{k/2}(f(\mathbf{x}_0) - f(\mathbf{x}^*)) + (1+2\theta^2)\bar{\epsilon}. \end{aligned}$$

It remains to note that $\theta^2 = \mu/(2L) \leq 1/2$. □

To obtain locally accelerated convergence, we show that from some iteration onwards, we can apply the accelerated method from Lemma 3.2 with $\mathcal{C}_k$ being the convex hull of the vertices from the active set and the sequence $\tilde{\mathbf{x}}_k$ being the sequence of the AFW steps. The pseudocode for the LaCG-AFW algorithm is provided in Algorithm 1 (Algorithm 4 in the appendix). For completeness, pseudocode for one iteration of the accelerated method (ACC), which is based on Eq. (3.2) is provided in Algorithm 5.

Algorithm 4 Locally Accelerated Conditional Gradients with Away-Step Frank-Wolfe (LaCG-AFW)

Let $\mathbf{x}_0 \in \mathcal{X}$ be an arbitrary point, $\mathcal{S}_0^{\text{AFW}} = \{\mathbf{x}_0\}$, $\boldsymbol{\lambda}_0^{\text{AFW}} = [1]$
Let $\mathbf{y}_0 = \hat{\mathbf{x}}_0 = \mathbf{w}_0 = \mathbf{x}_0$, $\mathbf{z}_0 = -\nabla f(\mathbf{y}_0) + L\mathbf{y}_0$, $\mathcal{C}_1 = \text{co}(\mathcal{S}_0^{\text{AFW}})$, $a_0 = A_0 = 1$, $\theta = \sqrt{\frac{\mu}{2L}}$, $\mu_0 = L - \mu$
$H = \frac{2}{\theta}\log(1/(2\theta^2) - 1)$ ▷ Minimum restart period
$r_f = \text{false}$, $r_c = 0$ ▷ Restart flag and restart counter initialization
for $k = 1$ to K **do**
  $\mathbf{x}_k^{\text{AFW}}, \mathcal{S}_k^{\text{AFW}}, \boldsymbol{\lambda}_k^{\text{AFW}} = \text{AFW}(\mathbf{x}_{k-1}^{\text{AFW}}, \mathcal{S}_{k-1}^{\text{AFW}}, \boldsymbol{\lambda}_{k-1}^{\text{AFW}})$ ▷ Independent AFW update
  $A_k = A_{k-1}/(1-\theta)$, $a_k = \theta A_k$
  if r_f and $r_c \geq H$ **then** ▷ Restart criterion is met
    $\mathbf{y}_k = \text{argmin}\{f(\mathbf{x}_k^{\text{AFW}}), f(\hat{\mathbf{x}}_k)\}$
    $\mathcal{C}_{k+1} = \text{co}(\mathcal{S}_k^{\text{AFW}})$ ▷ Updating feasible set for the accelerated sequence
    $a_k = A_k = 1$, $\mathbf{z}_k = -\nabla f(\mathbf{y}_k) + L\mathbf{y}_k$ ▷ Restarting accelerated sequence
    $\hat{\mathbf{x}}_k = \mathbf{w}_k = \text{argmin}_{\mathbf{u}\in\mathcal{C}_{k+1}}\{-\langle \mathbf{z}_k, \mathbf{u}\rangle + \frac{L}{2}\|\mathbf{u}\|^2\}$
    $r_c = 0$, $r_f = \text{false}$ ▷ Resetting the restart indicators
  else
    if $\mathcal{S}_k^{\text{AFW}} \setminus \mathcal{S}_{k-1}^{\text{AFW}} \neq \emptyset$ **then** ▷ If a vertex was added to the active set
      $r_f = \text{true}$ ▷ Raise restart flag
    if $r_f = \text{true}$ **then**
      $\mathcal{C}_k = \mathcal{C}_{k-1}$ ▷ Freeze the feasible set
      $\hat{\mathbf{x}}_k, \mathbf{z}_k, \mathbf{w}_k = \text{ACC}(\hat{\mathbf{x}}_{k-1}, \mathbf{z}_{k-1}, \mathbf{w}_{k-1}, a_k, A_k, \mathcal{C}_k)$ ▷ Run AGD+ uncoupled from CG.
    else
      $\mathcal{C}_k = \text{co}(\mathcal{S}_k^{\text{AFW}})$ ▷ Update the feasible set
      $\hat{\mathbf{x}}_k, \mathbf{z}_k, \mathbf{w}_k = \text{ACC}(\mathbf{x}_{k-1}, \mathbf{z}_{k-1}, \mathbf{w}_{k-1}, a_k, A_k, \mathcal{C}_k)$ ▷ Run AGD+ coupled to CG.
  $\mathbf{x}_k = \text{argmin}\{f(\mathbf{x}_k^{\text{AFW}}), f(\hat{\mathbf{x}}_k), f(\mathbf{x}_{k-1})\}$ ▷ Choose the better step + monotonicity
  $r_c = r_c + 1$ ▷ Increment the restart counter

