Restarting Frank-Wolfe

Thomas Kerdreux[1,4], Alexandre d'Aspremont[2,4], and Sebastian Pokutta[3]

[1]INRIA, Paris
[2]CNRS, UMR 8548
[3]Industrial and Systems Engineering, Georgia Institute of Technology, USA.
[4]D.I., École Normale Supérieure, Paris, France.

October 8, 2018

Abstract

Conditional Gradients (aka Frank-Wolfe algorithms) form a classical set of methods for constrained smooth convex minimization due to their simplicity, the absence of projection step, and competitive numerical performance. While the vanilla Frank-Wolfe algorithm only ensures a worst-case rate of $O(1/\epsilon)$, various recent results have shown that for strongly convex functions, the method can be slightly modified to achieve linear convergence. However, this still leaves a huge gap between sublinear $O(1/\epsilon)$ convergence and linear $O(\log 1/\epsilon)$ convergence to reach an ϵ-approximate solution. Here, we present a new variant of Conditional Gradients, that can dynamically adapt to the function's geometric properties using restarts and thus smoothly interpolates between the sublinear and linear regimes.

1 Introduction

We consider smooth constraint convex minimization, solving problems of the form

$$\min_{x \in \mathcal{C}} f(x),$$

where f is a smooth convex function and $\mathcal{C}$ is a polytope. As soon as the geometry of $\mathcal{C}$ is reasonably complicated, so that projections onto the set are computationally expensive, projection-free first-order methods such as Conditional Gradients [Levitin and Polyak, 1966] (also known as Frank-Wolfe methods [Frank and Wolfe, 1956]) become an efficient alternative as they only require first-order access to the function under consideration as well as access to an efficient linear optimization oracle for the feasible region $\mathcal{C} \subseteq \mathbb{R}^n$ which, given a linear objective $c \in \mathbb{R}^n$, outputs $\arg\min_{x \in \mathcal{C}} c^T x$.

In order to reach an ϵ-approximate solution $\hat{x}$, so that $f(\hat{x}) - f(x^*) < \epsilon$, where x^* is an optimal solution, the standard Frank-Wolfe algorithm requires a number of iterations of order $O(1/\epsilon)$, that cannot be improved upon in general. A series of recent works (see e.g., [Garber and Hazan, 2013, Lacoste-Julien and Jaggi, 2015]; see also [Lan and Zhou, 2014] for conditional gradient sliding) showed that when f is strongly convex the convergence rate of the standard case can be improved to $O(\log 1/\epsilon)$ and various extensions further improved upon these results for special cases (see e.g., [Lacoste-Julien et al., 2013, Freund and Grigas, 2016, Garber and Meshi, 2016, Braun et al., 2017, Lan et al., 2017, Bashiri and Zhang, 2017, Garber et al., 2018, Kerdreux et al., 2018, Braun et al., 2018]), applying Frank-Wolfe methods to machine learning problems (e.g., Joulin et al. [2014], Shah et al. [2015], Osokin et al. [2016], Freund et al. [2017], Miech et al. [2017]). Nonetheless, these results left a wide gap between the linear $O(\log 1/\epsilon)$ rate and the sublinear $O(1/\epsilon)$ rate.

Here, we present a new variant of Conditional Gradients that combines the scaling argument of the parameter-free Lazy Frank-Wolfe variant in [Braun et al., 2017, 2018] with scheduled restarts as in the unconstrained case for classical gradient methods [Nemirovskii and Nesterov, 1985, Giselsson and Boyd, 2014, O'Donoghue and Candes, 2015, Fercoq and Qu, 2016, Roulet and d'Aspremont, 2017] to obtain an algorithm that dynamically adapts to the properties of the function and the feasible region. At its core the algorithm relies on a condition similar to sub-analycity and the Łojasiewicz Factorization lemma as in [Bolte et al., 2007] to quantify the impact of restarts as in [Roulet and d'Aspremont, 2017]. Earlier work showed that a sharpness condition derived from the Łojasiewicz lemma (or a related Polyak-Łojasiewicz condition) could be used to improve convergence rates (see Nemirovskii and Nesterov [1985], Bolte et al. [2007], Karimi et al. [2016] for an overview), however these methods required exact knowledge of the corresponding constants appearing in the condition to achieve improved rates. In contrast to this, as in [Roulet and d'Aspremont, 2017], we show that our algorithm does not require knowledge of these constants using robust restarts, thus making it essentially parameter-free.

Contributions

Our contributions can be summarized as follows.

1. *Generalized Strong Convexity.* We define a notion of generalized strong convexity, inspired by the geometric strong convexity of [Lacoste-Julien and Jaggi, 2015] and use the Łojasiewicz Factorization Lemma to show that generalized strong convexity holds generically with appropriate parameters. Depending on the parameters in this condition, our bounds handle standard convexity on one end and (geometric) strong convexity on the other.

2. *Fractional Frank-Wolfe Algorithm.* We then define a new Conditional Gradients algorithm that dynamically adapts to the generalized strong convexity parameters using restarts. The resulting algorithm achieves either sublinear or linear convergence rates depending on the generalized strong convexity parameters and exploits generalized strong convexity to prove $O(1/\epsilon^q)$ rates with $q \leq 1$ depends on the sharpness of the strong Wolfe gap around the optimum, so the function is not required to be strongly convex in the traditional sense. Note, that our algorithm is a modification of the Away-step Frank-Wolfe method but can be immediately adjusted to a similar Pairwise Frank-Wolfe variant.

3. *Robust restarts.* Restart schedules often heavily depend on the value of unknown parameters. We show that because Frank-Wolfe type methods naturally produce a stopping criterion in the form of the Wolfe gap, our restarts are robust, and do not require knowledge of the generalized strong convexity parameters produced by the Łojasiewicz Factorization lemma in order to achieve improved rates.

Also, we would like to mention that our approach generalizes to general Holder-smooth function. However due to space constraints, we leave an in-depth discussion for the full-length version.

Outline

In Section 2 we briefly recall key notions and notation. We then introduce generalized strong convexity, a condition similar to the Polyak-Łojasiewicz condition, in Section 3 and present the Fractional Away-step Frank-Wolfe Algorithm in Section 4. We detail numerical experiments in Section 5.

