

Boosting Frank-Wolfe by Chasing Gradients

Cyrille W. Combettes

*School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA, USA*

cyrille@gatech.edu

Sebastian Pokutta

*Institute of Mathematics and Department for AI in Society, Science, and Technology
Technische Universität Berlin and Zuse Institute Berlin
Berlin, Germany*

pokutta@zib.de

Abstract

The Frank-Wolfe algorithm has become a popular first-order optimization algorithm for it is simple and projection-free, and it has been successfully applied to a variety of real-world problems. Its main drawback however lies in its convergence rate, which can be excessively slow due to naive descent directions. We propose to speed up the Frank-Wolfe algorithm by better aligning the descent direction with that of the negative gradient via a subroutine. This subroutine chases the negative gradient direction in a matching pursuit-style while still preserving the projection-free property. Although the approach is reasonably natural, it produces very significant results. We derive convergence rates $\mathcal{O}(1/t)$ to $\mathcal{O}(e^{-\omega t})$ of our method and we demonstrate its competitive advantage both per iteration and in CPU time over the state-of-the-art in a series of computational experiments.

1 Introduction

Let $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ be a Euclidean space. In this paper, we address the constrained convex optimization problem

$$\min_{x \in \mathcal{C}} f(x) \quad (1)$$

where $f : \mathcal{H} \rightarrow \mathbb{R}$ is a smooth convex function and $\mathcal{C} \subset \mathcal{H}$ is a compact convex set. A natural approach to solving Problem (1) is to apply any efficient method that works in the unconstrained setting and add projections back onto \mathcal{C} when the iterates leave the feasible region. However, there are situations where projections can be very expensive while linear minimizations over \mathcal{C} are much cheaper. For example, if $\mathcal{C} = \{X \in \mathbb{R}^{m \times n} \mid \|X\|_{\text{nuc}} \leq \tau\}$ is a nuclear norm-ball, a projection onto \mathcal{C} requires computing an SVD, which has complexity $\mathcal{O}(mn \min\{m, n\})$, while a linear minimization over \mathcal{C} requires only computing the pair of top singular vectors, which has complexity $\mathcal{O}(\text{nnz})$ where nnz denotes the number of nonzero entries. Other examples include the flow polytope, the Birkhoff polytope, the matroid polytope, or the set of rotations; see, e.g., Hazan and Kale (2012).

In these situations, the Frank-Wolfe algorithm (FW) (Frank and Wolfe, 1956), a.k.a. conditional gradient algorithm (Levitin and Polyak, 1966), becomes the method of choice, as it is a simple projection-free algorithm relying on a linear minimization oracle over \mathcal{C} . At each iteration, it calls the oracle $v_t \leftarrow \arg \min_{v \in \mathcal{C}} \langle \nabla f(x_t), v \rangle$ and moves in the direction of this vertex, ensuring that the new iterate $x_{t+1} \leftarrow x_t + \gamma_t(v_t - x_t)$ is feasible by convex combination, with a step-size $\gamma_t \in [0, 1]$. Hence, FW can be seen as a projection-free variant of projected gradient descent trading the gradient descent direction $-\nabla f(x_t)$ for the vertex direction $v_t - x_t$ minimizing the linear approximation of f at x_t over \mathcal{C} . FW has been applied to traffic assignment problems (LeBlanc et al., 1975), low-rank matrix approximation (Shalev-Shwartz et al., 2011), structural SVMs (Lacoste-Julien et al., 2013), video co-localization (Joulin et al., 2014), infinite RBMs (Ping et al., 2016), and, e.g., adversarial learning (Chen et al., 2020).

The main drawback of FW is that the modified descent direction leads to a sublinear convergence rate $\mathcal{O}(1/t)$, which cannot be improved upon in general as an asymptotic lower bound $\Omega(1/t^{1+\delta})$ holds for any $\delta > 0$ (Canon and Cullum, 1968). More recently, Jaggi (2013) provided a simple illustration of the phenomenon: if $f : x \in \mathbb{R}^n \mapsto \|x\|_2^2$ is the squared ℓ_2 -norm and $\mathcal{C} = \Delta_n$ is the standard simplex, then the primal gap at iteration $t \in \llbracket 1, n \rrbracket$ is lower bounded by $1/t - 1/n$; see also Lan (2013) for a lower bound $\Omega(LD^2/t)$ on an equivalent setup, exhibiting an explicit dependence on the smoothness constant L of f and the diameter D of \mathcal{C} .

Hence, a vast literature has been devoted to the analysis of higher convergence rates of FW if additional assumptions on the properties of f , the geometry of \mathcal{C} , or the location of $\arg \min_{\mathcal{C}} f$ are met. Important contributions include:

- (i) $\mathcal{O}(e^{-\omega t})$ if \mathcal{C} is strongly convex and $\inf_{\mathcal{C}} \|\nabla f\| > 0$ (Levitin and Polyak, 1966),
- (ii) $\mathcal{O}(e^{-\omega t})$ if f is strongly convex and $\arg \min_{\mathcal{C}} f \subset \text{relint}(\mathcal{C})$ (Guélat and Marcotte, 1986),
- (iii) $\mathcal{O}(1/t^2)$ if f is gradient dominated and \mathcal{C} is strongly convex (Garber and Hazan, 2015).

More recently, several variants to FW have been proposed, achieving linear convergence rates without excessively increasing the per-iteration complexity. These include the following:

- (i) $\mathcal{O}(e^{-\omega t})$ when f is strongly convex and \mathcal{C} is a polytope (Garber and Hazan, 2016; Lacoste-Julien and Jaggi, 2015; Braun et al., 2019),
- (ii) $\mathcal{O}(e^{-\omega t})$ with constants depending on the sparsity of the solution when f is strongly convex and \mathcal{C} is a polytope, of the form $\{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ with vertices in $\{0, 1\}^n$ (Garber and Meshi, 2016), or of arbitrary form (Bashiri and Zhang, 2017).

Contributions. We propose the *Boosted Frank-Wolfe* algorithm (BoostFW), a new and intuitive method speeding up the Frank-Wolfe algorithm by chasing the negative gradient direction $-\nabla f(x_t)$ via a matching pursuit-style subroutine, and moving in this better aligned direction. BoostFW thereby mimics gradient descent while remaining projection-free. We derive convergence rates $\mathcal{O}(1/t)$ to $\mathcal{O}(e^{-\omega t})$. Although the linear minimization oracle may be called multiple times per iteration, we demonstrate in a series of computational experiments the competitive advantage both per iteration and in CPU time of our method over the state-of-the-art. Furthermore, BoostFW does not require line search to achieve strong empirical performance, and it does not need to maintain the decomposition of the iterates. Naturally, our approach can also be used to boost the performance of any Frank-Wolfe-style algorithm.

Outline. We start with notation and definitions and we present some background material on the Frank-Wolfe algorithm (Section 2). We then move on to the intuition behind the design of the Boosted Frank-Wolfe algorithm and present its convergence analysis (Section 3). We validate the advantage of our approach in a series of computational experiments (Section 4). Finally, a couple of remarks conclude the paper (Section 5). All proofs are available in Appendix D. The Appendix further contains complementary plots (Appendix A), an application of our approach to boost the Decomposition-Invariant Pairwise Conditional Gradient algorithm (DICG) (Garber and Meshi, 2016) (Appendix B), and the convergence analysis of the line search-free Away-Step Frank-Wolfe algorithm (Appendix C). We were later informed that the latter analysis was already derived by Pedregosa et al. (2020) in a more general setting.

2 Preliminaries

We work in a Euclidean space $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ equipped with the induced norm $\|\cdot\|$. Let $\mathcal{C} \subset \mathcal{H}$ be a nonempty compact convex set. If \mathcal{C} is a polytope, let \mathcal{V} be its set of vertices. Else, slightly abusing notation, we refer to any point in $\mathcal{V} := \partial\mathcal{C}$ as a *vertex*. We denote by $D := \max_{x, y \in \mathcal{C}} \|y - x\|$ the diameter of \mathcal{C} .

2.1 Notation and definitions

For any $i, j \in \mathbb{N}$ satisfying $i \leq j$, the brackets $\llbracket i, j \rrbracket$ denote the set of integers between (and including) i and j . The indicator function for an event A is $\mathbb{1}_A := 1$ if A is true else 0. For any $x \in \mathbb{R}^n$ and $i \in \llbracket 1, n \rrbracket$, $[x]_i$ denotes the i -th entry of x . Given $p \geq 1$, the ℓ_p -norm in \mathbb{R}^n is $\|\cdot\|_p : x \in \mathbb{R}^n \mapsto (\sum_{i=1}^n |[x]_i|^p)^{1/p}$ and the closed ℓ_p -ball of radius $\tau > 0$ is $\mathcal{B}_p(\tau) := \{x \in \mathbb{R}^n \mid \|x\|_p \leq \tau\}$. The standard simplex in \mathbb{R}^n is $\Delta_n := \{x \in \mathbb{R}^n \mid \mathbf{1}^\top x = 1, x \geq 0\} = \text{conv}(e_1, \dots, e_n)$ where $\{e_1, \dots, e_n\}$ denotes the standard basis, i.e., $e_i = (\mathbb{1}_{\{1=i\}}, \dots, \mathbb{1}_{\{n=i\}})^\top$. The conical hull of a nonempty set $\mathcal{A} \subseteq \mathcal{H}$ is $\text{cone}(\mathcal{A}) := \{\sum_{k=1}^K \lambda_k a_k \mid K \in \mathbb{N} \setminus \{0\}, \lambda_1, \dots, \lambda_K \geq 0, a_1, \dots, a_K \in \mathcal{A}\}$. The number of its elements is denoted by $|\mathcal{A}|$.

Let $f : \mathcal{H} \rightarrow \mathbb{R}$ be a differentiable function. We say that f is:

(i) L -smooth if $L > 0$ and for all $x, y \in \mathcal{H}$,

$$f(y) - f(x) - \langle \nabla f(x), y - x \rangle \leq \frac{L}{2} \|y - x\|^2,$$

(ii) S -strongly convex if $S > 0$ and for all $x, y \in \mathcal{H}$,

$$f(y) - f(x) - \langle \nabla f(x), y - x \rangle \geq \frac{S}{2} \|y - x\|^2,$$

(iii) μ -gradient dominated if $\mu > 0$, $\arg \min_{\mathcal{H}} f \neq \emptyset$, and for all $x \in \mathcal{H}$,

$$f(x) - \min_{\mathcal{H}} f \leq \frac{\|\nabla f(x)\|^2}{2\mu}.$$

Note that although Definition (iii) is defined with respect to the global optimal value $\min_{\mathcal{H}} f$, the bound holds for the primal gap of f on any compact set $\mathcal{C} \subset \mathcal{H}$:

$$f(x) - \min_{\mathcal{C}} f \leq f(x) - \min_{\mathcal{H}} f \leq \frac{\|\nabla f(x)\|^2}{2\mu}.$$

Definition (iii) is also commonly referred to as the Polyak-Łojasiewicz inequality or PL inequality (Polyak, 1963; Łojasiewicz, 1963). It is a *local* condition, weaker than that of strong convexity (Fact 2.1), but it can still provide linear convergence rates for non-strongly convex functions (Karimi et al., 2016). For example, the least squares loss $x \in \mathbb{R}^n \mapsto \|Ax - b\|_2^2$ where $A \in \mathbb{R}^{m \times n}$ and $\text{rank}(A) = m < n$ is not strongly convex, however it is gradient dominated (Garber and Hazan, 2015). See also the Kurdyka-Łojasiewicz inequality (Kurdyka, 1998; Łojasiewicz, 1963) for a generalization to nonsmooth optimization (Bolte et al., 2017).

Fact 2.1. Let $f : \mathcal{H} \rightarrow \mathbb{R}$ be S -strongly convex. Then f is S -gradient dominated.

2.2 The Frank-Wolfe algorithm

The Frank-Wolfe algorithm (FW) (Frank and Wolfe, 1956), a.k.a. conditional gradient algorithm (Levitin and Polyak, 1966), is presented in Algorithm 1. It is a simple first-order projection-free algorithm relying on a linear minimization oracle over \mathcal{C} . At each iteration, it minimizes over \mathcal{C} the linear approximation of f at x_t , i.e., $\ell_f(x_t) : z \in \mathcal{C} \mapsto f(x_t) + \langle \nabla f(x_t), z - x_t \rangle$, by calling the oracle (Line 2) and moves in that direction by convex combination (Line 3). Hence, the new iterate x_{t+1} is guaranteed to be feasible by convexity and there is no need to use projections back onto \mathcal{C} . In short, FW solves Problem (1) by minimizing a sequence of linear approximations of f over \mathcal{C} .

Note that FW has access to the feasible region \mathcal{C} only via the linear minimization oracle, which receives any $c \in \mathcal{H}$ as input and outputs a point $v \in \arg \min_{z \in \mathcal{C}} \langle c, z \rangle = \arg \min_{v \in \mathcal{V}} \langle c, v \rangle$. For example, if $\mathcal{H} = \mathbb{R}^n$ and $\mathcal{C} = \{x \in \mathbb{R}^n \mid \|x\|_1 \leq \tau\}$ is an ℓ_1 -ball, then $\mathcal{V} = \{\pm \tau e_1, \dots, \pm \tau e_n\}$ so the linear minimization oracle simply picks the coordinate e_i with the largest absolute magnitude $|[c]_i|$ and returns $-\text{sign}([c]_i) \tau e_i$. In this case,

Algorithm 1 Frank-Wolfe (FW)

Input: Start point $x_0 \in \mathcal{C}$, step-size strategy $\gamma_t \in [0, 1]$.**Output:** Point $x_T \in \mathcal{C}$.

- 1: **for** $t = 0$ **to** $T - 1$ **do**
 - 2: $v_t \leftarrow \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_t), v \rangle$ ▷ FW oracle
 - 3: $x_{t+1} \leftarrow x_t + \gamma_t(v_t - x_t)$
 - 4: **end for**
-

FW accesses \mathcal{C} only by reading coordinates. Some other examples are covered in the experiments (Section 4).