Algorithm 5 Accelerated Step $\text{ACC}(\mathbf{x}_{k-1}, \mathbf{z}_{k-1}, \mathbf{w}_{k-1}, \mu, \mu_0, a_k, A_k, \mathcal{C}_k)$

$\theta = a_k/A_k$
$\mathbf{y}_k = \frac{1}{1+\theta}\mathbf{x}_{k-1} + \frac{\theta}{1+\theta}\mathbf{w}_{k-1}$
$\mathbf{z}_k = \mathbf{z}_{k-1} - a_k\nabla f(\mathbf{y}_k) + \mu a_k \mathbf{y}_k$, $\mathbf{w}_k = \text{argmin}_{\mathbf{u}\in\mathcal{C}_k}\{-\langle \mathbf{z}_k, \mathbf{u}\rangle + \frac{\mu A_k + \mu_0}{2}\|\mathbf{u}\|^2\}$
$\hat{\mathbf{x}}_k = (1-\theta)\mathbf{x}_{k-1} + \theta\mathbf{w}_k$
return $\hat{\mathbf{x}}_k, \mathbf{z}_k, \mathbf{w}_k$

Main Theorem 3.3. (Convergence analysis of *Locally Accelerated Conditional Gradients*) Let $\mathbf{x}_k$ be the solution

output by Algorithm 1 and r_0 be the critical radius (see Fact B.3 in Appendix B.2). If:

$$k \geq \min \left\{ \frac{8L}{\mu} \left(\frac{D}{\delta}\right)^2 \log \left(\frac{f(\mathbf{x}_0) - f(\mathbf{x}^*)}{\epsilon}\right), \right.$$
$$\left. K_0 + H + 2\sqrt{\frac{2L}{\mu}} \log \left(\frac{(L-\mu){r_0}^2}{2\epsilon}\right) \right\},$$

where $H = 2\sqrt{2L/\mu}\log(L/\mu - 1)$ and $K_0 = \frac{8L}{\mu}\left(\frac{D}{\delta}\right)^2 \log\left(\frac{2(f(\mathbf{x}_0)-f(\mathbf{x}^*))}{\mu {r_0}^2}\right)$, then:

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon.$$

Proof. The statement of the theorem is a direct consequence of the following observations about Algorithm 1 (Algorithm 4 in the appendix). First, observe that the AFW algorithm is run independently of the accelerated sequence, and, in particular, the accelerated sequence has no effect on the AFW-sequence whatsoever. Further, in any iteration, the set $\mathcal{C}_k$ that we project onto is the convex hull of some active set $\mathcal{S}_i^{\text{AFW}} \subseteq \mathcal{X}$ for some $0 \leq i \leq k-1$ implying $\hat{\mathbf{x}}_k \in \mathcal{X}$ – each $\hat{\mathbf{x}}_k$ is hence feasible.

Now, as in any iteration k the solution outputted by the algorithm is $\mathbf{x}_k = \operatorname{argmin}\{f(\mathbf{x}_k^{\text{AFW}}), f(\hat{\mathbf{x}}_k)\}$, the algorithm never makes less progress than AFW. This immediately implies (by a standard AFW guarantee; see Lacoste-Julien and Jaggi (2015) and Proposition B.4) that for $k \geq \frac{8L}{\mu}\left(\frac{D}{\delta}\right)^2 \log\left(\frac{f(\mathbf{x}_0)-f(\mathbf{x}^*)}{\epsilon}\right)$, it must be that $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \epsilon$, which establishes the unaccelerated part of the minimum in the asserted rate.