2 Preliminaries

Consider the following optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{C} \end{array} \tag{1}$$

in the variables $x \in \mathbb{R}^n$, where $\mathcal{C} \subset \mathbb{R}^n$ is a polytope and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function. We assume that the following linear minimization oracle

$$\mathrm{LP}_{\mathcal{C}}(x) \triangleq \operatorname*{argmin}_{z \in \mathcal{C}} x^T z \tag{2}$$

can be computed efficiently. By assumption here, we have $\mathcal{C} = \mathbf{Co}(\mathbf{Ext}(\mathcal{C}))$ where $\mathbf{Co}(.)$ is the convex hull $\mathbf{Ext}(\cdot)$ the set of extreme points, and Carathéodory's theorem shows that every point x of $\mathcal{C}$ can be written as a convex combination of at most $n+1$ points in $\mathbf{Ext}(\mathcal{C})$ although a given representation can contain more such points. We call these points the *support of* x in $\mathcal{C}$. We now define the *strong Wolfe gap* as follows.

Definition 2.1 (Strong Wolfe-gap)**.** *Let f be a smooth convex function, $\mathcal{C}$ a polytope and let $x \in \mathcal{C}$ be arbitrary. Then the* strong Wolfe-gap $w(x)$ over $\mathcal{C}$ *is defined as*

$$w(x) \triangleq \left(\min_{S \in \mathcal{S}_x} \max_{y \in S, z \in \mathcal{C}} \nabla f(x)^T (y - z) \right)_+ + d(x, \mathcal{C}),$$

where $x \in \mathbf{Co}(S)$ and $S_x = \{S \mid S \subset \mathbf{Ext}(\mathcal{C}), x \in \mathbf{Co}(S), |S| \text{ finite}\}$. We also write

$$w(x, S) \triangleq \left(\max_{y \in S, z \in \mathcal{C}} \nabla f(x)^T (y - z) \right)_+ + d(x, \mathcal{C}),$$

given $S \in \mathcal{S}_x$.

Here $d(x, \mathcal{C})$ is the distance from x to $\mathcal{C}$, with $d(x, \mathcal{C}) = 0$ iff $x \in \mathcal{C}$. By construction, we have $w(x) \leq w(x, S)$. Note also that for $x \in \mathcal{C}$, $w(x, S)$ is the sum of the Frank-Wolfe dual gap with the away dual gap in [Lacoste-Julien and Jaggi, 2015]. We first show the following lemma on $w(x, S)$ and $w(x)$.

Lemma 2.2. *Let $x \in \mathcal{C}$ and $S = \{v_i \mid i \in S\}$ with $v_i \in \mathbf{Ext}(\mathcal{C})$ for $i \in S$, be a set so that*

$$x = \sum_{i \in S} \lambda_i v_i, \quad \text{where } \mathbf{1}^T \lambda = 1 \text{ and } \lambda_i > 0 \text{ for } i \in S,$$

then $w(x, S) = 0$ if and only if x is an optimal solution of problem (1)*. In particular, $w(x) = 0$ if and only if x is an optimal solution of problem* (1)*.*

Proof. We can split $w(x, S)$ in two parts, with

$$w(x, S) = \left(\max_{y \in S} \nabla f(x)^T (y - x) + \max_{z \in \mathcal{C}} \nabla f(x)^T (x - z) \right)_+ \tag{3}$$

It is easy to see that both summands are nonnegative if $x \in C$. Here $g(x) \triangleq \max_{z \in \mathcal{C}} \nabla f(x)(x - z)$ is the usual Wolfe gap. When x is an optimal solution of problem (1), first order optimality conditions implies that $\nabla f(x)^T (x - v) \leq 0$ for all $v \in \mathcal{C}$. Since this last quantity is exactly zero when $v = x$, we have $g(x) = 0$.

On the other hand let $h(x) \triangleq \max_{y \in S} \nabla f(x)^T (y - x)$, and suppose x is optimal. If $\nabla f(x) = 0$ we immediately get $h(x) = 0$. Suppose then $\nabla f(x) \neq 0$, since x is optimal, $\nabla f(x)^T (x - v_i) \leq 0$ for all v_i and we can write

$$\begin{aligned} x &= \sum_{\{i : \nabla f(x)^T (x - v_i) = 0\}} \lambda_i v_i + \sum_{\{i : \nabla f(x)^T (x - v_i) < 0\}} \lambda_i v_i \\ &= (1 - \mu) z_1 + \mu z_2 \end{aligned}$$

for some $0 \leq \mu \leq 1$, where $\nabla f(x)^T(x - z_1) = 0$ and $\nabla f(x)^T(x - z_2) < 0$. Now $0 = \nabla f(x)^T(x - x) = \mu \nabla f(x)^T(x - z_2)$ implies $\mu = 0$, hence $\nabla f(x)^T(v_i - x) = 0$ for all $i \in S$, so $h(x) = 0$. Thus we obtain, x optimal implies $w(x) = 0$. Conversely, we have

$$\begin{aligned} f(x) - f^\star &\leq \nabla f(x)^T(x - x^\star) \\ &\leq \max_{z \in \mathcal{C}} \nabla f(x)^T(x - z) \\ &\leq \max_{y \in S, z \in \mathcal{C}} \nabla f(x)^T(y - z) \\ &= w(x, S) \end{aligned}$$

by convexity and the fact that $x \in \mathbf{Co}(S)$. Hence $w(x, S) = 0$ implies x optimal. The corollary on $w(x)$ immediately follows by construction. ■

Finally we recall the definition of curvature in [Lacoste-Julien and Jaggi, 2015], with

$$C_f^A \triangleq \sup_{\substack{x,s,v \in \mathcal{C} \\ \eta \in [0,1] \\ y = x + \eta(s-v)}} \frac{2}{\eta^2}\big(f(y) - f(x) - \eta \langle \nabla f(x), s - v \rangle\big), \tag{4}$$

where f and $\mathcal{C}$ are defined in (1) above.

3 Generalized Strong Convexity

The last part of the proof of Lemma 2.2 above shows that we always have $f(x) - f(x^*) \leq w(x)$. We will now use results on subanalytic functions to refine and potentially improve this bound. We first recall the Łojasiewicz Factorization Lemma and refer the reader to [Bierstone and Milman, 1988, Dedieu, 1992, Bolte et al., 2007] for a primer on subanalytic functions.