The general convergence rate of FW is $\mathcal{O}(LD^2/t)$, where L is the smoothness constant of f and D is the diameter of \mathcal{C} (Levitin and Polyak, 1966; Jaggi, 2013). There are different step-size strategies possible to achieve this rate. The default strategy is $\gamma_t \leftarrow 2/(t + 2)$. It is very simple to implement but it does not guarantee progress at each iteration. The next strategy, sometimes referred to as the *short step* strategy and which does make FW a descent algorithm, is $\gamma_t \leftarrow \min\{\langle \nabla f(x_t), x_t - v_t \rangle / (L\|x_t - v_t\|^2), 1\}$. It minimizes the quadratic smoothness upper bound on f . If $\varepsilon_t := f(x_t) - \min_{\mathcal{C}} f$ denotes the primal gap, then

$$\varepsilon_{t+1} \leq \begin{cases} \varepsilon_t - \frac{\langle \nabla f(x_t), x_t - v_t \rangle^2}{2L\|x_t - v_t\|^2} & \text{if } \gamma_t < 1 \\ \varepsilon_t/2 & \text{if } \gamma_t = 1. \end{cases}$$

As we can already see here, a quadratic improvement in progress is obtained if the direction $v_t - x_t$ in which FW moves is better aligned with that of the negative gradient $-\nabla f(x_t)$. The third step-size strategy is a line search $\gamma_t \leftarrow \arg \min_{\gamma \in [0, 1]} f(x_t + \gamma(v_t - x_t))$. It is the most expensive strategy but it does not require (approximate) knowledge of L and it often yields more progress per iteration in practice.

3 Boosting Frank-Wolfe

3.1 Motivation

Suppose that \mathcal{C} is a polytope and that the set of global minimizers $\arg \min_{\mathcal{H}} f$ lies on a lower dimensional face. Then FW can be very slow to converge as it is allowed only to follow vertex directions. As a simple illustration, consider the problem of minimizing $f : x \in \mathbb{R}^2 \mapsto \|x\|_2^2/2$ over the convex hull of $\{(-1, 0)^\top, (1, 0)^\top, (0, 1)^\top\}$, starting from $x_0 = (0, 1)^\top$. The minimizer is $x^* = (0, 0)^\top$. We computed the first iterates of FW and we present their trajectory in Figure 1. We can see that the iterates try to reach x^* by moving towards vertices but clearly these directions $v_t - x_t$ are inadequate as they become orthogonal to $x^* - x_t$.

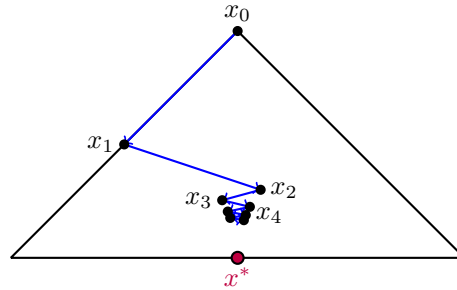


Figure 1: FW yields an inefficient zig-zagging trajectory towards the minimizer.

To remedy this phenomenon, Wolfe (1970) proposed the Away-Step Frank-Wolfe algorithm (AFW), a

variant of FW that allows to move *away* from vertices. The issue in Figure 1 is that the iterates are held back by the weight of vertex x_0 in their convex decomposition. Figure 2 shows that AFW is able to remove this weight and thereby to converge much faster to x^* . In fact, [Lacoste-Julien and Jaggi \(2015\)](#) established that AFW with line search converges at a linear rate $\mathcal{O}(LD^2 \exp(-(S/(8L))(W/D)^2t))$ for S -strongly convex functions over polytopes, where W is the *pyramidal width* of the polytope.

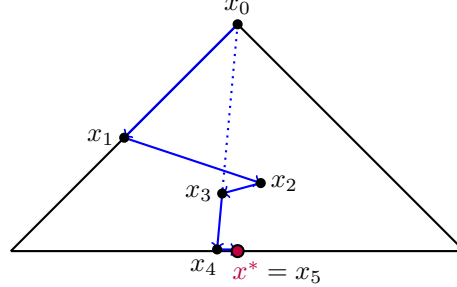


Figure 2: AFW breaks the zig-zagging trajectory by performing away steps. Here, x_4 is obtained using an away step which enables $x_5 = x^*$, speeding up the algorithm considerably.

However, these descent directions are still not as favorable as those of gradient descent, the pyramidal width is a dimension-dependent quantity, and AFW further requires to maintain the decomposition of the iterates onto \mathcal{V} which can become very expensive both in memory usage and computation time ([Garber and Meshi, 2016](#)). Thus, we aim at improving the FW descent direction by directly estimating the gradient descent direction $-\nabla f(x_t)$ using \mathcal{V} , in order to maintain the projection-free property. Suppose that $-\nabla f(x_t) \in \text{cone}(\mathcal{V} - x_t)$ and that we are able to compute its conical decomposition, i.e., we have $-\nabla f(x_t) = \sum_{k=0}^{K_t-1} \lambda_k (v_k - x_t)$ where $\lambda_0, \dots, \lambda_{K_t-1} > 0$ and $v_0, \dots, v_{K_t-1} \in \mathcal{V}$. Then by normalizing by $\Lambda_t := \sum_{k=0}^{K_t-1} \lambda_k$, we obtain a *feasible* descent direction $g_t := (1/\Lambda_t) \sum_{k=0}^{K_t-1} \lambda_k (v_k - x_t)$ in the sense that $[x_t, x_t + g_t] \subseteq \mathcal{C}$. Therefore, building x_{t+1} as a convex combination of x_t and $x_t + g_t$ ensures that $x_{t+1} \in \mathcal{C}$ and the projection-free property holds as in a typical FW step, all the while moving in the direction of the negative gradient $-\nabla f(x_t)$.

3.2 Boosting via gradient pursuit

In practice however, computing the exact conical decomposition of $-\nabla f(x_t)$, even when this is feasible, is not necessary and it may be overkill. Indeed, all we want is to find a descent direction g_t using \mathcal{V} that is *better* aligned with $-\nabla f(x_t)$ and we do not mind if $\|-\nabla f(x_t) - g_t\|$ is arbitrarily large. Thus, we propose to chase the direction of $-\nabla f(x_t)$ by sequentially picking up vertices in a matching pursuit-style ([Mallat and Zhang, 1993](#)). The procedure is described in Algorithm 2 (Lines 3-19). In fact, it implicitly addresses the cone constrained quadratic optimization subproblem

$$\min_{d \in \text{cone}(\mathcal{V} - x_t)} \frac{1}{2} \| -\nabla f(x_t) - d \|^2 \quad (2)$$

via the Non-Negative Matching Pursuit algorithm (NNMP) ([Locatello et al., 2017](#)), without however the aim of solving it. At each round k , the procedure looks to reduce the residual r_k by subtracting its projection $\lambda_k u_k$ onto the principal component u_k . The comparison $\langle r_k, v_k - x_t \rangle$ vs. $\langle r_k, -d_k / \|d_k\| \rangle$ in Line 9 is less intuitive than the rest of the procedure but it is necessary to ensure convergence; see [Locatello et al. \(2017\)](#). The normalization in Line 21 ensures the feasibility of the new iterate x_{t+1} .

Since we are only interested in the direction of $-\nabla f(x_t)$, the stopping criterion in the procedure (Line 12) is an alignment condition between $-\nabla f(x_t)$ and the current estimated direction d_k , which serves as descent direction for BoostFW. The function `align`, defined in (3), measures the alignment between a target direction

Algorithm 2 Boosted Frank-Wolfe (BoostFW)

Input: Input point $y \in \mathcal{C}$, maximum number of rounds $K \in \mathbb{N} \setminus \{0\}$, alignment improvement tolerance $\delta \in]0, 1[$, step-size strategy $\gamma_t \in [0, 1]$.

Output: Point $x_T \in \mathcal{C}$.

```
1:  $x_0 \leftarrow \arg \min_{v \in \mathcal{V}} \langle \nabla f(y), v \rangle$ 
2: for  $t = 0$  to  $T - 1$  do
3:    $d_0 \leftarrow 0$ 
4:    $\Lambda_t \leftarrow 0$ 
5:   flag  $\leftarrow$  false
6:   for  $k = 0$  to  $K - 1$  do
7:      $r_k \leftarrow -\nabla f(x_t) - d_k$  ▷  $k$ -th residual
8:      $v_k \leftarrow \arg \max_{v \in \mathcal{V}} \langle r_k, v \rangle$  ▷ FW oracle
9:      $u_k \leftarrow \arg \max_{u \in \{v_k - x_t, -d_k / \|d_k\|\}} \langle r_k, u \rangle$ 
10:     $\lambda_k \leftarrow \frac{\langle r_k, u_k \rangle}{\|u_k\|^2}$ 
11:     $d'_k \leftarrow d_k + \lambda_k u_k$ 
12:    if  $\text{align}(-\nabla f(x_t), d'_k) - \text{align}(-\nabla f(x_t), d_k) \geq \delta$  then
13:       $d_{k+1} \leftarrow d'_k$ 
14:       $\Lambda_t \leftarrow \begin{cases} \Lambda_t + \lambda_k & \text{if } u_k = v_k - x_t \\ \Lambda_t(1 - \lambda_k / \|d_k\|) & \text{if } u_k = -d_k / \|d_k\| \end{cases}$ 
15:    else
16:      flag  $\leftarrow$  true
17:      break ▷ exit  $k$ -loop
18:    end if
19:  end for
20:   $K_t \leftarrow k$  if flag = true else  $K$ 
21:   $g_t \leftarrow d_{K_t} / \Lambda_t$  ▷ normalization
22:   $x_{t+1} \leftarrow x_t + \gamma_t g_t$ 
23: end for
```

$d \in \mathcal{H} \setminus \{0\}$ and its estimate $\hat{d} \in \mathcal{H}$. It is invariant by scaling of d or \hat{d} , and the higher the value, the better the alignment:

$$\text{align}(d, \hat{d}) := \begin{cases} \frac{\langle d, \hat{d} \rangle}{\|d\| \|\hat{d}\|} & \text{if } \hat{d} \neq 0 \\ -1 & \text{if } \hat{d} = 0. \end{cases} \quad (3)$$

In order to optimize the trade-off between progress and complexity per iteration, we allow for (very) inexact alignments and we stop the procedure as soon as *sufficient* progress is not met (Lines 15-17). Furthermore, note that it is not possible to obtain a perfect alignment when $-\nabla f(x_t) \notin \text{cone}(\mathcal{V} - x_t)$, but this is not an issue as we only seek to better align the descent direction. The number of pursuit rounds at iteration t is denoted by K_t (Line 20). In the experiments (Section 4), we typically set $\delta \leftarrow 10^{-3}$ and $K \leftarrow +\infty$; the role of K is only to cap the number of pursuit rounds per iteration when the FW oracle is particularly expensive (see Section 4.3). Note that if $K = 1$ then BoostFW reduces to FW.

In the case of Figures 1-2, BoostFW exactly estimates the direction of $-\nabla f(x_0) = -(x_0 - x^*)$ in only two rounds and converges in 1 iteration. A more general illustration of the procedure is presented in Figure 3. See also Appendix A.2 for an illustration of the improvements in alignment of d_k during the procedure. Lastly, note that BoostFW does not need to maintain the decomposition of the iterates, which is very favorable in practice (Garber and Meshi, 2016).

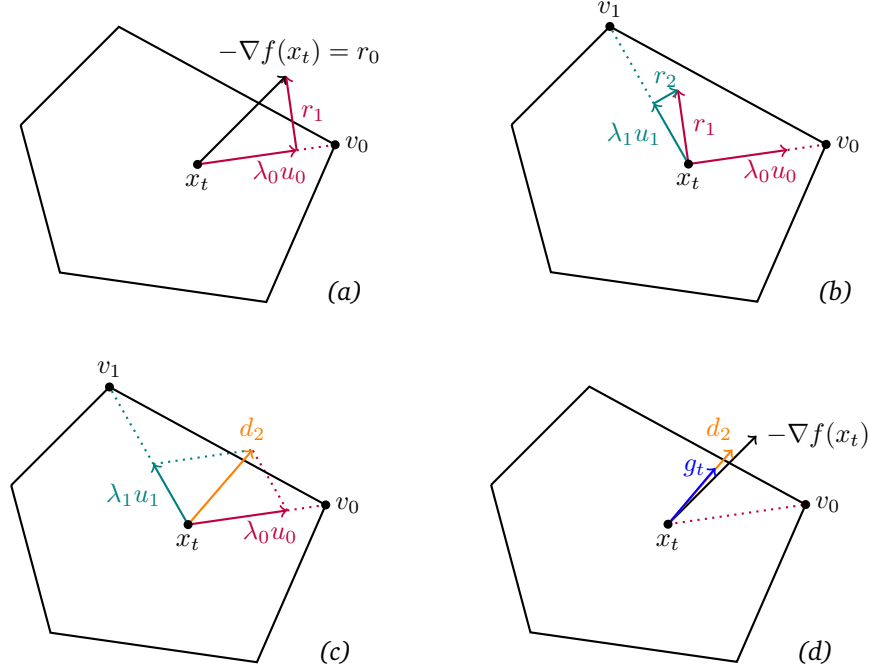


Figure 3: The gradient pursuit procedure builds a descent direction g_t better aligned with the negative gradient direction $-\nabla f(x_t)$, while the FW descent direction is that of $v_0 - x_t$. We have $g_t = d_2 / (\lambda_0 + \lambda_1)$ where $d_2 = \lambda_0 u_0 + \lambda_1 u_1$, $u_0 = v_0 - x_t$, and $u_1 = v_1 - x_t$. Furthermore, note that $[x_t, x_t + d_2] \not\subseteq \mathcal{C}$ but $[x_t, x_t + g_t] \subseteq \mathcal{C}$. Moving along the segment $[x_t, x_t + g_t]$ ensures feasibility of the new iterate x_{t+1} .

We present in Proposition 3.1 some properties satisfied by BoostFW (Algorithm 2). Proofs are available in Appendix D.2.

Proposition 3.1. *For all $t \in \llbracket 0, T-1 \rrbracket$,*

- (i) d_1 is defined and $K_t \geq 1$,
- (ii) $\lambda_0, \dots, \lambda_{K_t-1} \geq 0$,
- (iii) $d_k \in \text{cone}(\mathcal{V} - x_t)$ for all $k \in \llbracket 0, K_t \rrbracket$,
- (iv) $x_t + g_t \in \mathcal{C}$ and $x_{t+1} \in \mathcal{C}$,
- (v) $\text{align}(-\nabla f(x_t), g_t) \geq \text{align}(-\nabla f(x_t), v_t - x_t) + (K_t - 1)\delta$ where $v_t \in \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_t), v \rangle$ and $\text{align}(-\nabla f(x_t), v_t - x_t) \geq 0$.