Further, there exists an iteration $K \leq K_0$ such that for all $k \geq K$ it holds $\mathbf{x}^* \in \text{co}(\mathcal{S}_k^{\text{AFW}})$ (see Proposition B.4). Let K be the first such iteration. Then, the AFW algorithm must have added a vertex in iteration K as otherwise $\mathbf{x}^* \in \text{co}(\mathcal{S}_{k-1}^{\text{AFW}})$, contradicting the minimality of K. Due to the restarting criterion from Algorithm 1, a restart must happen by iteration $K_0 + H$. Thus, for $k \geq K_0 + H$, it must be $\mathbf{x}^* \in \mathcal{C}_k$.

Further, the restarting criterion implies that we perform at least $H = \frac{2}{\theta}\log(1/(2\theta^2) - 1)$ iterations between successive restarts of the accelerated sequence $\{\hat{\mathbf{x}}_k\}$ and, unless a restart happens, we also have that $\mathcal{C}_k \subseteq \mathcal{C}_{k-1}$. Thus, starting from iteration $K_0 + H$, Lemma 3.2 and Corollary B.5 apply and $\{\mathbf{x}_k\}$ converges to to $\mathbf{x}^*$ at an accelerated rate. The remaining $2\sqrt{\frac{L}{\mu}}\log\left(\frac{(L-\mu)r_0^2}{2\epsilon}\right)$ part of the minimum in the asserted rate follows now by Corollary B.5. □

Remark B.6 (Inexact projection oracles.)**.** For simplicity, we stated Theorem 3.3 assuming exact minimization oracle ($\epsilon_i^m = 0$ in Lemma 3.2). Clearly, it suffices to have $\epsilon_i^m = \frac{a_i}{8}\epsilon$ and invoke Theorem 3.3 for target accuracy $\epsilon/2$.

Remark B.7 (Further improvements to the practical performance.)**.** If in any iteration the Wolfe gap of the accelerated sequence on $\mathcal{C}_k$, $\max_{\mathbf{u}\in\mathcal{C}_k}\langle \nabla f(\mathbf{x}_k), \mathbf{x}_k - \mathbf{u}\rangle$, is smaller than the target accuracy of the projection subproblem (order-$\frac{\epsilon}{\sqrt{\mu L}}$), then f cannot be reduced by more than order-$\frac{\epsilon}{\sqrt{\mu L}}$ on $\mathcal{C}_k$, and one can safely perform an early restart without affecting the theoretical convergence guarantee.

Remark B.8 (Running Algorithm 1 when $\mathbf{x}^* \in \text{int}(\mathcal{X})$)**.** Usually we do not know ahead of time whether $\mathbf{x}^* \in \text{int}(\mathcal{X})$ or whether $\mathbf{x}^*$ is in the relative interior of a face of $\mathcal{X}$. However, we can simply run Algorithm 1 agnostically, as in the case where $\mathbf{x}^* \in \text{int}(\mathcal{X})$ we still exhibit local acceleration with an argumentation and convergence analysis analogous to the one in Section 3.1. In particular, the assumptions of Section 3.2 are only needed to establish a bound for the estimation in Proposition B.4.

Remark B.9 (Variant relying exclusively on a linear optimization oracle)**.** Similar as in the Conditional Gradient Sliding (CGS) algorithm (Lan and Zhou, 2016) we can also solve the projection problems using (variants of) CG. The resulting algorithm is then fully projection-free similar to CGS. In fact, a variant of CGS is recovered if we ignore the AFW steps and only run the accelerated sequence with such projections realized by CG.

C Computational Results

We provide a detailed comparison of the performance of different LaCG variants relative to other state of the art algorithms, comparing the primal gap and dual gap evolution both in terms of the iteration count and in terms of

wall-clock time. For the example over the Birkhoff polytope, the MIPLIB instance and the probability simplex we use the stepsize rule $\gamma_t = \min\left\{\frac{\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{v}_t\rangle}{L\|\mathbf{x}_t - \mathbf{v}_t\|^2}, \gamma_{\max}\right\}$ for all the algorithms that do not have a fixed step size rule, where $\gamma_{\max}$ is the maximum step size that can be taken without leaving the polytope. In the video colocalization and the traffic network example we use exact linesearch for all the algorithms whenever possible (as was done in the video colocalization case in Lacoste-Julien and Jaggi (2015) and Joulin et al. (2014)). Regarding the linear optimization oracle for the MIPLIB and video-colocalization instance, we used Gurobi to solve the MIP problem with linear cost function and the shortest path problem over the DAG respectively.