Lemma 3.1. *[Łojasiewicz Factorization Lemma] Let $K \subset \mathbb{R}^n$ be a compact set and $f, g : K \to \mathbb{R}$ two continuous (globally) subanalytic functions. If $f^{-1}(0) \subset g^{-1}(0)$, then there exists $c > 0$ and a positive real r such that $|g(x)|^r \leq c|f(x)|$ for all $x \in K$.*

We now show the following generalized strong convexity result.

Lemma 3.2. *[Generalized Strong Convexity] Suppose $f : \mathbb{R}^n \to \mathbb{R}$ in Problem* (1) *is globally subanalytic, then for any compact subset K of $\mathcal{C}$ of Problem* (1) *such that $x^* \in K$, there are $\mu > 0$ and $r > 0$ such that*

$$f(x) - f^\star \leq \mu w(x)^r, \quad \text{for } x \in K \tag{5}$$

where $f^\star$ is the optimal value of Problem (1)*. In particular we choose $r > 0$ as the greatest real such that* (5) *is true on K for some $\mu > 0$.*

Proof. If f is globally subanalytic, then so are the functions $\nabla f(x)^T(y - v)$ and their pointwise maximum over a polytope (see e.g. [Dedieu, 1992, §2.5.6] or [Bierstone and Milman, 1988, Bolte et al., 2007]). The distance function $d(x, \mathcal{C})$ is also globally subanalytic since $\mathcal{C}$ is a polytope [Bierstone and Milman, 1988], hence $w(x)$ is globally subanalytic. Now, let $h(x) = f(x) - f^\star$, Lemma 2.2 shows in particular that $w(x) = 0$ implies $h(x) = 0$, or in other words $w^{-1}(0) \subset h^{-1}(0)$. The Łojasiewicz factorization lemma (see e.g. [Bierstone and Milman, 1988, Th. 6.4] or [Bolte et al., 2007]) then states that there are constants $\mu > 0$ and $r \geq 0$ such that (5) holds (recall that $w(x) \geq 0$). ■

In what follows, we will implicitly choose K so that it contains all iterates. The above generalizes the notion of geometric strong convexity of [Lacoste-Julien and Jaggi, 2015, Th. 6 and Eq. (28)], recovered by choosing $r = 2$ and $\mu = 1/(2\mu^G)$, where μ^G is the geometric strong convexity constant.

Observation 3.3 ($r = 2$ with f strongly convex and $\mathcal{C}$ a polytope)**.** *[Lacoste-Julien and Jaggi, 2015, Theorem 6 in Eq (28)] shows that when f is strongly convex and $\mathcal{C}$ is a polytope then there exists $\mu_f^A > 0$ such that for $x \in \mathcal{C}$*

$$f(x) - f(x^*) \leq \frac{w(x)^2}{2\mu_f^A},$$

which means $r = 2$ in the setting of (5) *here.*

Observation 3.4. *[$1 \leq r \leq 2$] Consider $\mathcal{L} = \{x \in \mathcal{C} \mid w(x) \leq 1\}$. Because we always have $f(x) - f^* \leq w(x)$ on $\mathcal{C}$, we also have $r \geq 1$ in* (5)*. Besides, when f is $L-$smooth, it holds $r \leq 2$.*

4 The Fractional Away-Step Frank-Wolfe Algorithm

Given a polytope $\mathcal{C}$ and a smooth convex function f, let X^* be the set of minimizers of f over $\mathcal{C}$. We now state the Fractional Away-Step Frank-Wolfe algorithm, which can be easily obtained from [Braun et al., 2017], as Algorithm 1. Note that this algorithm is a variant of the Away-Step Frank-Wolfe algorithm, tailored for restarting.

Algorithm 1 Fractional Away-Step Frank-Wolfe Algorithm

Input: A smooth convex function f with curvature $C_f^{\mathcal{A}}$. Starting point $x_0 = \sum_{v \in \mathcal{S}_0} \alpha_0^v v \in \mathcal{C}$ with support $\mathcal{S}_0 \subset \mathbf{Ext}(\mathcal{C})$. LP oracle (2) and schedule parameter $\gamma > 0$.

$t := 0$
while $w(x_t, \mathcal{S}_t) > e^{-\gamma} w(x_0, \mathcal{S}_0)$ **do**
  $v_t := \mathrm{LP}_{\mathcal{C}}(\nabla f(x_t))$ and $d_t^{FW} \triangleq v_t - x_t$
  $s_t := \mathrm{LP}_{S_t}(-\nabla f(x_t))$ with S_t current active set and $d_t^{Away} \triangleq x_t - s_t$
  if $-\nabla f(x_t)^T d_t^{FW} > e^{-\gamma} w(x_0, \mathcal{S}_0)/2$ **then**
    $d_t := d_t^{FW}$ with $\eta_{\max} = 1$
  else
    $d_t := d_t^{Away}$ with $\eta_{\max} = \frac{\alpha_t^{s_t}}{1-\alpha_t^{s_t}}$
  end if
  $x_{t+1} := x_t + \eta_t d_t$ with $\eta_t \in [0, \eta_{\max}]$ via line-search
  Update active set $\mathcal{S}_{t+1}$ and coefficients $\{\alpha_{t+1}^v\}_{v \in \mathcal{S}_{t+1}}$
  $t := t + 1$
end while

Output: $x_t \in \mathcal{C}$ such that $w(x_t, \mathcal{S}_t) \leq e^{-\gamma} w(x_0, \mathcal{S}_0)$

In the following we will call a step a *full-progress step* if it is a Frank-Wolfe Step or an Away Step that is not a drop step, *i.e.,* when $\eta_t < \alpha_{s_t}/(1 - \alpha_{s_t})$. The support $\mathcal{S}_t$ and the weights α_t are updated exactly as in [Lacoste-Julien and Jaggi, 2015, §Away-Steps Frank-Wolfe].

Algorithm 1 depends on a parameter $\gamma > 0$ which explicitly controls the number of iterations needed for the algorithm to stop. In particular, a large value of γ will increase the number of iterations and when γ converges to infinity, Algorithm 1 tends to behave exactly like the classical Frank-Wolfe, (*i.e.,* it never chooses the away direction as an update direction, see Appendix A for a proof).

Proposition 4.1 below gives an upper bound on the number of iterations required for Algorithm 1 to reach a given target gap value $w(x_T, \mathcal{S}_T) \leq w(x_0, \mathcal{S}_0)e^{-\gamma}$. The assumption $e^{-\gamma}w(x_0, \mathcal{S}_0)/2 \leq C_f^A$ in this proposition measures the complexity of a burn-in phase whose cost is marginal as shown in Proposition 4.2.