3.3 Convergence analysis

We denote by $\eta_t := \text{align}(-\nabla f(x_t), g_t)$. We provide in Theorem 3.2 the general convergence rate of BoostFW. All proofs are available in Appendix D.3. Note that $\eta_t \|\nabla f(x_t)\| / (L \|g_t\|) = \langle -\nabla f(x_t), g_t \rangle / (L \|g_t\|^2)$ corresponds to the short step strategy.

Theorem 3.2 (Universal rate). *Let $f : \mathcal{H} \rightarrow \mathbb{R}$ be L -smooth, convex, and μ -gradient dominated, and set $\gamma_t \leftarrow \min\{\eta_t \|\nabla f(x_t)\| / (L \|g_t\|), 1\}$ or $\gamma_t \leftarrow \arg \min_{\gamma \in [0,1]} f(x_t + \gamma g_t)$. Then for all $t \in \llbracket 0, T \rrbracket$,*

$$f(x_t) - \min_{\mathcal{C}} f \leq \frac{LD^2}{2} \prod_{s=0}^{t-1} \left(1 - \eta_s^2 \frac{\mu}{L}\right)^{\mathbb{1}_{\{\gamma_s < 1\}}} \left(1 - \frac{\|g_s\|}{2\|v_s - x_s\|}\right)^{\mathbb{1}_{\{\gamma_s = 1\}}}$$

where $v_s \in \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_s), v \rangle$ for all $s \in \llbracket 0, T-1 \rrbracket$.

Strictly speaking, the rate in Theorem 3.2 is not *explicit* although it still provides a quantitative estimation. Note that $\gamma_t = 1$ is extremely rare in practice, and we observed no more than 1 such iteration in each of the experiments (Section 4). This is a similar phenomenon to that in the Away-Step and Pairwise Frank-Wolfe algorithms (Lacoste-Julien and Jaggi, 2015). Similarly, $K_t > 1$ simply means that it is possible to increase the alignment by δ twice and consecutively, where δ is typically set to a low value. In the experiments, we set $\delta \leftarrow 10^{-3}$ and we observed $K_t > 1$ (or even $K_t > 5$) almost everytime.

For completeness, we disregard these observations and address in Theorem 3.3 the case where the number of iterations with $\gamma_t < 1$ and $K_t > 1$ is not dominant, and we add a minor adjustment to Algorithm 2: if $\gamma_t = 1$ then we choose to do a simple FW step, i.e., to move in the direction of $v_{k=0} - x_t$ instead of the direction of g_t , where $v_{k=0}$ is computed in the first round of the procedure (Line 8). Although this usually provides less progress, we do it for the sole purpose of presenting a *fully explicit* convergence rate; again, there is no need for such tweaks in practice as typically almost every iteration satisfies $\gamma_t < 1$ and $K_t > 1$. Theorem 3.3 states the convergence rate for this scenario, which is very loose as it accommodates for these FW steps.

Theorem 3.3 (Worst-case rate). *Let $f : \mathcal{H} \rightarrow \mathbb{R}$ be L -smooth, convex, and μ -gradient dominated, and set $\gamma_t \leftarrow \min\{\eta_t \|\nabla f(x_t)\| / (L\|g_t\|), 1\}$ or $\gamma_t \leftarrow \arg \min_{\gamma \in [0,1]} f(x_t + \gamma g_t)$. Consider Algorithm 2 with the minor adjustment $x_{t+1} \leftarrow x_t + \gamma'_t(v_t - x_t)$ in Line 22 when $\gamma_t = 1$, where $v_t \leftarrow v_{k=0}$ is computed in Line 8 and $\gamma'_t \leftarrow \min\{\langle \nabla f(x_t), x_t - v_t \rangle / (L\|x_t - v_t\|^2), 1\}$ or $\gamma'_t \leftarrow \arg \min_{\gamma \in [0,1]} f(x_t + \gamma(v_t - x_t))$. Then for all $t \in \llbracket 0, T \rrbracket$,*

$$f(x_t) - \min_c f \leq \frac{4LD^2}{t+2}.$$

We now provide in Theorem 3.4 the more realistic convergence rate of BoostFW, where $N_t := |\{s \in \llbracket 0, t-1 \rrbracket \mid \gamma_s < 1, K_s > 1\}|$ is *nonnegligeable*, i.e., $N_t \geq \omega t^p$ for some $\omega > 0$ and $p \in]0, 1]$. This is the rate observed in practice, where $N_t \approx t-1$ so $\omega \lesssim 1$ and $p = 1$ (Section 4).

Theorem 3.4 (Practical rate). *Let $f : \mathcal{H} \rightarrow \mathbb{R}$ be L -smooth, convex, and μ -gradient dominated, and set $\gamma_t \leftarrow \min\{\eta_t \|\nabla f(x_t)\| / (L\|g_t\|), 1\}$ or $\gamma_t \leftarrow \arg \min_{\gamma \in [0,1]} f(x_t + \gamma g_t)$. Assume that $|\{s \in \llbracket 0, t-1 \rrbracket \mid \gamma_s < 1, K_s > 1\}| \geq \omega t^p$ for all $t \in \llbracket 0, T-1 \rrbracket$, for some $\omega > 0$ and $p \in]0, 1]$. Then for all $t \in \llbracket 0, T \rrbracket$,*

$$f(x_t) - \min_c f \leq \frac{LD^2}{2} \exp\left(-\delta^2 \frac{\mu}{L} \omega t^p\right).$$

Remark 3.5. *Note that when $\gamma_t < 1$ and $K_t > 1$, we have (see proofs in Appendix D.3)*

$$\left(f(x_t) - \min_c f\right) - \left(f(x_{t+1}) - \min_c f\right) \geq \delta^2 \frac{\|\nabla f(x_t)\|^2}{2L}$$

so if $N_T := |\{t \in \llbracket 0, T-1 \rrbracket \mid \gamma_t < 1, K_t > 1\}|$, then

$$f(x_0) - \min_c f \geq \delta^2 \frac{\inf_c \|\nabla f\|^2}{2L} N_T.$$

Thus, if $\inf_c \|\nabla f\| > 0$ then

$$N_T \leq \frac{2L(f(x_0) - \min_c f)}{\delta^2 \inf_c \|\nabla f\|^2} \leq \left(\frac{LD}{\delta \inf_c \|\nabla f\|}\right)^2$$

since $f(x_0) - \min_c f \leq LD^2/2$ (see proofs in Appendix D.3). However, the assumption in Theorem 3.4 can still hold as convergence is usually achieved within T iterations where

$$T = \mathcal{O}\left(\left(\frac{1}{\omega} \left(\frac{LD}{\delta \inf_c \|\nabla f\|}\right)^2\right)^{1/p}\right)$$

for some $\omega > 0$ and $p \in]0, 1]$. In the experiments for example (Section 4), convergence is always achieved within $\mathcal{O}(10^3)$ iterations. Furthermore, early stopping to increase the generalization error of a model also prevents T from blowing up.

Lastly, we provide in Corollary 3.6 a bound on the number of FW oracle calls, i.e., the number of linear minimizations over \mathcal{C} , performed to achieve ε -convergence. In comparison, FW and AFW respectively require $\mathcal{O}(LD^2/\varepsilon)$ and $\mathcal{O}((L/S)(D/W)^2 \ln(1/\varepsilon))$ oracle calls, where f is assumed to be S -strongly convex and \mathcal{C} is assumed to be a polytope with pyramidal width W for AFW (Lacoste-Julien and Jaggi, 2015). It is clear from its design that BoostFW performs more oracle calls per iteration, however it uses them more efficiently and the progress obtained overcomes the cost. This is demonstrated in the experiments (Section 4).

Corollary 3.6. *In order to achieve ε -convergence, the number of linear minimizations performed over \mathcal{C} is*

$$\begin{cases} \mathcal{O}\left(\frac{LD^2 \min\{K, 1/\delta\}}{\varepsilon}\right) & \text{in the worst-case scenario} \\ \mathcal{O}\left(\min\left\{K, \frac{1}{\delta}\right\} \left(\frac{1}{\omega\delta^2} \frac{L}{\mu} \ln\left(\frac{1}{\varepsilon}\right)\right)^{1/p}\right) & \text{in the practical scenario.} \end{cases}$$

Note that the practical scenario assumes that we have set $K \geq 2$ in BoostFW ($K = 1$ reduces BoostFW to FW).

4 Computational experiments

We compared the Boosted Frank-Wolfe algorithm (BoostFW, Algorithm 2) to the Away-Step Frank-Wolfe algorithm (AFW) (Wolfe, 1970), the Decomposition-Invariant Pairwise Conditional Gradient algorithm (DICG) (Garber and Meshi, 2016), and the Blended Conditional Gradients algorithm (BCG) (Braun et al., 2019) in a series of computational experiments. We ran two strategies for AFW, one with the default line search (AFW-ls) and one using the smoothness of f (AFW-L):

$$\gamma_t \leftarrow \begin{cases} \min\left\{\frac{\langle \nabla f(x_t), x_t - v_t^{\text{FW}} \rangle}{L\|x_t - v_t^{\text{FW}}\|_2^2}, 1\right\} & \text{if FW step} \\ \min\left\{\frac{\langle \nabla f(x_t), v_t^{\text{away}} - x_t \rangle}{L\|v_t^{\text{away}} - x_t\|_2^2}, \gamma_{\max}\right\} & \text{if away step} \end{cases}$$

where γ_{\max} is defined in the algorithm (see Algorithm 5 in Appendix C). Contrary to common belief, both strategies yield the same linear convergence rate; see Lacoste-Julien and Jaggi (2015) for AFW-ls and Theorem C.3 in the Appendix for AFW-L (Pedregosa et al., 2020). For BoostFW, we also ran a line search strategy to demonstrate that the speed-up really comes from the boosting procedure and not from being line search-free. Results further show that the line search-free strategy $\gamma_t \leftarrow \min\{\eta_t \|\nabla f(x_t)\|/(L\|g_t\|), 1\} = \min\{\langle -\nabla f(x_t), g_t \rangle/(L\|g_t\|^2), 1\}$ is very performant in CPU time. The line search-free strategy of DICG was not competitive in the experiments.

DICG is not applicable to optimization problems over the ℓ_1 -ball

$$\begin{aligned} \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. } \|x\|_1 \leq \tau, \end{aligned} \tag{4}$$

however we can perform a change of variables $x_i = z_i - z_{n+i}$ and use the following reformulation over the simplex:

$$\begin{aligned} \min_{z \in \mathbb{R}^{2n}} f([z]_{1:n} - [z]_{n+1:2n}) \\ \text{s.t. } z \in \tau \Delta_{2n} \end{aligned} \tag{5}$$

where $[z]_{1:n}$ and $[z]_{n+1:2n}$ denote the truncation to \mathbb{R}^n of the first n entries and the last n entries of $z \in \mathbb{R}^{2n}$ respectively. Fact 4.1 formally states the equivalence between problems (4) and (5). A proof can be found in Appendix D.4.

Fact 4.1. *Consider \mathbb{R}^n and let $\tau > 0$. Then $\mathcal{B}_1(\tau) = \{[z]_{1:n} - [z]_{n+1:2n} \mid z \in \tau \Delta_{2n}\}$.*

We implemented all the algorithms in Python using the same code framework for fair comparisons. In the case of synthetic data, we generated them from Gaussian distributions. We ran the experiments on a laptop under Linux Ubuntu 18.04 with Intel Core i7 3.5GHz CPU and 8GB RAM. Code is available at <https://github.com/cyrillewcombettes/boostfw>. In each experiment, we estimated the smoothness constant L of the (convex) objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, i.e., the Lipschitz constant of the gradient function $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, by sampling a few pairs of points $(x, y) \in \mathcal{C} \times \mathcal{C}$ and computing an upper bound on $\|\nabla f(y) - \nabla f(x)\|_2 / \|y - x\|_2$. Unless specified otherwise, we set $\delta \leftarrow 10^{-3}$ and $K \leftarrow +\infty$ in BoostFW. The role of K is only to cap the number of pursuit rounds per iteration when the FW oracle is particularly expensive (see Section 4.3).

4.1 Sparse signal recovery

Let $x^* \in \mathbb{R}^n$ be a signal which we want to recover as a sparse representation from observations $y = Ax^* + w$, where $A \in \mathbb{R}^{m \times n}$ and $w \sim \mathcal{N}(0, \sigma^2 I_m)$ is the noise in the measurements. The natural formulation of the problem is

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|y - Ax\|_2^2 \\ \text{s.t.} \quad & \|x\|_0 \leq \|x^*\|_0 \end{aligned}$$

but the ℓ_0 -pseudo-norm $\|\cdot\|_0 : x \in \mathbb{R}^n \mapsto |\{i \in \llbracket 1, n \rrbracket \mid [x]_i \neq 0\}|$ is nonconvex and renders the problem intractable in many situations (Natarajan, 1995). To remedy this, the ℓ_1 -norm is often used as a convex surrogate and leads to the following lasso formulation (Tibshirani, 1996) of the problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|y - Ax\|_2^2 \\ \text{s.t.} \quad & \|x\|_1 \leq \|x^*\|_1. \end{aligned}$$

In order to compare to DICG, which is not applicable to this formulation, we ran all algorithms on the reformulation (5). We set $m = 200$, $n = 500$, $\sigma = 0.05$, and $\tau = \|x^*\|_1$. Since the objective function is quadratic, we can derive a closed-form solution to the line search and there is no need for AFW-L or BoostFW-L. The results are presented in Figure 4.

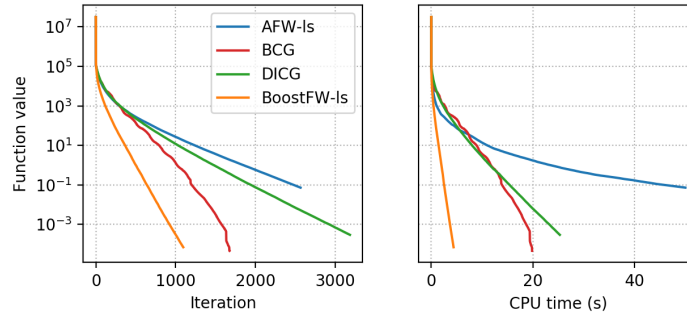


Figure 4: Sparse signal recovery.