Note that when the algorithm starts running, the CG variant will add a vertex to its active set in the first iteration, at which point the set $\mathcal{C}_k$ will be frozen and will contain only two vertices until the next restart happens. For this reason, the convergence in the first H iterations is driven by the CG steps, and due to the low overhead of computing the accelerated steps over $\mathcal{C}_k$ with $|\mathcal{C}_k| = 2$, the wall-clock performance in the first H iterations will be approximately equal to that of the CG variant running by itself.

C.1 Computational Enhancements to Algorithm 1

In order to speed up the convergence of the algorithm when the burn-in phase has not been completed we can substitute Line 23 in Algorithm 1 (or Line 23 in Algorithm 4) with the following:

Algorithm 6 LaCG Enhancement

1: **if** $f(\mathbf{x}_{k-1}) \leq f(\mathbf{x}_k^{\text{AFW}})$ and $f(\mathbf{x}_{k-1}) \leq f(\hat{\mathbf{x}}_k)$ **then**
2: $\quad \mathbf{x}_k = \mathbf{x}_{k-1}$
3: $\quad$ **if** $f(\mathbf{x}_k) \leq f(\mathbf{x}_k^{\text{AFW}})$ and $\mathcal{C}_{k+1} \subseteq \mathcal{S}_{k+1}^{\text{AFW}}$ **then**
4: $\quad\quad \mathcal{S}_{k+1}^{\text{AFW}} = \text{vert}\,(\mathcal{C}_{k+1})$
5: $\quad\quad \mathbf{x}_k^{\text{AFW}} = \mathbf{x}_k$
6: **else**
7: $\quad \mathbf{x}_k = \text{argmin}\{f(\mathbf{x}_k^{\text{AFW}}),\, f(\hat{\mathbf{x}}_k)\}$

Note that although this means that the AFW sequence is not independent of the accelerated steps anymore, this does not affect the theoretical guarantees shown in Theorem 5. The previous operation leads to greater progress during the burn-in phase, as after a restart the accelerated sequence active set $\mathcal{C}_{k+1}$ is usually frozen, and the accelerated steps tend to converge to the function minimizer over $\mathcal{C}_{k+1}$ with relative ease, progressing very quickly before stagnating (as little further progress can be made over $\mathcal{C}_{k+1}$). If this happens and $f(\hat{\mathbf{x}}_k) \leq f(\mathbf{x}_k^{\text{AFW}})$ along with $\mathcal{C}_{k+1} \subseteq \text{co}(\mathcal{S}_{k+1}^{\text{AFW}})$ (this is equivalent to the AFW steps not having dropped a vertex contained in $\mathcal{C}_{k+1}$) the AFW steps can pick up progress from where the accelerated sequence is located.

C.2 Solving Problem (3.3)

At each iteration, LaCG has to solve the following subproblem to accuracy $\frac{\epsilon a_i}{8}$:

$$\mathbf{w}_k = \underset{u \in \mathcal{C}_k}{\text{argmin}} \left\{ - \langle \mathbf{z}_k, \mathbf{u}\rangle + \frac{\mu A_k + \mu_0}{2} \|\mathbf{u}\|^2 \right\},$$

where $\mathbf{z}_k \in \mathbb{R}^n$, μ, μ_0, and A_k are given. This problem over the convex hull of $\mathcal{C}_k$ can transformed to one over the probability simplex by noting that $\mathbf{u} = \mathcal{V}_k \boldsymbol{\lambda}$, where $\mathcal{V}_k$ is the matrix that contains the elements in $\mathcal{C}_k$ as column vectors and $\boldsymbol{\lambda} \in \Delta_{|\mathcal{C}_k|}$, where $|\mathcal{C}_k|$ is the cardinality of the set $\mathcal{C}_k$. Rewritting the previous subproblem leads to:

$$\boldsymbol{\lambda}_k = \underset{\boldsymbol{\lambda} \in \Delta_{|\mathcal{C}_k|}}{\text{argmin}} \left\{ - \langle \mathcal{V}_k \mathbf{z}_k, \boldsymbol{\lambda}\rangle + \frac{\mu A_k + \mu_0}{2} \boldsymbol{\lambda}^T \mathcal{V}_k^T \mathcal{V}_k \boldsymbol{\lambda} \right\}, \tag{C.1}$$