Proposition 4.1 (Fractional Away-Step Frank-Wolfe Complexity)**.** *Let f be a globally subanalytic, smooth convex function with away curvature C_f^A, satisfying the Generalized Strong Convexity condition in Lemma 3.2 on a compact set K for some $r \geq 1$ and $\mu > 0$. Let $\gamma > 0$ and assume $x_0 \in K$ is such that $e^{-\gamma}w(x_0)/2 \leq C_f^A$. Algorithm 1 outputs an iterate $x_T \in K$ such that*

$$w(x_T, \mathcal{S}_T) \leq w(x_0, \mathcal{S}_0)e^{-\gamma}$$

after at most

$$T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 16e^{2\gamma}C_f^A \mu w(x_0, \mathcal{S}_0)^{r-2}$$

iterations, where $\mathcal{S}_0$ and $\mathcal{S}_T$ are the supports of respectively x_0 and x_T.

Proof. Because of the test criterion in line 5, the update direction d_t satisfies (writing $r_t \triangleq -\nabla f(x_t)$),

$$r_t^T d_t > e^{-\gamma}w(x_0, \mathcal{S}_0)/2 \; .$$

This holds by definition when choosing the FW direction, otherwise (3) yields

$$w(x_t, \mathcal{S}_t) = r_t^T d_t^{FW} + r_t^T d_t^{Away} > e^{-\gamma}w_0,$$

(writing $w_0 \triangleq w(x_0, \mathcal{S}_0)$ to simplify notations) so that

$$r_t^T d_t^{Away} > e^{-\gamma}w_0 - r_t^T d_t^{FW} \geq e^{-\gamma}w_0 - e^{-\gamma}w_0/2 = e^{-\gamma}w_0/2.$$

Using curvature in (4), we have for d_t,

$$f(x_t + \eta d_t) \leq f(x_t) + \eta \nabla f(x_t)^T d_t + \frac{\eta^2}{2}C_f^A \; ,$$

which implies

$$f(x_t) - f(x_t + \eta d_t) \geq \eta r_t^T d_t - \frac{\eta^2}{2}C_f^A \; .$$

We can lower bound progress $f(x_t) - f(x_{t+1})$ with $x_{t+1} = x_t + \eta d_t$ at each iteration for full-progress steps. For Frank-Wolfe steps,

$$\begin{aligned} f(x_t) - f(x_{t+1}) &\geq \max_{\eta \in [0,1]} \Big\{ \eta r_t^T d_t - \frac{\eta^2}{2}C_f^A \Big\} \\ &\geq \max_{\eta \in [0,1]} \Big\{ \eta e^{-\gamma}w_0/2 - \frac{\eta^2}{2}C_f^A \Big\} \end{aligned}$$

Hence because of exact line-search, assuming $e^{-\gamma}w_0/2 \leq C_f^A$ holds,

$$f(x_t) - f(x_{t+1}) \geq \frac{w_0^2}{8C_f^A e^{2\gamma}}. \tag{7}$$

For all away steps, we have

$$f(x_t) - f(x_t + \eta d_t) \geq \max_{\eta \in [0, \eta_{\max}]} \Big\{ \eta e^{-\gamma}w_0/2 - \frac{\eta^2}{2}C_f^A \Big\}.$$

Yet for away steps that are not drop steps, assuming $e^{-\gamma} w_0/2 \leq C_f^A$ again the optimum is obtained for $0 < \eta^* < \eta_{\max}$, and the same conclusion as in (7) for Frank-Wolfe steps follows.

Write $T = T_d + T_f$ the number of iterations for Algorithm 1 to finish. T_d is the number of drop steps, while T_f stands for the number of full-progress steps. Hence we have,

$$\begin{aligned} f(x_0) - f(x_T) &= \sum_{t=0}^{T-1} f(x_t) - f(x_{t+1}) \\ &\geq T_f \frac{w_0^2}{8 C_f^A e^{2\gamma}}. \end{aligned}$$

Because f satisfies (5) on K we have when $x_0 \in K$,

$$f(x_0) - f(x_T) \leq f(x_0) - f(x^*) \leq \mu w(x_0)^r \leq \mu w(x_0, \mathcal{S}_0)^r,$$

by definition of $w(x)$. We then get an upper bound on the number T_f of full-progress steps

$$T_f \leq 8 C_f^A e^{2\gamma} \mu w_0^{r-2} \, .$$

Finally writing $|\mathcal{S}_0|$ (resp. $|\mathcal{S}_T|$) the size of the support of x_0 (resp. x_T), and T_{FW} the number of Frank-Wolfe steps which add a new vertex to an iterate of the Fractional-Away-Step Frank-Wolfe Algorithm, we get $T_{FW} \leq T_f$ and the size of the support $\mathcal{S}_t$ of x_t satisfies $|\mathcal{S}_0| - T_d + T_{FW} = |\mathcal{S}_T|$ hence

$$|\mathcal{S}_0| - |\mathcal{S}_T| + T_f \geq T_d,$$

and we finally get $T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 16 C_f^A e^{2\gamma} \mu w_0^{r-2}$. ■

The following observation shows that the assumption $e^{-\gamma} w(x_0, \mathcal{S}_0)/2 \leq C_f^A$ in Proposition 4.1 has a marginal impact on complexity.

Proposition 4.2 (Burn-in phase)**.** *After at most*

$$8 \frac{e^\gamma}{\gamma} \ln \frac{w(x_0, \mathcal{S}_0)}{2 C_f^A} + |\mathcal{S}_0|,$$

cumulative iterations of Algorithm 1, with constant schedule parameter $\gamma > 0$, we get a point x such that $e^{-\gamma} w(x, \mathcal{S})/2 \leq C_f^A$.