4.2 Sparsity-constrained logistic regression

We consider the task of recognizing the handwritten digits 4 and 9 from the Gisette dataset (Guyon et al., 2005), available at <https://archive.ics.uci.edu/ml/datasets/Gisette>. The dataset includes a high number of distractor features with no predictive power. Hence, a sparsity-constrained logistic regression

model is suited for the task. The sparsity-inducing constraint is realized via the ℓ_1 -norm:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \frac{1}{m} \sum_{i=1}^m \ln(1 + \exp(-y_i a_i^\top x)) \\ \text{s.t.} \quad & \|x\|_1 \leq \tau \end{aligned}$$

where $a_1, \dots, a_m \in \mathbb{R}^n$ and $y \in \{-1, +1\}^m$. In order to compare to DICG, which is not applicable to this formulation, we ran all algorithms on the reformulation (5). We used $m = 2000$ samples and the number of features is $n = 5000$. We set $\tau = 10$, $L = 0.5$, and $\delta \leftarrow 10^{-4}$ in BoostFW. The results are presented in Figure 5. As expected, AFW-L and BoostFW-L converge faster in CPU time as they do not rely on line search, however they converge slower per iteration as each iteration provides less progress.

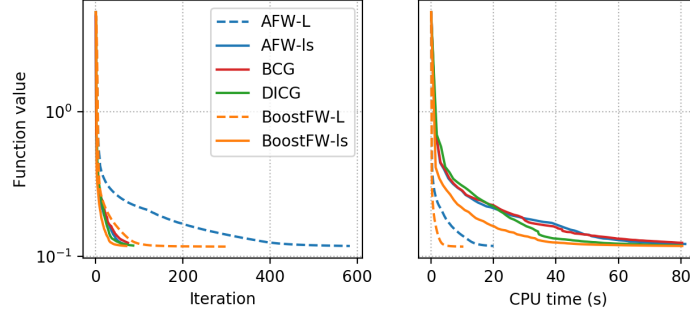


Figure 5: Sparse logistic regression on the Gisette dataset.

4.3 Traffic assignment

We consider the traffic assignment problem. The task is to assign vehicles on a traffic network in order to minimize congestion while satisfying travel demands. Let \mathcal{A} , \mathcal{R} , and \mathcal{S} be the sets of links, routes, and origin-destination pairs respectively. For every pair $(i, j) \in \mathcal{S}$, let $\mathcal{R}_{i,j}$ and $d_{i,j}$ be the set of routes and the travel demand from i to j . Let x_a and t_a be the flow and the travel time on link $a \in \mathcal{A}$, and let y_r be the flow on route $r \in \mathcal{R}$. The Beckmann formulation of the problem (Beckmann et al., 1956), derived from the Wardrop equilibrium conditions (Wardrop, 1952), is

$$\begin{aligned} \min_{x \in \mathbb{R}^{|\mathcal{A}|}} \quad & \sum_{a \in \mathcal{A}} \int_0^{x_a} t_a(\xi) d\xi \\ \text{s.t.} \quad & x_a = \sum_{r \in \mathcal{R}} \mathbb{1}_{\{a \in r\}} y_r \quad a \in \mathcal{A} \\ & \sum_{r \in \mathcal{R}_{i,j}} y_r = d_{i,j} \quad (i, j) \in \mathcal{S} \\ & y_r \geq 0 \quad r \in \mathcal{R}_{i,j}, (i, j) \in \mathcal{S}. \end{aligned} \tag{6}$$

A commonly used expression for the travel time t_a as a function of the flow x_a , developed by the Bureau of Public Records, is $t_a : x_a \in \mathbb{R}_+ \mapsto \tau_a(1 + 0.15(x_a/c_a)^4)$ where τ_a and c_a are the free-flow travel time and the capacity of the link. A linear minimization over the feasible region in (6) amounts to computing the shortest routes between all origin-destination pairs. Thus, the FW oracle is particularly expensive here so we capped the maximum number of rounds in BoostFW to $K \leftarrow 5$; see Figure 12 in Appendix A.2. We implemented the oracle using the function `all_pairs_dijkstra_path` from the Python package `networkx` (Hagberg et al., 2008). We created a directed acyclic graph with 500 nodes split into 20 layers of 25 nodes each, and randomly dropped links with probability 0.5 so $|\mathcal{A}| \approx 6000$ and $|\mathcal{S}| \approx 113000$. We set $d_{i,j} \sim \mathcal{U}([0, 1])$ for every $(i, j) \in \mathcal{S}$. DICG is not applicable here and AFW-L and BoostFW-L were not competitive. The results are

presented in Figure 6.

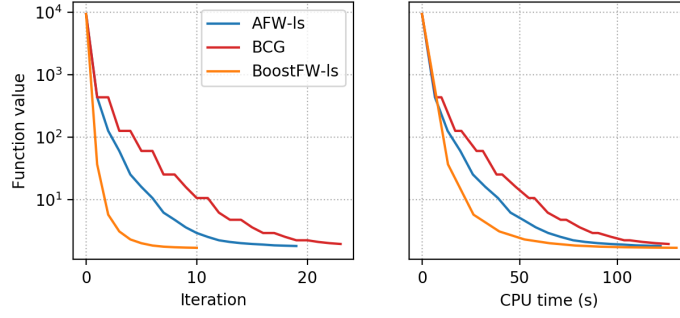


Figure 6: Traffic assignment.

4.4 Collaborative filtering

We consider the task of collaborative filtering on the MovieLens 100k dataset (Harper and Konstan, 2015), available at <https://grouplens.org/datasets/movielens/100k/>. The low-rank assumption on the solution and the approach of Mehta et al. (2007) lead to the following problem formulation:

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & \frac{1}{|\mathcal{I}|} \sum_{(i,j) \in \mathcal{I}} h_{\rho}(Y_{i,j} - X_{i,j}) \\ \text{s.t.} \quad & \|X\|_{\text{nuc}} \leq \tau \end{aligned}$$

where h_{ρ} is the Huber loss with parameter $\rho > 0$ (Huber, 1964):

$$h_{\rho} : t \in \mathbb{R} \mapsto \begin{cases} t^2/2 & \text{if } |t| \leq \rho \\ \rho(|t| - \rho/2) & \text{if } |t| > \rho, \end{cases}$$

$Y \in \mathbb{R}^{m \times n}$ is the given matrix to complete, $\mathcal{I} \subseteq \llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket$ is the set of indices of observed entries in Y , and $\|\cdot\|_{\text{nuc}} : X \in \mathbb{R}^{m \times n} \mapsto \text{tr}(\sqrt{X^{\top} X}) = \sum_{i=1}^{\min\{m,n\}} \sigma_i(X)$ is the nuclear norm and equals the sum of the singular vectors. It serves as a convex surrogate for the rank constraint (Fazel et al., 2001). Since

$$\{X \in \mathbb{R}^{m \times n} \mid \|X\|_{\text{nuc}} = 1\} = \text{conv}(\{uv^{\top} \mid u \in \mathbb{R}^m, v \in \mathbb{R}^n, \|u\|_2 = \|v\|_2 = 1\}),$$

a linear minimization over the nuclear norm-ball of radius τ amounts to computing the top left and right singular vectors u and v of $-\nabla f(X_t)$ and to return τuv^{\top} . To this end, we used the function `svds` from the Python package `scipy.sparse.linalg` (Virtanen et al., 2020). We have $m = 943$, $n = 1682$, and $|\mathcal{I}| = 10^5$, and we set $\rho = 1$, $\tau = 5000$, and $L = 5 \cdot 10^{-6}$. DICG is not applicable here. The results are presented in Figure 7.

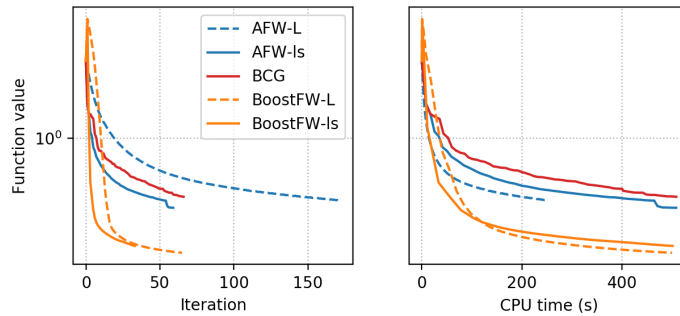


Figure 7: Collaborative filtering on the MovieLens 100k dataset.

The time limit here was set to 500 seconds but for AFW-L we reduced it to 250 seconds, else it raises a memory error on our machine shortly after. This is because AFW requires storing the decomposition of the iterate onto \mathcal{V} . Note that BoostFW-ls converges faster in CPU time than AFW-L, although it relies on line search, and that BoostFW-L converges faster per iteration than the other methods although it does not rely on line search.

4.5 Video co-localization

We consider the task of video co-localization on the aeroplane class of the YouTube-Objects dataset (Prest et al., 2012), using the problem formulation of Joulin et al. (2014). The goal is to localize (with bounding boxes) the aeroplane object across the video frames. It consists in minimizing $f : x \in \mathbb{R}^{660} \mapsto x^\top Ax/2 + b^\top x$ over a flow polytope, where $A \in \mathbb{R}^{660 \times 660}$, $b \in \mathbb{R}^{660}$, and the polytope each encode a part of the temporal consistency in the video frames. We obtained the data from <https://github.com/Simon-Lacoste-Julien/linearFW>. A linear minimization over the flow polytope amounts to computing a shortest path in the corresponding directed acyclic graph. We implemented the boosting procedure for DICG, which we labeled BoostDICG; see details in Appendix B. Since the objective function is quadratic, we can derive a closed-form solution to the line search and there is no need for AFW-L or BoostFW-L. We set $\delta \leftarrow 10^{-7}$ in BoostFW and $\delta \leftarrow 10^{-15}$ in BoostDICG. The results are presented in Figure 8.

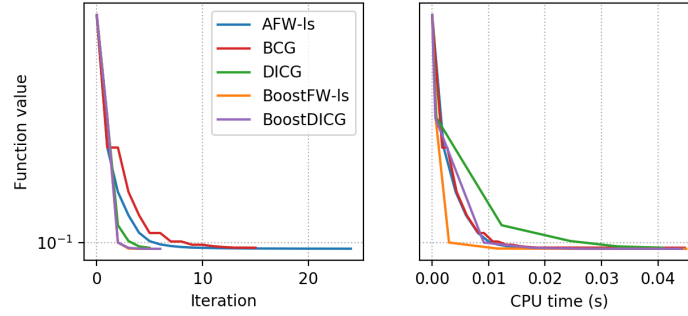


Figure 8: Video co-localization on the YouTube-Objects dataset.

All algorithms provide a similar level of performance in function value. In Garber and Meshi (2016), the algorithms are compared with respect to the duality gap $\max_{v \in \mathcal{V}} \langle \nabla f(x_t), x_t - v \rangle$ (Jaggi, 2013) on the same experiment. For completeness, we report a similar study in Figure 9. The boosting procedure applied to DICG produces very promising empirical results.

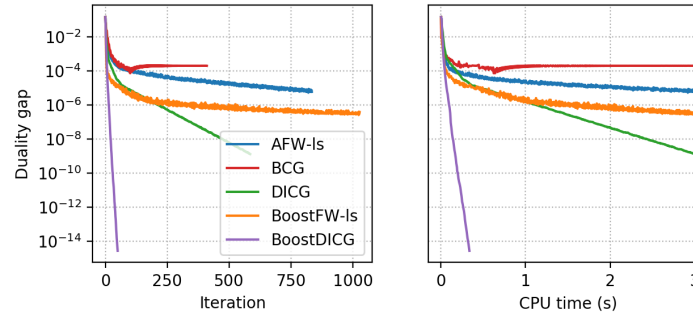


Figure 9: Video co-localization on the YouTube-Objects dataset.

Appendix A.3 presents comparisons in duality gap for the other experiments. DICG converges faster than BoostFW in duality gap here (after closing it to 10^{-6} though), but it is not the case in the other experiments.

5 Final remarks

We have proposed a new and intuitive method to speed up the Frank-Wolfe algorithm by descending in directions better aligned with those of the negative gradients $-\nabla f(x_t)$, all the while remaining projection-free. Our method does not need to maintain the decomposition of the iterates and can naturally be used to boost the performance of any Frank-Wolfe-style algorithm. Although the linear minimization oracle may be called multiple times per iteration, the progress obtained greatly overcomes this cost and leads to strong gains in performance. We demonstrated in a variety of experiments the computational advantage of our method both per iteration and in CPU time over the state-of-the-art. Furthermore, it does not require line search to produce strong performance in practice, which is particularly useful on instances where these are excessively expensive.

Future work may replace the gradient pursuit procedure with a faster conic optimization algorithm to potentially reduce the number of oracle calls. It could also be interesting to investigate how to make each oracle call cheaper via, e.g., *lazification* (Braun et al., 2017) or subsampling (Kerdreux et al., 2018). Lastly, we expect significant gains in performance when applying our approach to chase the gradient estimators in (non)convex stochastic Frank-Wolfe algorithms as well (Hazan and Luo, 2016; Xie et al., 2020).

Acknowledgments

Research reported in this paper was partially supported by NSF CAREER Award CMMI-1452463.

References

- M. A. Bashiri and X. Zhang. Decomposition-invariant conditional gradient for general polytopes with line search. In *Advances in Neural Information Processing Systems 30*, pages 2690–2700. 2017.
- M. J. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.
- J. Bolte, T.-P. Nguyen, J. Peypouquet, and B. W. Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507, 2017.
- G. Braun, S. Pokutta, and D. Zink. Lazifying conditional gradient algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, pages 566–575, 2017.
- G. Braun, S. Pokutta, D. Tu, and S. Wright. Blended conditional gradients: the unconditioning of conditional gradients. In *Proceedings of the 36th International Conference on Machine Learning*, pages 735–743, 2019.
- M. D. Canon and C. D. Cullum. A tight upper bound on the rate of convergence of Frank-Wolfe algorithm. *SIAM Journal on Control*, 6(4):509–516, 1968.
- J. Chen, D. Zhou, J. Yi, and Q. Gu. A Frank-Wolfe framework for efficient and effective adversarial attacks. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020. To appear.
- M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, pages 4734–4739, 2001.
- M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.
- D. Garber and E. Hazan. Faster rates for the Frank-Wolfe method over strongly-convex sets. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 541–549, 2015.
- D. Garber and E. Hazan. A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization. *SIAM Journal on Optimization*, 26(3):1493–1528, 2016.
- D. Garber and O. Meshi. Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. In *Advances in Neural Information Processing Systems 29*, pages 1001–1009. 2016.