where now the solution to our problem is provided by $\mathbf{w}_k = \mathcal{V}_k \boldsymbol{\lambda}_k$. We begin by noting that whenever the AFW step adds a vertex to $\mathcal{S}_k$, the set $\mathcal{C}_k$ is frozen and remains fixed until the next restart happens. This effectively means that the term $\mathcal{V}_k^T \mathcal{V}_k$ in Equation C.1 remains fixed and only the term $\mathcal{V}_k \mathbf{z}_k$ changes over iterations until the next restart happens. If, on the other hand, the AFW step removes a vertex from $\mathcal{S}_k$ and the set $\mathcal{C}_k$ is not

frozen, both $\mathcal{V}_k^T\mathcal{V}_k$ and $\mathcal{V}_k\mathbf{z}_k$ have to be updated at that iteration. In our experiments, the AFW step usually adds a vertex to $\mathcal{S}_k$ immediately after a restart has happened and so the set $\mathcal{C}_k$ remains frozen for most of the iterations, and we only need to update $\mathcal{V}_k\mathbf{z}_k$ at each iteration, which has a complexity of $\mathcal{O}(|\mathcal{C}_k|n)$

At each iteration, we solve Problem C.1 using Nesterov accelerated gradient descent (Nesterov, 2018) for smooth convex or strongly convex functions. In order to do so, we calculate the largest and smallest eigenvalue of the matrix $\mathcal{V}_k^T\mathcal{V}_k$ when these are updated, and this is done with scipy's ARPACK package, which for symmetric matrices uses a variant of the Lanczos method. As an initial point for Nesterov accelerated gradient descent, we use the solution to the problem from the previous iteration, i.e., $\boldsymbol{\lambda}_{k-1}$, if $\mathcal{C}_k = \mathcal{C}_{k-1}$, which allows the algorithm to find a suitable solution after only a few iterations.

In order to project onto the simplex, we use the $\mathcal{O}(n \log n)$ algorithm described in Duchi et al. (2008, Algorithm 1). An alternative would be to use a negative entropy regularizer in the implementation of Nesterov method, for which Bregman projection steps can be computed in closed form. However, we found in our experiments that Euclidean projections using Duchi et al. (2008, Algorithm 1). were faster, and thus opted for using them despite their slightly worse theoretical guarantee.

C.3 LaCG over the Probability Simplex

The probability simplex is a simple polytope for which efficient projection operators exist, with complexity $\mathcal{O}(n \log n)$. Due to the existence of these operators and the $\mathcal{O}(\sqrt{L/\mu} \log 1/\epsilon)$ global convergence guarantee of accelerated projected methods, CG methods are seldom used over this feasible region (despite the $\mathcal{O}(n)$ complexity for the linear optimization oracle over the simplex).

Despite being a toy-example, the structure of this polytope lends itself well to several computational simplifications. As we mentioned earlier, there is no need to maintain an active set in this case, as a single pass over the current iterate (which has complexity $\mathcal{O}(n)$) allows us to recover the active set by retrieving the non-zero components of the current point, whose corresponding standard orths correspond to the elements in the active set. This means that the away step oracle simply returns the largest gradient component over the non-zero coordinates of the current iterate.

Furthermore, if we consider the subproblem that needs to be solved at each iteration of the LaCG algorithm, shown in Equation (C.1), and we note that $\mathcal{V}_k^T\mathcal{V}_k = I$, we see that the subproblem can be rephrased as follows:

$$\begin{aligned}\boldsymbol{\lambda}_k &= \operatorname*{argmin}_{\boldsymbol{\lambda}\in\Delta_{|\mathcal{C}_k|}} \left\{ -\langle \mathcal{V}_k\mathbf{z}_k, \boldsymbol{\lambda}\rangle + \frac{\mu A_k + \mu_0}{2}\boldsymbol{\lambda}^T\boldsymbol{\lambda} \right\} \\ &= \operatorname*{argmin}_{\boldsymbol{\lambda}\in\Delta_{|\mathcal{C}_k|}} \left\| \boldsymbol{\lambda} - \frac{\mathcal{V}_k\mathbf{z}_k}{\mu A_k + \mu_0} \right\|_2^2,\end{aligned}$$

where the term $\mathcal{V}_k\mathbf{z}_k$ can be efficiently computed in $\mathcal{O}(n)$ time, and a single call to the simplex projection over $\Delta_{|\mathcal{C}_k|}$ will return the solution to the subproblem in barycentric coordinates.