Proof. The proof closely follows that of Proposition 4.1. Suppose that $e^{-\gamma} w_0/2 > C_f^A$ writing again $w_0 = w(x_0, \mathcal{S}_0)$, by curvature for every full progress step we have

$$f(x_t) - f(x_{t+1}) \geq \eta_t e^{-\gamma} w_0/2 - \frac{\eta_t^2 C_f^A}{2} \geq e^{-\gamma} w_0/2 - \frac{C_f^A}{2} \geq e^{-\gamma} w_0/4 \, .$$

Moreover, via the strong Wolfe gap we have

$$f(x_0) - f(x^*) \leq w_0 \, .$$

Writing T the number of iterations of the Algorithm 1 before it stopped, with same notation as in Proposition 4.1, combining the equations above yields

$$T_f e^{-\gamma} w_0/4 \quad \leq \quad f(x_0) - f(x_T) \leq f(x_0) - f(x^*) \leq w_0$$

Hence

$$T_f e^{-\gamma} w_0/4 \leq w_0$$

and $T_f \leq 4e^{\gamma}$. Also

$$T = T_d + T_f \leq 2T_f + |\mathcal{S}_0| - |\mathcal{S}_T| \,,$$

so that

$$T \leq 8e^{\gamma} + |\mathcal{S}_0| \,.$$

Because x_T is the output of Algorithm 1, we have $w(x_T, \mathcal{S}_T) < e^{-\gamma} w_0$. Write N the smallest integer such that $e^{-N\gamma} w_0 \leq 2C_f^A e^{\gamma}$ and $\hat{x}_i$ (for $0 \leq i \leq N$) the output of the i^{th} call to Algorithm 1. It is sufficient that N satisfies

$$N \geq \frac{1}{\gamma} \ln \frac{w_0}{2C_f^A} - 1.$$

Similarly write $i_0 \leq N$ the first integer such that $w(\hat{x}_{i_0}) < 2C_f^A e^{\gamma}$. If $i_0 = N$, each of the first N calls to Algorithm 1 runs in less than $8e^{\gamma} + |\mathcal{S}_{\hat{x}_i}| - |\mathcal{S}_{\hat{x}_{i+1}}|$ iterations. And we finally need at most

$$8\frac{e^{\gamma}}{\gamma} \ln \frac{w_0}{2C_f^A} + |\mathcal{S}_0| \quad \text{iterations.}$$

Otherwise $i_0 < N$ and hence $e^{-i_0\gamma} w_0 \geq C_f^A e^{\gamma}$ from which it follows that

$$i_0 \leq \frac{1}{\gamma} \ln \frac{w_0}{2C_f^A e^{\gamma}} \,,$$

and similarly, each call before the i_0^{th} of Algorithm 1 requires also a bounded number of iterations $8e^{\gamma} + |\mathcal{S}_{\hat{x}_i}| - |\mathcal{S}_{\hat{x}_{i+1}}|$ so that we need at most

$$8\frac{e^{\gamma}}{\gamma} \ln \frac{w(x_0, \mathcal{S}_0)}{2C_f^A e^{\gamma}} + |\mathcal{S}_0| \;\text{ iterations,}$$

which is the desired result. ■

Algorithm 1 can be immediately adapted to a *Fractional Pairwise Frank-Wolfe* variant, we opted for the leaner away-step variant to simplify the exposition.

4.1 Restart Schemes

Consider a point x_{k-1} with strong Wolfe gap $w(x_{k-1}, \mathcal{S}_{k-1})$. Algorithm 1 with parameter $\gamma_k > 0$, outputs a point x_k and we write

$$x_k \triangleq \mathcal{F}(x_{k-1}, w(x_{k-1}, \mathcal{S}_{k-1}), \gamma_k) \,.$$

Following [Roulet and d'Aspremont, 2017] we define *scheduled restarts* for Algorithm 1 as follows.

Algorithm 2 Scheduled restarts for Fractional Away-step Frank-Wolfe

Input: $\tilde{x}_0 \in \mathbb{R}^n$ and a sequence $\gamma_k > 0$ and $\epsilon > 0$.
Burn-in phase: compute x_0 via $8\frac{e^{\gamma}}{\gamma} \ln \frac{w(x_0, \mathcal{S}_0)}{2C_f^A} + |\mathcal{S}_0|$ steps of Algorithm 1.
while $w(x_{k-1}) > \epsilon$ **do**

$$x_k = \mathcal{F}(x_{k-1}, w(x_{k-1}, \mathcal{S}_{k-1}), \gamma_k)$$

end while
Output: $\hat{x} := x_T$

Note that one burn-in phase is sufficient to ensure the condition $e^{-\gamma_i} w(x_{i-1}, \mathcal{S}_{i-1})/2 \leq C_f^A$ at each restart. Algorithm 2 is similar to the restarting schemes in [Roulet and d'Aspremont, 2017, Section 4] when a termination criterion is available. In particular, a choice of a constant $(\gamma_k)_k$ implies linear convergence of the restart schemes for $r = 2$. Note that $1 \leq r \leq 2$.

Theorem 4.3 (Rate for constant restart schemes)**.** *Let f be a globally subanalytic, smooth convex function with away curvature C_f^A, satisfying the Generalized Strong Convexity in Lemma 3.2 on a compact set K with $r \geq 1$ and $\mu > 0$. Let $\gamma > 0$ and assume $x_0 \in K$ is such that $e^{-\gamma} w(x_0, \mathcal{S}_0)/2 \leq C_f^A$. With $\gamma_k = \gamma$, the output of Algorithm 2 satisfies ($h(x) = f(x) - f(x^\star)$)*

$$\begin{cases} h(x_T) \leq w_0 \dfrac{1}{\left(1 + \tilde{T} C_\gamma^r\right)^{\frac{1}{2-r}}} & \text{when } 0 < r < 2 \\ \\ h(x_T) \leq w_0 \exp\left(-\dfrac{\gamma}{e^{2\gamma}} \dfrac{\tilde{T}}{8 C_f^A \mu}\right) & \text{when } r = 2 \,, \end{cases}$$

after T steps, with $w_0 = w(x_0, \mathcal{S}_0)$ and $\tilde{T} \triangleq T - (|\mathcal{S}_0| - |\mathcal{S}_T|)$. Also

$$C_\gamma^r \triangleq \frac{e^{\gamma(2-r)} - 1}{8 e^{2\gamma} C_f^A \mu w(x_0, \mathcal{S}_0)^{r-2}} \,. \tag{15}$$

Proof. Denote by R the number of restarts in Algorithm 1 for T total iterations. From Proposition 4.1, it follows directly that

$$w(x_R, \mathcal{S}_R) \leq w_0 e^{-\gamma R} \,,$$

By Proposition 4.1, the total number T of steps of Algorithms 1 is upper-bounded by

$$T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 8 C_f^A \mu e^{2\gamma} w_0^{r-2} \sum_{i=0}^{R-1} e^{-\gamma i (r-2)} \,.$$