- J. Guélat and P. Marcotte. Some comments on Wolfe’s ‘away step’. *Mathematical Programming*, 35(1):110–119, 1986.
- I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the NIPS 2003 feature selection challenge. In *Advances in Neural Information Processing Systems 17*, pages 545–552. 2005.
- A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, 2008.
- F. M. Harper and J. A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19:1–19:19, 2015.
- E. Hazan and S. Kale. Projection-free online learning. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.
- E. Hazan and H. Luo. Variance-reduced and projection-free stochastic optimization. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1263–1271, 2016.
- P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.
- M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435, 2013.
- 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, 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, 2016.
- T. Kerdreux, F. Pedregosa, and A. d’Aspremont. Frank-Wolfe with subsampling oracle. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2591–2600, 2018.
- K. Kurdyka. On gradients of functions definable in o-minimal structures. *Annales de l’Institut Fourier*, 48(3):769–783, 1998.
- S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *Advances in Neural Information Processing Systems 28*, pages 496–504. 2015.
- S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *Proceedings of the 30th International Conference on Machine Learning*, pages 53–61, 2013.
- G. Lan. The complexity of large-scale convex programming under a linear optimization oracle. Technical report, Department of Industrial and Systems Engineering, University of Florida, 2013.
- L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research*, 9(5):309–318, 1975.
- E. S. Levitin and B. T. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6(5):1–50, 1966.
- F. Locatello, M. Tschannen, G. Rätsch, and M. Jaggi. Greedy algorithms for cone constrained optimization with convergence guarantees. In *Advances in Neural Information Processing Systems 30*, pages 773–784. 2017.
- S. Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. In *Les Équations aux Dérivées Partielles*, 117, pages 87–89. Colloques Internationaux du CNRS, 1963.
- S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.
- B. Mehta, T. Hofmann, and W. Nejdl. Robust collaborative filtering. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, pages 49–56, 2007.
- B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

- F. Pedregosa, G. Négia, A. Askari, and M. Jaggi. Linearly convergent Frank-Wolfe with backtracking line-search. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pages 1–10, 2020.
- W. Ping, Q. Liu, and A. T. Ihler. Learning infinite RBMs with Frank-Wolfe. In *Advances in Neural Information Processing Systems 29*, pages 3063–3071. 2016.
- B. T. Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.
- A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3282–3289, 2012.
- S. Shalev-Shwartz, A. Gonen, and O. Shamir. Large-scale convex minimization with a low-rank constraint. In *Proceedings of the 28th International Conference on Machine Learning*, pages 329–336, 2011.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
- P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. Scipy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 2020.
- J. G. Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers*, volume 1, pages 325–378, 1952.
- P. Wolfe. Convergence theory in nonlinear programming. In *Integer and Nonlinear Programming*, pages 1–36. North-Holland, 1970.
- J. Xie, Z. Shen, C. Zhang, H. Qian, and B. Wang. Efficient projection-free online methods with stochastic recursive gradient. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020. To appear.

A Complementary plots

A.1 Lower bound on the number of oracle calls

Recall that for any $x \in \mathbb{R}^n$, $\|x\|_0 := |\{i \in \llbracket 1, n \rrbracket \mid [x]_i \neq 0\}|$ denotes the number of nonzero entries in x . Consider the problem of minimizing $f : x \in \mathbb{R}^n \mapsto \|x\|_2^2$ over the standard simplex Δ_n :

$$\min_{x \in \Delta_n} \|x\|_2^2.$$

Since Δ_n is the convex hull of the standard basis, Lemma A.1 establishes a lower bound on the function value of any Frank-Wolfe-style algorithm with respect to the number of oracle calls.

Lemma A.1 (Jaggi, 2013, Lemma 3). *Let $f : x \in \mathbb{R}^n \mapsto \|x\|_2^2$. Then for all $k \in \llbracket 1, n \rrbracket$,*

$$\min_{\substack{x \in \Delta_n \\ \|x\|_0 = k}} f(x) = \frac{1}{k}.$$

Indeed, suppose that $x_0 \in \{e_1, \dots, e_n\}$ is a standard vector and consider FW (Algorithm 1). FW makes exactly one call to the oracle in each iteration and adds the new vertex to the convex decomposition of the iterate, hence $\|x_t\|_0 \leq t + 1$ for all $t \in \llbracket 1, n - 1 \rrbracket$. Therefore, Lemma A.1 shows that the iterates of FW satisfy the lower bound

$$f(x_t) \geq \frac{1}{t + 1}.$$

This derivation can be extended to any Frank-Wolfe-style algorithm by comparing the function value at iteration t vs. the number of oracle calls up to iteration t . In Figure 10, we demonstrate that although BoostFW may call the oracle multiple times per iteration, it is still compatible with the lower bound. We set $n = 1000$ and since the objective is quadratic, we used an exact line search step-size strategy in FW-ls and BoostFW-ls. Note that the optimal value of the problem is $1/n = 10^{-3}$.

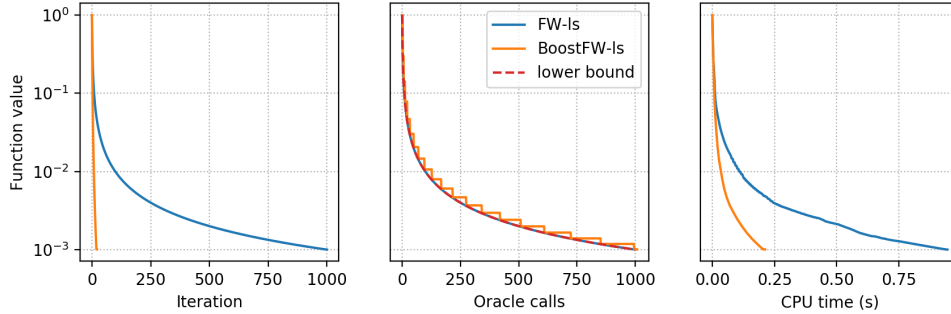


Figure 10: Lower bound on the number of oracle calls.

A.2 Illustration of the improvements in alignment during the gradient pursuit procedure

We define the relative improvement in alignment between rounds $k - 1$ and $k \in \llbracket 2, K_t \rrbracket$ in the gradient pursuit procedure at iteration $t \in \llbracket 0, T - 1 \rrbracket$ of BoostFW (Algorithm 2) as

$$\theta_{t,k} := \frac{\text{align}(-\nabla f(x_t), d_k) - \text{align}(-\nabla f(x_t), d_{k-1})}{\text{align}(-\nabla f(x_t), d_{k-1})}.$$

For a fixed round k , we plot in Figure 11 the mean of $\theta_{t,k}$ across all iterations t that performed a k -th round, i.e.,

$$\theta_k := \frac{1}{|\{t \in \llbracket 0, T - 1 \rrbracket \mid k \leq K_t\}|} \sum_{t=0}^{T-1} \theta_{t,k} \mathbb{1}_{\{k \leq K_t\}},$$

in the sparse signal recovery experiment (Section 4.1). The error bars represent ± 1 standard deviation. We see that on average the second round produces an improvement in alignment of $\sim 32\%$, the third round produces an improvement of $\sim 16\%$, etc. In particular, the plot could suggest that 7 rounds in each iteration are enough.

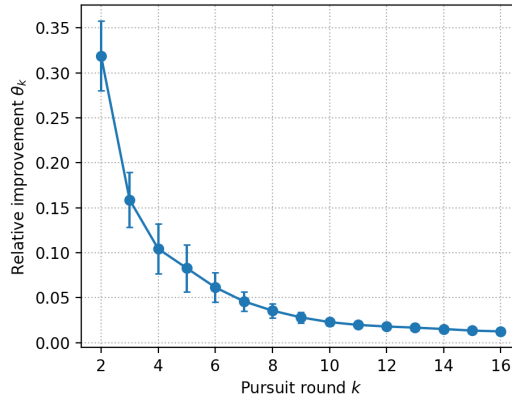


Figure 11: Relative improvements in alignment during the gradient pursuit procedure of BoostFW.

In the traffic assignment experiment (Section 4.3), the FW oracle is particularly expensive so we decided to cap the maximum number of rounds K . We plot in Figure 12 the relative improvements in alignment, and we chose to set $K \leftarrow 5$.

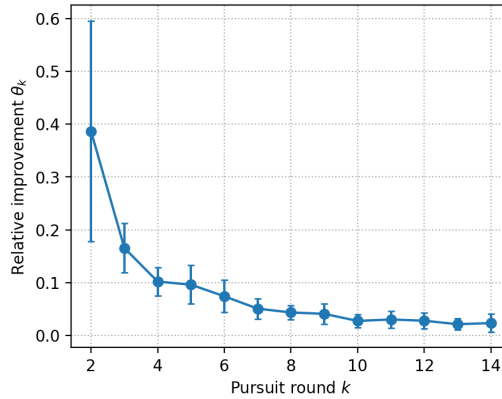


Figure 12: Traffic assignment (Section 4.3).

A.3 Computational experiments

Here we provide additional plots for each experiment of Section 4: comparisons in number of oracle calls and in duality gap. The duality gap is $\max_{v \in \mathcal{V}} \langle \nabla f(x_t), x_t - v \rangle$ (Jaggi, 2013) and we did not account for the CPU time taken to plot it. In number of oracle calls, the plots have a stair-like behavior as multiple calls can be made within an iteration. We see that BoostFW performs more oracle calls than the other methods in general however it converges faster both per iteration and in CPU time. Note that in the traffic assignment experiment (Figure 15), BoostFW also converges faster per oracle call. In the sparse logistic regression experiment (Figure 14), the line search-free strategies converge faster in CPU time and the line search strategies converge faster per iteration, but in the collaborative filtering experiment (Figure 16), BoostFW-ls and BoostFW-L respectively converge faster than expected.

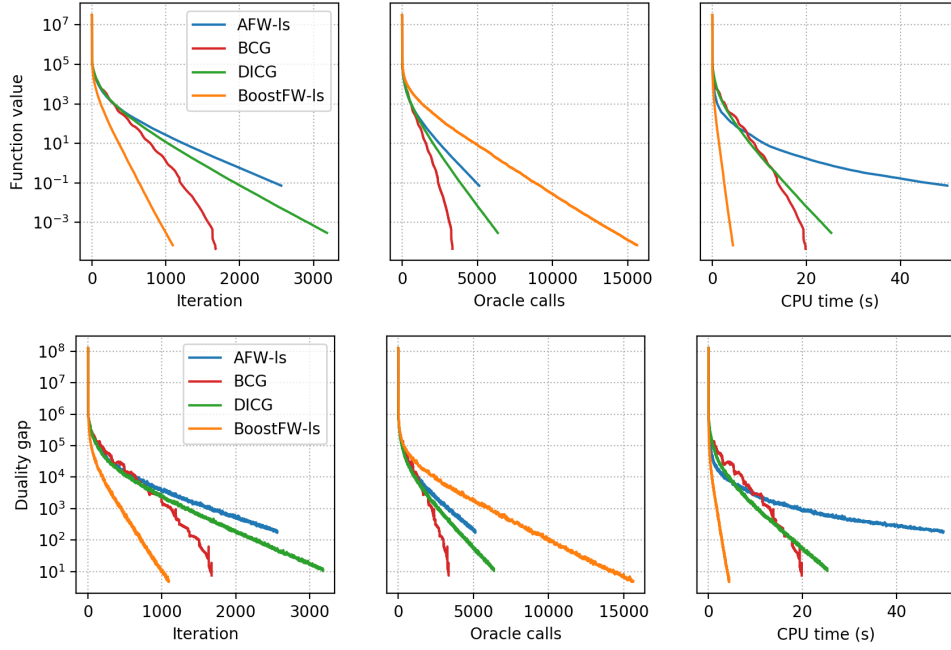


Figure 13: Sparse signal recovery (Section 4.1).

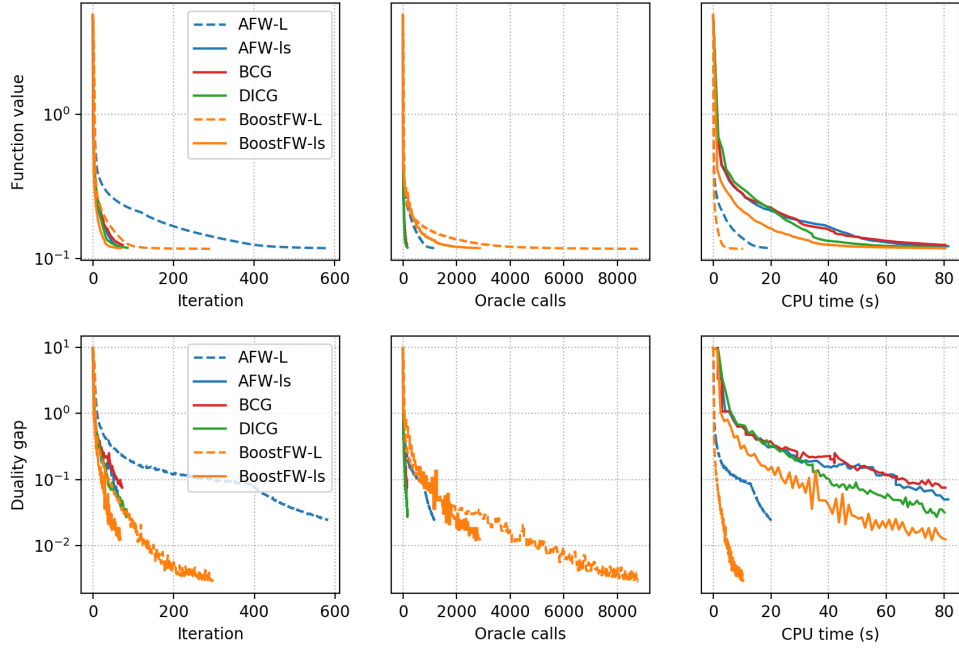


Figure 14: Sparse logistic regression on the Gisette dataset (Section 4.2).