C.3.1 Going from the ℓ_1-ball to the probability simplex

Lasso regression is a problem of interest in the benchmarking of many first-order methods, where the goal is to solve a quadratic problem with $f(\mathbf{x}) = \|A\mathbf{x} - \mathbf{b}\|_2^2 = \frac{\mathbf{x}^TQ\mathbf{x}}{2} + \mathbf{c}^T\mathbf{x}$ over a scaled ℓ_1^n-ball in $\mathbb{R}^n$, where Q is positive semidefinite. It would be advantageous if all the computational simplifications that applied to the simplex could be extended to the ℓ_1-ball. In fact, there is a simple change of variables that allows us to write a problem over the simplex that is equivalent to that of the lasso. Consider $\mathbf{z} \in \Delta_{2n}$, and define $x_i = z_i - z_{n+i}$ for all $i \in [\![1, n]\!]$, then:

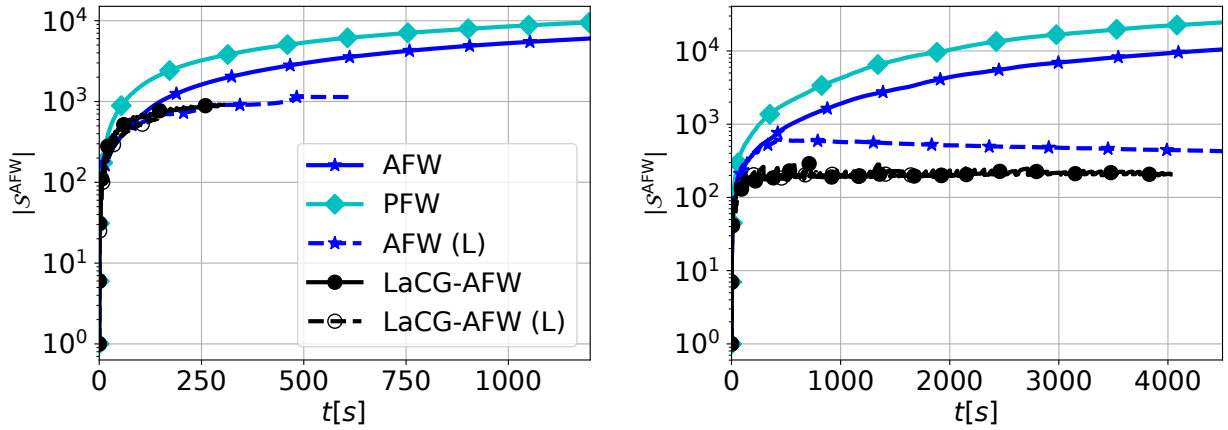
$$\min_{\mathbf{x}\in\ell_1^n} \frac{\mathbf{x}^TQ\mathbf{x}}{2} + \mathbf{c}^T\mathbf{x} = \min_{\mathbf{z}\in\Delta_{2n}} \mathbf{z}^T \left[\begin{array}{c|c} Q & -Q \\ \hline -Q & Q \end{array}\right] \mathbf{z} + \left[\begin{array}{c} c \\ \hline -c \end{array}\right]^T \mathbf{z} \tag{C.2}$$

In order for the LaCG algorithm to be useful, we need the objective function to be positive definite, in order to achieve acceleration. But the equivalent problem over the simplex is only convex, as the determinant of the block-matrix on the right-hand side of Equation (C.2) is equal to zero. Therefore, the LaCG algorithm has to be

applied directly to the problem over the ℓ_1-ball, shown on the left-hand side. This means that there is no way of rewriting the lasso problem as a strongly convex problem over the probability simplex.

Note that the DICG algorithm in Garber and Hazan (2016) is applicable to 0/1 polytopes (such as the probability simplex) with L-smooth and μ-strongly convex objective functions. If we perform this change of variables to go from a lasso regression problem over the ℓ_1-ball to one over the probability simplex, the resulting objective function over the simplex will no longer be strongly convex, and therefore the theoretical guarantees in Garber and Hazan (2016) no longer hold in this case. However, the DICG algorithm for general polytopes (Bashiri and Zhang, 2017) can be applied to the lasso regression problem, and its theoretical guarantees hold true.

C.4 On the Evolution of $\mathcal{C}$ and $\mathcal{S}$



(a) Cardinality of the AFW active set for the Birkhoff example ($n = 1600$).

(b) Cardinality of the AFW active set for the MIPLIB example ($n = 504$).