Suppose $r < 2$, we have the following upper bound on T,

$$T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 8 C_f^A \mu e^{2\gamma} w_0^{r-2} \frac{e^{\gamma(2-r)R} - 1}{e^{\gamma(2-r)} - 1}$$

hence

$$e^{-\gamma R} \leq \frac{1}{\left(1 + \tilde{T} C_\gamma^r\right)^{\frac{1}{2-r}}}$$

Hence for $1 \leq r < 2$,

$$w(x_R, \mathcal{S}_R) \leq w_0 \frac{1}{\left(1 + \tilde{T} C_\gamma^r\right)^{\frac{1}{2-r}}} \,,$$

while the case $r = 2$ leads to

$$T \leq |\mathcal{S}_0| - |\mathcal{S}_T| + 8 C_f^A \mu e^{2\gamma} R.$$

and

$$w(x_R, \mathcal{S}_R) \leq w_0 \exp\left(-\gamma \frac{\tilde{T}}{8 C_f^A \mu e^{2\gamma}}\right),$$

which yields the desired result. ■

For $r = 2$, this recovers the bound for the strongly convex case as derived in Lacoste-Julien and Jaggi [2015] and for $r = 1$ we recovers the convergence rate with smoothness assumption only. Note also that for $r \to 2$, we recover the same complexity rates as for $r = 2$

$$\lim_{r \to 2} \frac{1}{\left(1 + \tilde{T} C_\gamma^r\right)^{\frac{1}{2-r}}} = \exp\Big(-\frac{\gamma}{e^{2\gamma}} \frac{\tilde{T}}{8 C_f^A \mu}\Big) \,.$$

The complexity bounds in Theorem 4.3 depend on γ, which controls the convergence rate. Optimal choices of γ depend on r, a constant that we generally do not know nor observe. However, in the following we show that simply picking $\gamma = 1/2$ leads to optimal complexity bounds up to a constant factor.

Proposition 4.4 (Robustness in γ)**.** *Assume f satisfies Generalized Strong Convexity with $r > 0$. Write $\gamma^*(r)$ as the optimal choice of $\gamma > 0$ in the complexity bound of Theorem 4.3. Consider running Algorithm 2 with $\gamma = 1/2$ and same assumptions as in Theorem 4.3, the output $\hat{x}$ satisfies*

$$h(\hat{x}) \leq \sqrt{\frac{e}{e-1}} w_0 \frac{1}{\left(1 + \tilde{T} C^r_{\gamma^*(r)}\right)^{\frac{1}{2-r}}} \quad \textit{when } 0 \leq r < 2 \,.$$

When $r = 2$, we have $\gamma^*(r) = 1/2$.

Proof. When $1 \leq r < 2$, from Theorem 4.3 we have

$$f(x_T) - f(x^\star) \quad \leq w_0 \frac{1}{\left(1 + \tilde{T} C^r_{\gamma}\right)^{\frac{1}{2-r}}} \,.$$

From (15), the optimal value $\gamma^*(r)$ is the maximum of

$$B(\gamma, r) = \left(\frac{e^{\gamma(2-r)} - 1}{e^{2\gamma}}\right),$$

with respect to γ. Hence

$$\gamma^*(r) = \frac{\ln(2) - \ln(r)}{2 - r} \;\; \text{when } 1 \leq r < 2.$$

The function

$$H(r) = \frac{\left(1 + \tilde{T} C^r_{\gamma^*(r)}\right)^{\frac{1}{2-r}}}{\left(1 + \tilde{T} C^r_{1/2}\right)^{\frac{1}{2-r}}}$$

is a decreasing in r, hence

$$\begin{aligned} \frac{1}{\left(1 + \tilde{T} \frac{e^{(2-r)/2} - 1}{e\tilde{C}}\right)^{\frac{1}{2-r}}} \quad &\leq \quad \sqrt{\frac{1 + \frac{\tilde{T}}{\tilde{C}}}{1 + \frac{\tilde{T}}{\tilde{C}} \frac{e-1}{e}}} \frac{1}{\left(1 + \tilde{T} C^r_{\gamma^*(r)}\right)^{\frac{1}{2-r}}} \\ &\leq \quad \sqrt{\frac{e}{e-1}} \frac{1}{\left(1 + \tilde{T} C^r_{\gamma^*(r)}\right)^{\frac{1}{2-r}}}, \end{aligned}$$

where $\tilde{C} \triangleq 8 C_f^A \mu w(x_0, \mathcal{S}_0)^{r-2}$. When $r = 2$, the optimal value of γ is $1/2$, maximizing the function $\gamma / e^{2\gamma}$. ■

5 Numerical Experiments

We test our results on problem instances taken from Lacoste-Julien and Jaggi [2015] and Braun et al. [2017]. Our method is a modification of the classical Away-Step Frank-Wolfe Algorithm (AFW) with restarts, giving better complexity bounds in a much broader setting (i.e. on problems that do not satisfy the gemoetric strong convexity condition), so we expect both methods to have similar behavior in practice, especially given their robustness to the restart parameter γ. This is what we observe in Figure 1. Aside from Figure 1, we benchmark Algorithm 1 against classical Frank-Wolfe on problems where geometric strong convexity does not hold.

We use regression problems with a variety of loss functions (quadratic, powered norm, logistic) on the ℓ_1 ball (Figure 1 and 3), with the same data sets as in Lacoste-Julien and Jaggi [2015], using in particular smooth but non-strongly convex losses. Our regression problem is written

$$l(w) = \frac{1}{\alpha n} \sum_{i=1}^{n} (y_i - x_i^T w)^\alpha \tag{20}$$

for $\alpha \geq 1$.

Each plot contains two graphs, the left one shows primal gap $f(x_t) - f^*$ versus total oracle calls (where f^* is found by running once AFW requiring high precision), while the right one shows the strong-Wolfe gap $w(x_t, \mathcal{S}_t)$ versus total oracle calls. Initialization of all algorithms is made from a random extreme point of the set.

In Figure 1 we observe that as shown by Proposition 4.4, the value of γ in Algorithm 2 does not impact the primal convergence behavior. The classical Away Frank-Wolfe method has very similar behavior.

In Figures 1 and 2, we scaled the constraint sets so that the optimum is not in the strict interior. When the optimum is in the strict interior, already the classic FW algorithm converges linearly [Guélat and Marcotte, 1986] and we expect Algorithm 1 to have a similar performance to the classical FW algorithm, which is what we observe in Figure 3.