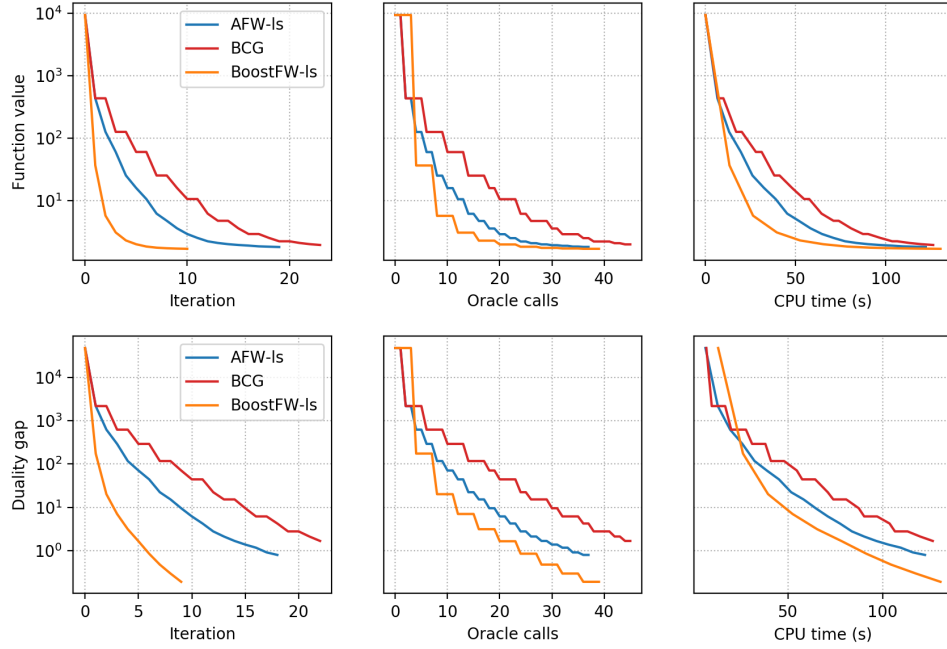


Figure 15: Traffic assignment (Section 4.3).

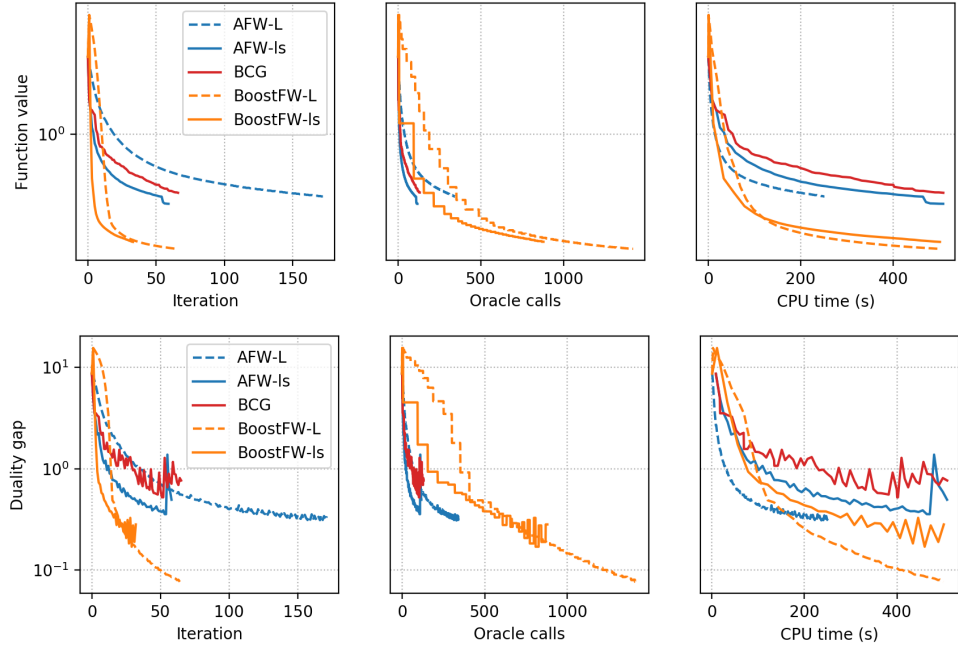


Figure 16: Collaborative filtering on the MovieLens 100k dataset (Section 4.4).

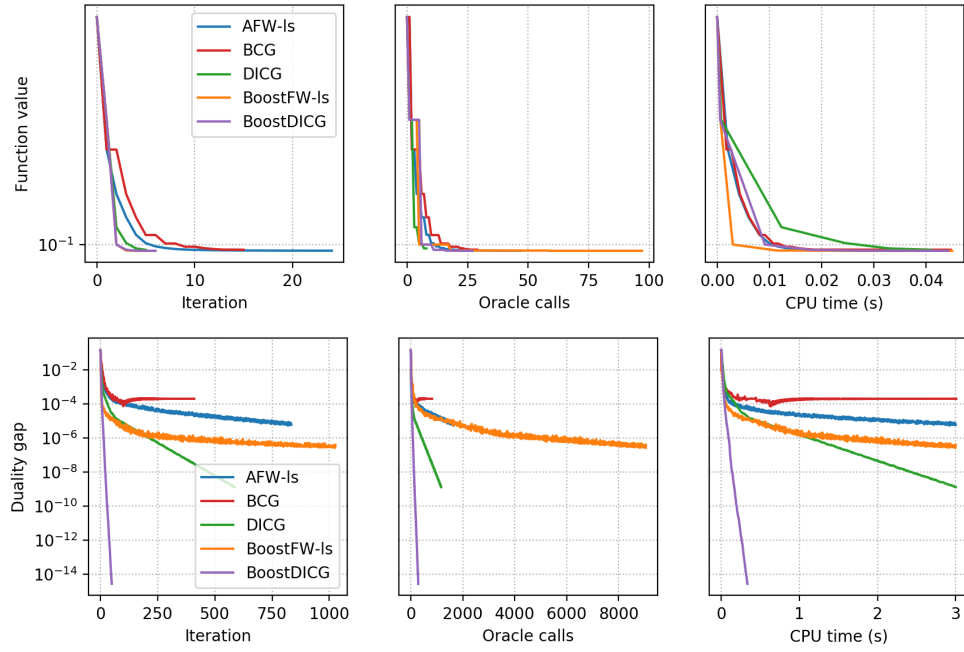


Figure 17: Video co-localization on the YouTube-Objects dataset (Section 4.5).

B Boosting DICG

We present an application of the boosting procedure to another Frank-Wolfe-style algorithm. Although the Away-Step and Pairwise Frank-Wolfe algorithms (Lacoste-Julien and Jaggi, 2015) are more similar in essence to the vanilla Frank-Wolfe algorithm, we chose to apply our approach to the Decomposition-Invariant Pairwise Conditional Gradient (DICG) (Garber and Meshi, 2016) because it does not need to maintain the decomposition of the iterates, which is a very favorable property in practice.

We recall DICG in Algorithm 3 and present our proposition for BoostDICG in Algorithm 4. Notice that since DICG moves in the pairwise direction $v_t^{\text{FW}} - v_t^{\text{away}}$, in BoostDICG we chase the direction of $-\nabla f(x_t)$ from v_t^{away} (and not from x_t). Proposition B.1 shows that the iterates of BoostDICG are feasible. Similarly to DICG, BoostDICG is applicable only to polytopes of the form $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ with set of vertices $\mathcal{V} \subseteq \{0, 1\}^n$, and it does not need to maintain the decomposition of the iterates. See also Bashiri and Zhang (2017) for a follow-up work extending DICG to arbitrary polytopes.

Algorithm 3 Decomposition-Invariant Pairwise Conditional Gradient (DICG)

Input: Start point $x_0 \in \mathcal{P}$.

Output: Point $x_T \in \mathcal{P}$.

- 1: $x_1 \leftarrow \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_0), v \rangle$
 - 2: **for** $t = 1$ **to** $T - 1$ **do**
 - 3: $v_t^{\text{FW}} \leftarrow \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_t), v \rangle$ ▷ FW oracle
 - 4: $\left[\tilde{\nabla} f(x_t) \right]_i \leftarrow \begin{cases} [\nabla f(x_t)]_i & \text{if } [x_t]_i > 0 \\ -\infty & \text{if } [x_t]_i = 0 \end{cases} \quad \text{for } i \in \llbracket 1, n \rrbracket$
 - 5: $v_t^{\text{away}} \leftarrow \arg \max_{v \in \mathcal{V}} \langle \tilde{\nabla} f(x_t), v \rangle$ ▷ FW oracle
 - 6: $\bar{\gamma}_t \leftarrow \max\{\gamma \in [0, 1] \mid x_t + \gamma(v_t^{\text{FW}} - v_t^{\text{away}}) \geq 0\}$
 - 7: $\gamma_t \leftarrow \arg \min_{\gamma \in [0, \bar{\gamma}_t]} f(x_t + \gamma(v_t^{\text{FW}} - v_t^{\text{away}}))$
 - 8: $x_{t+1} \leftarrow x_t + \gamma_t(v_t^{\text{FW}} - v_t^{\text{away}})$
 - 9: **end for**
-

Proposition B.1. *The iterates of BoostDICG (Algorithm 4) are feasible.*

Proof. We proceed by induction. By definition, $x_0 \in \arg \min_{v \in \mathcal{V}} \langle \nabla f(y), v \rangle \subseteq \mathcal{P}$. Let $t \in \llbracket 0, T-1 \rrbracket$ and suppose that $x_t \in \mathcal{P}$. Then $x_{t+1} \leftarrow x_t + \gamma_t g_t$ where $\gamma_t \in [0, \bar{\gamma}_t]$ and $\bar{\gamma}_t \leftarrow \max\{\gamma \in [0, 1] \mid x_t + \gamma g_t \geq 0\}$. Similarly to the proof of Proposition 3.1(iv), we can show that $g_t + v_t^{\text{away}} \in \mathcal{P}$. Thus, since $x_{t+1} = x_t + \gamma_t((g_t + v_t^{\text{away}}) - v_t^{\text{away}})$ and $Ax_t = b$, $A(g_t + v_t^{\text{away}}) = b$, and $Av_t^{\text{away}} = b$ by feasibility, we have $Ax_{t+1} = b$. Lastly, $x_t + \bar{\gamma}_t g_t \geq 0$ so, since $x_t \geq 0$ and $\gamma_t \in [0, \bar{\gamma}_t]$, we have $x_t + \gamma_t g_t \geq 0$, i.e., $x_{t+1} \geq 0$. Therefore, $x_{t+1} \in \mathcal{P}$. \square

Algorithm 4 Boosted Decomposition-Invariant Pairwise Conditional Gradient (BoostDICG)

Input: Input point $y \in \mathcal{C}$, maximum number of rounds $K \in \mathbb{N} \setminus \{0\}$, alignment improvement tolerance $\delta \in]0, 1[$, step-size strategy $\gamma_t \in [0, 1]$.

Output: Point $x_T \in \mathcal{P}$.

```

1:  $x_0 \leftarrow \arg \min_{v \in \mathcal{V}} \langle \nabla f(y), v \rangle$ 
2: for  $t = 0$  to  $T - 1$  do
3:    $\left[ \tilde{\nabla} f(x_t) \right]_i \leftarrow \begin{cases} [\nabla f(x_t)]_i & \text{if } [x_t]_i > 0 \\ -\infty & \text{if } [x_t]_i = 0 \end{cases} \quad \text{for } i \in \llbracket 1, n \rrbracket$ 
4:    $v_t^{\text{away}} \leftarrow \arg \max_{v \in \mathcal{V}} \langle \tilde{\nabla} f(x_t), v \rangle$  ▷ FW oracle
5:    $d_0 \leftarrow 0$ 
6:    $\Lambda_t \leftarrow 0$ 
7:   for  $k = 0$  to  $K - 1$  do ▷  $k$ -th residual
8:      $r_k \leftarrow -\nabla f(x_t) - d_k$  ▷ FW oracle
9:      $v_k \leftarrow \arg \max_{v \in \mathcal{V}} \langle r_k, v \rangle$ 
10:     $u_k \leftarrow \arg \max_{u \in \{v_k - v_t^{\text{away}}, -d_k / \|d_k\|\}} \langle r_k, u \rangle$ 
11:     $\lambda_k \leftarrow \frac{\langle r_k, u_k \rangle}{\|u_k\|^2}$ 
12:     $d'_{k+1} \leftarrow d_k + \lambda_k u_k$ 
13:    if  $\text{align}(-\nabla f(x_t), d'_{k+1}) - \text{align}(-\nabla f(x_t), d_k) \geq \delta$  then
14:       $d_{k+1} \leftarrow d'_{k+1}$ 
15:       $\Lambda_t \leftarrow \begin{cases} \Lambda_t + \lambda_k & \text{if } u_k = v_k - v_t^{\text{away}} \\ \Lambda_t(1 - \lambda_k / \|d_k\|) & \text{if } u_k = -d_k / \|d_k\| \end{cases}$ 
16:    else
17:      break ▷ exit  $k$ -loop
18:    end if
19:  end for
20:   $K_t \leftarrow k$ 
21:   $g_t \leftarrow d_{K_t} / \Lambda_t$  ▷ normalization
22:   $\bar{\gamma}_t \leftarrow \max\{\gamma \in [0, 1] \mid x_t + \gamma g_t \geq 0\}$ 
23:   $\gamma_t \leftarrow \arg \min_{\gamma \in [0, \bar{\gamma}_t]} f(x_t + \gamma g_t)$ 
24:   $x_{t+1} \leftarrow x_t + \gamma_t g_t$ 
25: end for

```

C A result on the Away-Step Frank-Wolfe algorithm

We first recall the Away-Step Frank-Wolfe algorithm (AFW) (Wolfe, 1970) in Algorithm 5 and its convergence rate over polytopes with line search in Theorem C.2 (Lacoste-Julien and Jaggi, 2015). This analysis is based on the pyramidal width of the polytope and the geometric strong convexity of the objective function (Lemma C.1). We refer the reader to Lacoste-Julien and Jaggi (2015, Section 3) for the definition of the pyramidal width.

Let $\mathcal{V} \subset \mathcal{P}$ denote the set of vertices of the polytope \mathcal{P} . In Algorithm 5, $\lambda_t \in \mathbb{R}^{|\mathcal{V}|}$ denotes the distribution of coefficients of the convex decomposition of x_t over \mathcal{V} and $\lambda_t(v)$ is the coefficient of vertex $v \in \mathcal{V}$ (as determined by the algorithm). Note that when an away step is taken and $\gamma_t = \gamma_{\max}$, then $\lambda_{t+1}(v_t^{\text{away}}) = 0$ where $v_t^{\text{away}} \in \mathcal{S}_t$. Thus, these steps always decrease the size of the active set: $|\mathcal{S}_{t+1}| < |\mathcal{S}_t|$. They are often referred to as *drop steps*.

Algorithm 5 Away-Step Frank-Wolfe (AFW)

Input: Start vertex $x_0 \in \mathcal{V}$, step-size strategy $\gamma_t \in [0, 1]$.