Figure 5: Comparison of the evolution of the cardinality of the active set $\mathcal{S}_{k+1}^{\text{AFW}}$ for two of the examples.

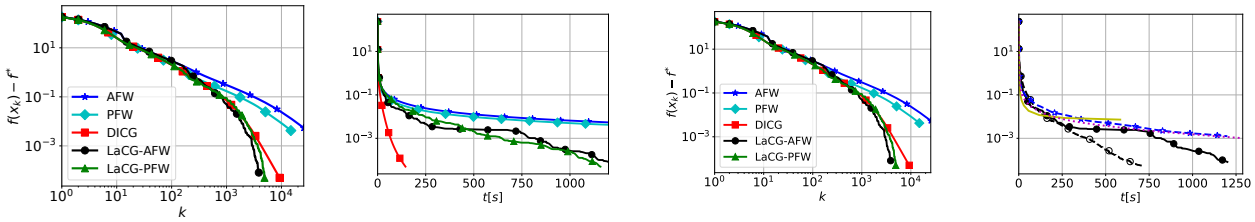
The LaCG algorithm benefits from having a sparse active set. This is due to the fact that finding the away vertex in the AFW algorithm has complexity $\mathcal{O}(|\mathcal{S}_{k+1}|n)$ and Problem (C.1) will in general be easier to solve the smaller the cardinality of the set $\mathcal{C}_{k+1}$. Because of this, it is often useful to cull the active sets $\text{vert}(\mathcal{C}_{k+1})$ and $\mathcal{S}_{k+1}$ to promote sparsity. This can be done in conjunction with the operations in Algorithm 6. More specifically, before Line 4, we can discard the stale vertices in the convex decomposition of $\hat{\mathbf{x}}_k$ over the set $\mathcal{C}_{k+1}$, that is, we can eliminate the vertices in $\mathcal{C}_{k+1}$ that have a zero coefficient when $\hat{\mathbf{x}}_k$ is expressed as the convex combination of the elements in $\mathcal{C}_{k+1}$ (also referred to as barycentric coordinates). This can be easily done as the accelerated sequence maintains a decomposition of the current point in the barycentric coordinates of $\mathcal{C}_{k+1}$ in order to solve the projection subproblem. This culling can be performed regardless of which CG variant is used in the LaCG algorithm.

The evolution of the cardinality of the active set $\mathcal{S}^{\text{AFW}}$ in terms of time is shown in Fig. 5(a)-5(b) for two of the examples in the computational section. As can be seen in Fig. 5, this culling of the active set is effective in keeping in check the tendency of the AFW steps in the LaCG algorithm to add more vertices.

C.5 Additional Experiments

We also consider the video co-localization problem, which can be shown to be equivalent to minimizing a quadratic objective function over a flow polytope (Joulin et al., 2014). In this problem, the linear optimization oracle corresponds to finding a shortest path in a directed acyclic graph (DAG), for which there are algorithms that solve this problem in running time $\mathcal{O}(E + V)$, where E and V are the number of edges and vertices of the graph, respectively. We solve this problem over a directed acyclic graph with 3180 edges and 227 nodes that mimics the structure shown in shown in Joulin et al. (2014), i.e., it has a source node connected to a layer of 15 nodes, and each layer is fully connected with directed edges to the next layer, to make a total of 15 layers (of 15 nodes per layer). The last layer of nodes is connected to a node that acts as a sink. The quadratic was generated

in the same way as in the Birkhoff polytope example, i.e. $f(\mathbf{x}) = \mathbf{x}^T \frac{M^T M + I}{2} \mathbf{x}$, with M having 1% non-zero entries drawn from a standard Gaussian distribution. The matrix $M^T M$ has 27% non-zero entries. The condition number in this instance is $L/\mu = 140$.



(a) Iteration (b) Wall-clock time (c) Iteration (d) Wall-clock time

Figure 6: Video co-localization: Algorithm comparison in terms of (a),(c) iteration count and (b),(d) wall-clock time.

In this example, DiCG is much more efficient than any of the active set based methods. Even though DiCG exhibits a slower convergence rate than LaCG variants, it greatly benefits from not maintaining an active set, which makes its iterations much more efficient. However, as discussed before, DiCG is not broadly applicable. Moreover, even in cases where it is applicable, DiCG can still be slower than LaCG variants once the linear optimization oracle is not very fast or the active sets do not get too large. It is an interesting open question whether local acceleration can be achieved with decomposition invariant methods.