Figure 3 compares classical FW and Algorithm 1 with $\gamma = 0.5$ on regression problem (20) with $\alpha = 1.5$, so that the classical geometric strong convexity condition does not hold. Algorithm 1 and Theorem 4.3 in this case guarantee much faster convergence.

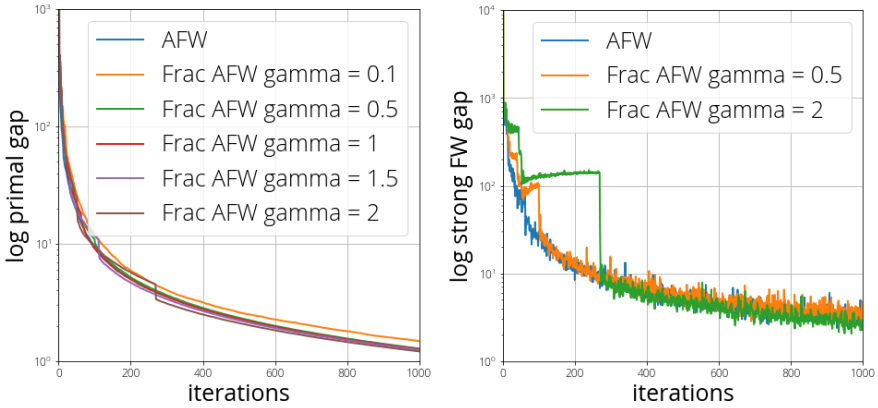

Figure 1: Representative examples on Lasso with various values of γ in restart schemes of algorithm 1. Less instances are displayed on the right diagram because of the oscillating behavior of the strong FW gaps.

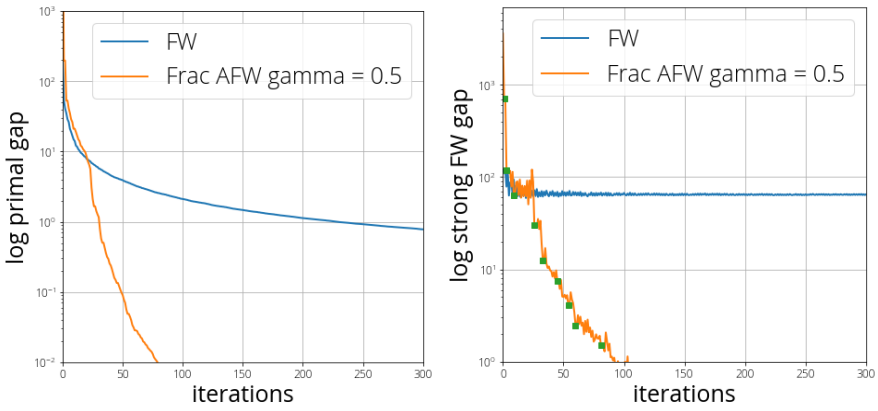

Figure 2: Comparing classical FW and Algorithm 1 with $\gamma = 0.5$ on regression problem (20) with $\alpha = 1.5$, so that the classical geometric strong convexity condition does not hold. Green squares indicate restart times.

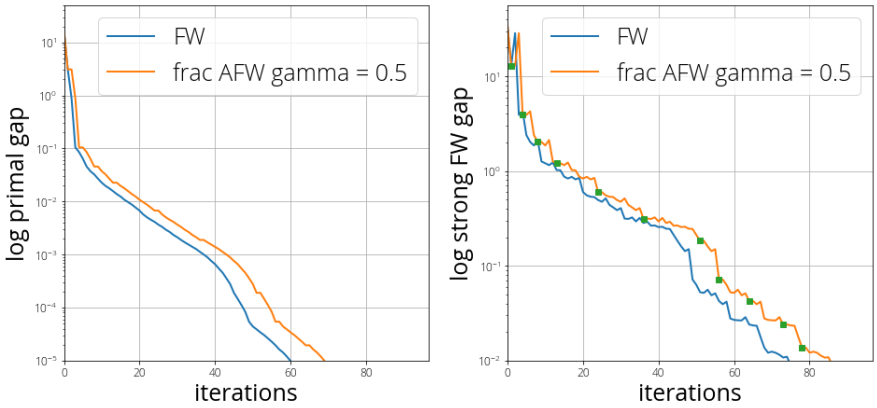

Figure 3: Comparing classical FW and Algorithm 1 with $\gamma = 0.5$ on logistic regression with ℓ_1 constraint, where the constrained minimum lies in the interior of the ball. Here AFW and FW share the very same curve.

6 Conclusion

We derived a variant of the Away-Step Frank-Wolfe algorithm and showed improved complexity bounds when the strong Wolfe gap satisfies a generalized strong convexity condition. The Łojasiewicz factorization lemma shows that this condition actually holds generically for some value of the parameters, producing complexity bounds of the form $O(1/\epsilon^q)$ with $q \leq 1$, thus smoothly interpolating between the complexity of the classical FW algorithm with rate $O(1/\epsilon)$ and that of the Away-Step Frank-Wolfe with rate $O(\log(1/\varepsilon))$. Our method is adaptive to the value of the generalized strong convexity parameters and robustly yields optimal performance. Numerical experiments show that our algorithm is competitive with classical versions of AFW in the geometric strongly convex case and very significantly outperforms FW when geometric strongly convexity does not hold.

References

M. A. Bashiri and X. Zhang. Decomposition-invariant conditional gradient for general polytopes with line search. In *Advances in Neural Information Processing Systems*, pages 2687–2697, 2017.

E. Bierstone and P. D. Milman. Semianalytic and subanalytic sets. *Publications Mathématiques de l'IHÉS*, 67:5–42, 1988.

J. Bolte, A. Daniilidis, and A. Lewis. The lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007.

G. Braun, S. Pokutta, and D. Zink. Lazifying conditional gradient algorithms. *Proceedings of ICML*, 2017.

G. Braun, S. Pokutta, D. Tu, and S. Wright. Blended conditional gradients: the unconditioning of conditional gradients. *arXiv preprint arXiv:1805.07311*, 2018.