Output: Point $x_T \in \mathcal{P}$.

```

1:  $\mathcal{S}_0 \leftarrow \{x_0\}$  ▷ active set
2:  $\lambda_0 \leftarrow \{\mathbb{1}_{\{v=x_0\}} \text{ for } v \in \mathcal{V}\}$  ▷ distribution of coefficients
3: for  $t = 0$  to  $T - 1$  do
4:    $v_t^{\text{FW}} \leftarrow \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_t), v \rangle$  ▷ FW oracle
5:    $v_t^{\text{away}} \leftarrow \arg \max_{v \in \mathcal{S}_t} \langle \nabla f(x_t), v \rangle$ 
6:   if  $\langle \nabla f(x_t), x_t - v_t^{\text{FW}} \rangle \geq \langle \nabla f(x_t), v_t^{\text{away}} - x_t \rangle$  then
7:      $x_{t+1} \leftarrow x_t + \gamma_t(v_t^{\text{FW}} - x_t)$  ▷ FW step
8:      $\mathcal{S}_{t+1} \leftarrow \mathcal{S}_t \cup \{v_t^{\text{FW}}\}$ 
9:      $\lambda_{t+1}(v) \leftarrow (1 - \gamma_t)\lambda_t(v) + \gamma_t \mathbb{1}_{\{v=v_t^{\text{FW}}\}}$  for  $v \in \mathcal{S}_{t+1}$ 
10:  else
11:     $\gamma_{\max} \leftarrow \lambda_t(v_t^{\text{away}})/(1 - \lambda_t(v_t^{\text{away}}))$ 
12:     $x_{t+1} \leftarrow x_t + \gamma_t(x_t - v_t^{\text{away}})$  ▷ away step
13:     $\mathcal{S}_{t+1} \leftarrow \mathcal{S}_t$ 
14:     $\lambda_{t+1}(v) \leftarrow (1 + \gamma_t)\lambda_t(v) - \gamma_t \mathbb{1}_{\{v=v_t^{\text{away}}\}}$  for  $v \in \mathcal{S}_{t+1}$ 
15:  end if
16:   $\mathcal{S}_{t+1} \leftarrow \{v \in \mathcal{S}_{t+1} \mid \lambda_{t+1}(v) > 0\}$ 
17: end for
```

Lemma C.1 (Lacoste-Julien and Jaggi, 2015, Equations (28) and (23)). Let $\mathcal{P} \subset \mathbb{R}^n$ be a polytope with pyramidal width $W > 0$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a S -strongly convex function. Then the Away-Step Frank-Wolfe algorithm (AFW, Algorithm 5) ensures for all $t \in \llbracket 0, T - 1 \rrbracket$,

$$f(x_t) - \min_{\mathcal{P}} f \leq \frac{\langle \nabla f(x_t), v_t^{\text{away}} - v_t^{\text{FW}} \rangle^2}{2SW^2}.$$

Theorem C.2 (AFW with line search, Lacoste-Julien and Jaggi (2015, Theorem 1)). Let $\mathcal{P} \subset \mathbb{R}^n$ be a polytope and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a L -smooth and S -strongly convex function. Consider the Away-Step Frank-Wolfe algorithm (AFW, Algorithm 5) with the line search strategy

$$\gamma_t \leftarrow \begin{cases} \arg \min_{\gamma \in [0, 1]} f(x_t + \gamma(v_t^{\text{FW}} - x_t)) & \text{in the case of a FW step (Line 7)} \\ \arg \min_{\gamma \in [0, \gamma_{\max}]} f(x_t + \gamma(x_t - v_t^{\text{away}})) & \text{in the case of an away step (Line 12).} \end{cases}$$

Then for all $t \in \llbracket 0, T \rrbracket$,

$$f(x_t) - \min_{\mathcal{P}} f \leq \left(1 - \frac{S}{4L} \left(\frac{W}{D}\right)^2\right)^{t/2} \left(f(x_0) - \min_{\mathcal{P}} f\right)$$

where D and W are the diameter and the pyramidal width of \mathcal{P} respectively.

We now show in Theorem C.3 that AFW can also achieve the convergence rate of Theorem C.2 without line search, by using the short step strategy. We were later informed that this result was already derived by Pedregosa et al. (2020) in a more general setting.

Theorem C.3 (AFW with short steps). *Let $\mathcal{P} \subset \mathbb{R}^n$ be a polytope and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a L -smooth and S -strongly convex function. Consider the Away-Step Frank-Wolfe algorithm (AFW, Algorithm 5) with the step-size strategy*

$$\gamma_t \leftarrow \begin{cases} \min \left\{ \frac{\langle \nabla f(x_t), x_t - v_t^{\text{FW}} \rangle}{L \|x_t - v_t^{\text{FW}}\|_2^2}, 1 \right\} & \text{in the case of a FW step (Line 7)} \\ \min \left\{ \frac{\langle \nabla f(x_t), v_t^{\text{away}} - x_t \rangle}{L \|v_t^{\text{away}} - x_t\|_2^2}, \gamma_{\max} \right\} & \text{in the case of an away step (Line 12).} \end{cases} \quad (7)$$

Then for all $t \in \llbracket 0, T \rrbracket$,

$$f(x_t) - \min_{\mathcal{P}} f \leq \left(1 - \frac{S}{4L} \left(\frac{W}{D}\right)^2\right)^{t/2} \left(f(x_0) - \min_{\mathcal{P}} f\right)$$

where D and W are the diameter and the pyramidal width of \mathcal{P} respectively.

Proof. Let $t \in \llbracket 0, T - 1 \rrbracket$ and denote $\varepsilon_t := f(x_t) - \min_{\mathcal{P}} f$. By geometric strong convexity (Lemma C.1),

$$\varepsilon_t \leq \frac{\langle \nabla f(x_t), v_t^{\text{away}} - v_t^{\text{FW}} \rangle^2}{2SW^2}. \quad (8)$$

Furthermore, since

$$\langle \nabla f(x_t), v_t^{\text{away}} - v_t^{\text{FW}} \rangle = \langle \nabla f(x_t), v_t^{\text{away}} - x_t \rangle + \langle \nabla f(x_t), x_t - v_t^{\text{FW}} \rangle \geq 0$$

we have

$$\max\{\langle \nabla f(x_t), v_t^{\text{away}} - x_t \rangle, \langle \nabla f(x_t), x_t - v_t^{\text{FW}} \rangle\} \geq \frac{\langle \nabla f(x_t), v_t^{\text{away}} - v_t^{\text{FW}} \rangle}{2} \geq 0. \quad (9)$$

Note that AFW performs a step corresponding to $\max\{\langle \nabla f(x_t), v_t^{\text{away}} - x_t \rangle, \langle \nabla f(x_t), x_t - v_t^{\text{FW}} \rangle\}$ (Lines 6 and 10).

FW step. In the case where the algorithm performs a FW step, we have $\gamma_t \leftarrow \min\{\langle \nabla f(x_t), x_t - v_t^{\text{FW}} \rangle / (L \|x_t - v_t^{\text{FW}}\|_2^2), 1\}$, $x_{t+1} \leftarrow x_t + \gamma_t(v_t^{\text{FW}} - x_t)$, and by Line 6 and (9),

$$\langle \nabla f(x_t), x_t - v_t^{\text{FW}} \rangle \geq \frac{\langle \nabla f(x_t), v_t^{\text{away}} - v_t^{\text{FW}} \rangle}{2} \geq 0. \quad (10)$$

By smoothness of f ,

$$\varepsilon_{t+1} \leq \varepsilon_t + \gamma_t \langle \nabla f(x_t), v_t^{\text{FW}} - x_t \rangle + \frac{L}{2} \gamma_t^2 \|v_t^{\text{FW}} - x_t\|_2^2. \quad (11)$$

Consider the choice of step-size (7) and suppose $\gamma_t < 1$. Then, with (10) and (8),

$$\begin{aligned} \varepsilon_{t+1} &\leq \varepsilon_t - \frac{\langle \nabla f(x_t), x_t - v_t^{\text{FW}} \rangle^2}{2L \|x_t - v_t^{\text{FW}}\|_2^2} \\ &\leq \varepsilon_t - \frac{\langle \nabla f(x_t), v_t^{\text{away}} - v_t^{\text{FW}} \rangle^2}{8LD^2} \\ &\leq \left(1 - \frac{S}{4L} \left(\frac{W}{D}\right)^2\right) \varepsilon_t. \end{aligned}$$

If $\gamma_t = 1$ then $\langle \nabla f(x_t), x_t - v_t^{\text{FW}} \rangle \geq L \|x_t - v_t^{\text{FW}}\|_2^2$. By (11), the optimality of v_t^{FW} (Line 4), and the convexity of f ,

$$\begin{aligned} \varepsilon_{t+1} &\leq \varepsilon_t + \langle \nabla f(x_t), v_t^{\text{FW}} - x_t \rangle + \frac{L}{2} \|v_t^{\text{FW}} - x_t\|_2^2 \\ &\leq \varepsilon_t + \frac{\langle \nabla f(x_t), v_t^{\text{FW}} - x_t \rangle}{2} \\ &\leq \varepsilon_t + \frac{\langle \nabla f(x_t), x^* - x_t \rangle}{2} \\ &\leq \frac{\varepsilon_t}{2}. \end{aligned}$$

Therefore, the progress obtain by a FW step is

$$\begin{aligned} \varepsilon_{t+1} &\leq \left(1 - \min \left\{ \frac{1}{2}, \frac{S}{4L} \left(\frac{W}{D} \right)^2 \right\} \right) \varepsilon_t \\ &= \left(1 - \frac{S}{4L} \left(\frac{W}{D} \right)^2 \right) \varepsilon_t \end{aligned} \quad (12)$$

since $S \leq L$ and $W \leq D$.

Away step. In the case where the algorithm performs an away step, we have $\gamma_t \leftarrow \min\{\langle \nabla f(x_t), v_t^{\text{away}} - x_t \rangle / (L \|v_t^{\text{away}} - x_t\|_2^2), \gamma_{\max}\}$, $x_{t+1} \leftarrow x_t + \gamma_t(x_t - v_t^{\text{away}})$, and by Line 6 and (9),

$$\langle \nabla f(x_t), v_t^{\text{away}} - x_t \rangle \geq \frac{\langle \nabla f(x_t), v_t^{\text{away}} - v_t^{\text{FW}} \rangle}{2} \geq 0. \quad (13)$$

By smoothness of f ,

$$\varepsilon_{t+1} \leq \varepsilon_t + \gamma_t \langle \nabla f(x_t), x_t - v_t^{\text{away}} \rangle + \frac{L}{2} \gamma_t^2 \|x_t - v_t^{\text{away}}\|_2^2 \quad (14)$$

Consider the choice of step-size (7) and suppose $\gamma_t < \gamma_{\max}$. Then, with (13) and (8),

$$\begin{aligned} \varepsilon_{t+1} &\leq \varepsilon_t - \frac{\langle \nabla f(x_t), v_t^{\text{away}} - x_t \rangle^2}{2L \|v_t^{\text{away}} - x_t\|_2^2} \\ &\leq \varepsilon_t - \frac{\langle \nabla f(x_t), v_t^{\text{away}} - v_t^{\text{FW}} \rangle^2}{8LD^2} \\ &\leq \left(1 - \frac{S}{4L} \left(\frac{W}{D} \right)^2 \right) \varepsilon_t. \end{aligned} \quad (15)$$

If $\gamma_t = \gamma_{\max}$ then $\gamma_{\max} \leq \gamma_t^* := \langle \nabla f(x_t), v_t^{\text{away}} - x_t \rangle / (L \|v_t^{\text{away}} - x_t\|_2^2)$. Let $\varphi_t : \gamma \in [0, \gamma_t^*] \mapsto \gamma \langle \nabla f(x_t), x_t - v_t^{\text{away}} \rangle + L\gamma^2 \|x_t - v_t^{\text{away}}\|_2^2 / 2$. By (14),

$$\varepsilon_{t+1} \leq \varepsilon_t + \varphi_t(\gamma_{\max}).$$

The quadratic function φ_t attains its unique global minimum at $\gamma = \gamma_t^*$ and satisfies $\varphi_t(\gamma_t^*) = -\langle \nabla f(x_t), x_t - v_t^{\text{away}} \rangle^2 / (2L \|x_t - v_t^{\text{away}}\|_2^2) \leq 0$ and $\varphi_t(0) = 0$. Since $\gamma_{\max} \in [0, \gamma_t^*]$, we have $\varphi_t(\gamma_{\max}) \leq 0$ so

$$\varepsilon_{t+1} \leq \varepsilon_t \quad (16)$$

which shows that the progress is always nonnegative.

Wrapping up. Since the steps with progress (16) always decrease the number of vertices in the active set S_t , there are at most $\lfloor t/2 \rfloor$ after t iterations. By (12) and (15), we conclude that

$$\varepsilon_t \leq \left(1 - \frac{S}{4L} \left(\frac{W}{D} \right)^2 \right)^{\lfloor t/2 \rfloor} \varepsilon_0 \leq \left(1 - \frac{S}{4L} \left(\frac{W}{D} \right)^2 \right)^{t/2} \varepsilon_0.$$

□

D Proofs

D.1 Preliminaries

Fact D.1 (Fact 2.1). *Let $f : \mathcal{H} \rightarrow \mathbb{R}$ be S -strongly convex. Then f is S -gradient dominated.*

Proof. The function f is strongly convex hence it has a unique minimizer, which we denote by $x^* \in \mathcal{H}$. Let $x \in \mathcal{H}$. By strong convexity, for all $y \in \mathcal{H}$ we have

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{S}{2} \|y - x\|^2.$$

Now we minimize both sides with respect to $y \in \mathcal{H}$. The left-hand side is minimized for $y = x^*$ and the right-hand side is minimized for $y = x - \nabla f(x)/S$. Thus,

$$f(x^*) \geq f(x) - \frac{\|\nabla f(x)\|^2}{2S}$$

i.e.,

$$f(x) - \min_{\mathcal{H}} f \leq \frac{\|\nabla f(x)\|^2}{2S}.$$

□

D.2 Boosting via gradient pursuit

Proposition D.2 (Proposition 3.1). *Let $t \in \llbracket 0, T - 1 \rrbracket$ and suppose that $x_t \in \mathcal{C}$. Then:*

- (i) d_1 is defined and $K_t \geq 1$,
- (ii) $\lambda_0, \dots, \lambda_{K_t-1} \geq 0$,
- (iii) $d_k \in \text{cone}(\mathcal{V} - x_t)$ for all $k \in \llbracket 0, K_t \rrbracket$,
- (iv) $x_t + g_t \in \mathcal{C}$ and $x_{t+1} \in \mathcal{C}$,
- (v) $\text{align}(-\nabla f(x_t), g_t) \geq \text{align}(-\nabla f(x_t), v_t - x_t) + (K_t - 1)\delta$ where $v_t \in \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_t), v \rangle$ and $\text{align}(-\nabla f(x_t), v_t - x_t) \geq 0$.