J. P. Dedieu. Penalty functions in subanalytic optimization. *Optimization*, 26(1-2):27–32, 1992.

O. Fercoq and Z. Qu. Restarting accelerated gradient methods with a rough strong convexity estimate. *arXiv preprint arXiv:1609.07358*, 2016.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

R. M. Freund and P. Grigas. New analysis and results for the frank–wolfe method. *Mathematical Programming*, 155(1): 199–230, 2016. ISSN 1436-4646. doi: 10.1007/s10107-014-0841-6.

R. M. Freund, P. Grigas, and R. Mazumder. An extended frank–wolfe method with "in-face" directions, and its application to low-rank matrix completion. *SIAM Journal on Optimization*, 27(1):319–346, 2017.

D. Garber and E. Hazan. A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. *arXiv preprint arXiv:1301.4666*, 2013.

D. Garber and O. Meshi. Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. *arXiv preprint, arXiv:1605.06492v1*, May 2016.

D. Garber, S. Sabach, and A. Kaplan. Fast generalized conditional gradient method with applications to matrix recovery problems. *arXiv preprint arXiv:1802.05581*, 2018.

P. Giselsson and S. Boyd. Monotonicity and restart in fast gradient methods. In *53rd IEEE Conference on Decision and Control*, pages 5058–5063. IEEE, 2014.

J. Guélat and P. Marcotte. Some comments on wolfe's 'away step'. *Mathematical Programming*, 35(1):110–119, 1986.

A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *European Conference on Computer Vision*, pages 253–268. Springer, 2014.

H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

T. Kerdreux, F. Pedregosa, and A. d'Aspremont. Frank-wolfe with subsampling oracle. *arXiv preprint arXiv:1803.07348*, 2018.

S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank–Wolfe optimization variants. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 496–504. Curran Associates, Inc., 2015.

S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate frank-wolfe optimization for structural svms. In *ICML 2013 International Conference on Machine Learning*, pages 53–61, 2013.

G. Lan and Y. Zhou. Conditional gradient sliding for convex optimization. *Optimization-Online preprint (4605)*, 2014.

G. Lan, S. Pokutta, Y. Zhou, and D. Zink. Conditional accelerated lazy stochastic gradient descent. *Proceedings of ICML*, 2017.

E. S. Levitin and B. T. Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50, 1966.

A. Miech, J.-B. Alayrac, P. Bojanowski, I. Laptev, and J. Sivic. Learning from video and text via large-scale discriminative

clustering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5267–5276. IEEE, 2017.

A. Nemirovskii and Y. E. Nesterov. Optimal methods of smooth convex minimization. *USSR Computational Mathematics and Mathematical Physics*, 25(2):21–30, 1985.

B. O'Donoghue and E. Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015.

A. Osokin, J.-B. Alayrac, I. Lukasewitz, P. K. Dokania, and S. Lacoste-Julien. Minding the gaps for block frank-wolfe optimization of structured svms. *ICML 2016 International Conference on Machine Learning / arXiv preprint arXiv:1605.09346*, 2016.

V. Roulet and A. d'Aspremont. Sharpness, restart and acceleration. *ArXiv preprint arXiv:1702.03828*, 2017.

N. Shah, V. Kolmogorov, and C. H. Lampert. A multi-plane block-coordinate frank-wolfe algorithm for training structural svms with a costly max-oracle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2737–2745, 2015.

A One shot application of the Fractional Away-step Frank Wolfe

Running once Fractional Away-step Frank-Wolfe with a large value of γ allows to find an approximate minimizer with the desired precision. The following lemma explains the rate of convergence. Importantly the rate does not depend on r. Hence there is no hope of observing linear convergence for the strongly convex case.

Lemma A.1. *Let f be a smooth convex function, $\epsilon > 0$ be a target accuracy, and $x_0 \in \mathcal{C}$ be an initial point. Then for any $\gamma > \ln \frac{w(x_0)}{\epsilon}$, Algorithm 1 satisfies:*

$$f(x_T) - f(x^*) \leq \epsilon,$$

for $T \geq \frac{2C_f^{\mathcal{A}}}{\epsilon}$.

Proof. We can stop the algorithm as soon as the criterion $w(x_t) < \epsilon$ in step 2 is met or we observe an away step, whichever comes first. In former case we have $f(x_t) - f^* \leq w(t) < \epsilon$, in the latter it holds

$$f(x_t) - f^* \leq -\nabla f(x_t)(d_t^{FW}) \leq \epsilon/2 < \epsilon.$$

Thus, when the algorithms stops, we have achieved the target accuracy and it suffices to bound the number of iterations required to achieve that accuracy. Moreover, while running, the algorithm only executes Frank-Wolfe and we drop the FW superscript in the directions; otherwise we would have stopped.

From the proof of Proposition 4.1, we have each Frank-Wolfe step ensures progress of the form

$$f(x_t) - f(x_{t+1}) \geq \begin{cases} \frac{\langle r_t; d_t \rangle^2}{2C_f^{\mathcal{A}}} & \text{if } \langle r_t; d_t \rangle \leq C_f^{\mathcal{A}} \\ \langle r_t; d_t \rangle - C_f^{\mathcal{A}}/2 & \text{otherwise.} \end{cases}$$

For convenience, let $h_t \triangleq f(x_t) - f^*$. By convexity we have $h_t \leq \langle r_t; d_t \rangle$, so that the above becomes

$$f(x_t) - f(x_{t+1}) \geq \begin{cases} \frac{h_t^2}{2C_f^{\mathcal{A}}} & \text{if } h_t \leq C_f^{\mathcal{A}} \\ h_t - C_f^{\mathcal{A}}/2 & \text{otherwise.} \end{cases},$$

and moreover observe that the second case can only happen in the very first step: $h_1 \leq h_0 - (h_0 - C_f^{\mathcal{A}}/2) = C_f^{\mathcal{A}}/2 \leq 2C_f^{\mathcal{A}}/t$ for $t = 1$ providing the start of the following induction: we claim $h_t \leq \frac{2C_f^{\mathcal{A}}}{t}$.

Suppose we have established the bound for t, then for $t + 1$, we have

$$h_{t+1} \leq \left(1 - \frac{h_t}{2C_f^{\mathcal{A}}}\right) h_t \leq \frac{2C_f^{\mathcal{A}}}{t} - \frac{2C_f^{\mathcal{A}}}{t^2} \leq \frac{2C_f^{\mathcal{A}}}{t+1}.$$

Therefore the induction is complete and it follows that the algorithm requires $T \geq \frac{2C_f^{\mathcal{A}}}{\epsilon}$ to reach ϵ-accuracy. ■