Since $x_0 \in \mathcal{C}$, these properties are satisfied for all $t \in \llbracket 0, T - 1 \rrbracket$ by induction.

Proof. We analyze BoostFW (Algorithm 2).

- (i) We have $d_0 = 0$ so $r_0 = -\nabla f(x_t)$ and $\text{align}(r_0, d_0) = -1$ by definition (3). Furthermore, since $v_0 \in \arg \max_{v \in \mathcal{V}} \langle r_0, v \rangle$ and $x_t \in \mathcal{C}$, we have

$$\langle r_0, v_0 - x_t \rangle = \langle r_0, v_0 \rangle - \langle r_0, x_t \rangle \geq 0$$

so $u_0 = v_0 - x_t$ by Line 9 since $d_0 = 0$. Thus, $\lambda_0 \geq 0$ (Line 10) and $d'_1 = \lambda_0(v_0 - x_t)$ (Line 11) so

$$\begin{aligned} \text{align}(-\nabla f(x_t), d'_1) &= \frac{\langle -\nabla f(x_t), d'_1 \rangle}{\|\nabla f(x_t)\| \|d'_1\|} \\ &= \frac{\langle r_0, v_0 - x_t \rangle}{\|r_0\| \|v_0 - x_t\|} \\ &\geq 0 \\ &\geq -1 + \delta \\ &= \text{align}(-\nabla f(x_t), d_0) + \delta. \end{aligned}$$

Therefore, by Line 12 the gradient pursuit procedure continues.

(ii) Let $k \in \llbracket 0, K_t - 1 \rrbracket$. Since $v_k \in \arg \max_{v \in \mathcal{V}} \langle r_k, v \rangle$ (Line 8) and $x_t \in \text{conv}(\mathcal{V}) = \mathcal{C}$,

$$\langle r_k, v_k - x_t \rangle = \max_{v \in \mathcal{V}} \langle r_k, v \rangle - \langle r_k, x_t \rangle \geq 0.$$

Thus, by Lines 9-10 we have

$$\lambda_k = \frac{\langle r_k, u_k \rangle}{\|u_k\|^2} \geq 0.$$

Furthermore, note that $-r_k$ is the gradient of the objective function in subproblem (2) and $\langle r_k, u_k \rangle$ is a scaled upper bound on its primal gap (Locatello et al., 2017). Thus, if $\lambda_k = 0$ then the gradient pursuit procedure has already converged.

(iii) We show by induction that $d_k \in \text{cone}(\mathcal{V} - x_t)$ for all $k \in \llbracket 0, K_t \rrbracket$. We have $d_0 = 0 \in \text{cone}(\mathcal{V} - x_t)$ so the base case is satisfied. Suppose that $d_k \in \text{cone}(\mathcal{V} - x_t)$ for some $k \in \llbracket 0, K_t - 1 \rrbracket$. If $u_k = v_k - x_t$ then $u_k \in \mathcal{V} - x_t$ and since $\lambda_k \geq 0$ by (ii), we have $d_{k+1} = d_k + \lambda_k(v_k - x_t) \in \text{cone}(\mathcal{V} - x_t)$. Else, $u_k = -d_k/\|d_k\|$ so $d_{k+1} = (1 - \lambda_k/\|d_k\|)d_k$ and it remains to show that $1 - \lambda_k/\|d_k\| \geq 0$. We will show that $1 - \lambda_k/\|d_k\| \geq 1/2$. We have

$$\begin{aligned} 1 - \frac{\lambda_k}{\|d_k\|} &\geq \frac{1}{2} \Leftrightarrow \frac{1}{2} \geq \frac{\lambda_k}{\|d_k\|} = \frac{\langle r_k, -d_k/\|d_k\| \rangle}{\|d_k\|} \\ &\Leftrightarrow \frac{\|d_k\|^2}{2} \geq \langle r_k, -d_k \rangle. \end{aligned} \tag{17}$$

so it suffices to show that $\|d_k\|^2/2 \geq \langle r_k, -d_k \rangle$. Now, the procedure satisfies for all $k' \in \llbracket 0, K_t - 1 \rrbracket$,

$$\begin{aligned} \|r_{k'+1}\|^2 &= \|r_{k'} - \lambda_{k'} u_{k'}\|^2 \\ &= \|r_{k'}\|^2 - 2\lambda_{k'} \langle r_{k'}, u_{k'} \rangle + \lambda_{k'}^2 \|u_{k'}\|^2 \\ &= \|r_{k'}\|^2 - \frac{\langle r_{k'}, u_{k'} \rangle^2}{\|u_{k'}\|^2} \\ &\leq \|r_{k'}\|^2 \end{aligned}$$

where we used $\lambda_{k'} = \langle r_{k'}, u_{k'} \rangle / \|u_{k'}\|^2$. Thus $\|r_k\|^2 \leq \|r_0\|^2$, i.e., since $d_0 = 0$, $\|\nabla f(x_t) + d_k\|^2 \leq \|\nabla f(x_t)\|^2$ so

$$\|\nabla f(x_t)\|^2 \geq \|\nabla f(x_t) + d_k\|^2 = \|\nabla f(x_t)\|^2 + 2\langle \nabla f(x_t), d_k \rangle + \|d_k\|^2$$

hence

$$\langle \nabla f(x_t), d_k \rangle \leq -\frac{\|d_k\|^2}{2}.$$

Thus,

$$\begin{aligned} \langle r_k, -d_k \rangle &= \langle \nabla f(x_t) + d_k, d_k \rangle \\ &= \langle \nabla f(x_t), d_k \rangle + \|d_k\|^2 \\ &\leq \frac{\|d_k\|^2}{2}. \end{aligned}$$

Therefore, with (17) we can conclude that $d_{k+1} \in \text{cone}(\mathcal{V} - x_t)$.

(iv) By (iii), $d_{K_t} \in \text{cone}(\mathcal{V} - x_t)$ so since $g_t = d_{K_t}/\Lambda_t$, to show that $x_t + g_t \in \mathcal{C}$ it suffices to show that the sum of coefficients in the conical decomposition of d_{K_t} is equal to Λ_t , and then it follows that $g_t \in \text{conv}(\mathcal{V} - x_t) = \text{conv}(\mathcal{V}) - x_t = \mathcal{C} - x_t$. By Line 14, this is true and is verified by a simple induction on k : the base case is satisfied and if $u_k = v_k - x_t$ then $d_{k+1} = d_k + \lambda_k(v_k - x_t)$ and Line 14 shows

that $\Lambda_t \leftarrow \Lambda_t + \lambda_k$ is updated accordingly, else $u_k = -d_k/\|d_k\|$ so $d_{k+1} = (1 - \lambda_k/\|d_k\|)d_k$ and Line 14 shows that $\Lambda_t \leftarrow \Lambda_t(1 - \lambda_k/\|d_k\|)$ is again updated accordingly. Thus, $x_t + g_t \in \mathcal{C}$. Then,

$$\begin{aligned} x_{t+1} &= x_t + \gamma_t g_t \\ &= x_t + \gamma_t ((x_t + g_t) - x_t) \\ &= (1 - \gamma_t)x_t + \gamma_t \underbrace{(x_t + g_t)}_{\in \mathcal{C}}. \end{aligned}$$

Since $\gamma_t \in [0, 1]$, we conclude that $x_{t+1} \in \mathcal{C}$ by convex combination.

- (v) Since $d_0 = 0$, we have $r_0 = -\nabla f(x_t)$ so by Line 8, $v_0 \in \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_t), v \rangle$, and we have $d_1 = v_t - x_t$. Let $v_t := v_0$. Since $g_t = d_{K_t}/\Lambda_t$ where $K_t \geq 1$ by (i), by Line 12 we obtain

$$\begin{aligned} \text{align}(-\nabla f(x_t), g_t) &= \text{align}(-\nabla f(x_t), d_{K_t}) \\ &\geq \text{align}(-\nabla f(x_t), v_t - x_t) + (K_t - 1)\delta. \end{aligned}$$

Lastly,

$$\begin{aligned} \text{align}(-\nabla f(x_t), v_t - x_t) &= \frac{\langle \nabla f(x_t), x_t - v_t \rangle}{\|\nabla f(x_t)\| \|x_t - v_t\|} \\ &\geq 0 \end{aligned}$$

because $\langle \nabla f(x_t), x_t - v_t \rangle \geq 0$ since $x_t \in \mathcal{C}$.

□

D.3 Convergence analysis

Theorem D.3 (Theorem 3.2). *Let $f : \mathcal{H} \rightarrow \mathbb{R}$ be L -smooth, convex, and μ -gradient dominated, and set $\gamma_t \leftarrow \min\{\eta_t \|\nabla f(x_t)\|/(L\|g_t\|), 1\}$ or $\gamma_t \leftarrow \arg \min_{\gamma \in [0,1]} f(x_t + \gamma g_t)$. Then for all $t \in \llbracket 0, T \rrbracket$,*

$$f(x_t) - \min_{\mathcal{C}} f \leq \frac{LD^2}{2} \prod_{s=0}^{t-1} \left(1 - \eta_s^2 \frac{\mu}{L}\right)^{\mathbb{1}_{\{\gamma_s < 1\}}} \left(1 - \frac{\|g_s\|}{2\|v_s - x_s\|}\right)^{\mathbb{1}_{\{\gamma_s = 1\}}}$$

where $v_s \in \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_s), v \rangle$ for all $s \in \llbracket 0, T-1 \rrbracket$.

Proof. Let $\varepsilon_t := f(x_t) - \min_{\mathcal{C}} f$ for all $t \in \llbracket 0, T \rrbracket$. We will prove the theorem for the step-size strategy $\gamma_t \leftarrow \min\{\eta_t \|\nabla f(x_t)\|/(L\|g_t\|), 1\}$. The line search strategy follows since it achieves at least the same progress at every iteration. Let $t \in \llbracket 0, T-1 \rrbracket$. We have

$$\begin{aligned} \eta_t &= \text{align}(-\nabla f(x_t), g_t) \\ &= \frac{\langle -\nabla f(x_t), g_t \rangle}{\|\nabla f(x_t)\| \|g_t\|}. \end{aligned} \tag{18}$$

Suppose that $\gamma_t = \eta_t \|\nabla f(x_t)\|/(L\|g_t\|)$. Then since f is L -smooth and is μ -gradient dominated,

$$\begin{aligned} \varepsilon_{t+1} &\leq \varepsilon_t + \gamma_t \langle \nabla f(x_t), g_t \rangle + \frac{L}{2} \gamma_t^2 \|g_t\|^2 \\ &= \varepsilon_t - \gamma_t \eta_t \|\nabla f(x_t)\| \|g_t\| + \frac{L}{2} \gamma_t^2 \|g_t\|^2 \\ &= \varepsilon_t - \eta_t^2 \frac{\|\nabla f(x_t)\|^2}{2L} \\ &\leq \varepsilon_t - \eta_t^2 \frac{2\mu\varepsilon_t}{2L} \\ &= \left(1 - \eta_t^2 \frac{\mu}{L}\right) \varepsilon_t. \end{aligned} \tag{19}$$

Else, $\eta_t \|\nabla f(x_t)\| / (L \|g_t\|) > 1$ and $\gamma_t = 1$. By (18),

$$\begin{aligned} 1 &< \frac{\eta_t \|\nabla f(x_t)\|}{L \|g_t\|} \\ &= \frac{\langle -\nabla f(x_t), g_t \rangle}{L \|g_t\|^2} \end{aligned}$$

so

$$L \|g_t\|^2 < \langle -\nabla f(x_t), g_t \rangle.$$

Hence,

$$\begin{aligned} \varepsilon_{t+1} &\leq \varepsilon_t + \gamma_t \langle \nabla f(x_t), g_t \rangle + \frac{L}{2} \gamma_t^2 \|g_t\|^2 \\ &= \varepsilon_t + \langle \nabla f(x_t), g_t \rangle + \frac{L}{2} \|g_t\|^2 \\ &< \varepsilon_t + \frac{\langle \nabla f(x_t), g_t \rangle}{2}. \end{aligned} \tag{20}$$

Recall that $g_t = d_{K_t} / \Lambda_t$ and $d_1 = \lambda_0(v_0 - x_t)$ where $v_0 \in \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_t), v \rangle$. If $K_t = 1$ then $d_{K_t} = d_1$, else $\text{align}(-\nabla f(x_t), d_{K_t}) > \text{align}(-\nabla f(x_t), d_1)$ by the condition in Line 12. In both cases, we obtain

$$\frac{\langle -\nabla f(x_t), g_t \rangle}{\|g_t\|} \geq \frac{\langle -\nabla f(x_t), v_0 - x_t \rangle}{\|v_0 - x_t\|}. \tag{21}$$

Let $x^* \in \arg \min_{\mathcal{C}} f$. By convexity and optimality of v_0 ,

$$\begin{aligned} \varepsilon_t &= f(x_t) - f(x^*) \\ &\leq \langle \nabla f(x_t), x_t - x^* \rangle \\ &\leq \langle \nabla f(x_t), x_t - v_0 \rangle. \end{aligned} \tag{22}$$

Thus, with (20) and (21) we have

$$\varepsilon_{t+1} < \left(1 - \frac{\|g_t\|}{2\|v_0 - x_t\|}\right) \varepsilon_t.$$

Therefore, together with (19) we conclude that for all $t \in \llbracket 0, T \rrbracket$,

$$\varepsilon_t \leq \varepsilon_0 \prod_{s=0}^{t-1} \left(1 - \eta_s^2 \frac{\mu}{L}\right)^{\mathbb{1}_{\{\gamma_s < 1\}}} \left(1 - \frac{\|g_s\|}{2\|v_s - x_s\|}\right)^{\mathbb{1}_{\{\gamma_s = 1\}}}$$

where $v_s \in \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_s), v \rangle$ for all $s \in \llbracket 0, T-1 \rrbracket$. We conclude by using the smoothness of f , the definition of x_0 (Line 1), and the convexity of f :

$$\begin{aligned} \varepsilon_0 &= f(x_0) - f(x^*) \\ &\leq f(y) + \langle \nabla f(y), x_0 - y \rangle + \frac{L}{2} \|x_0 - y\|^2 - f(x^*) \\ &\leq f(y) + \langle \nabla f(y), x^* - y \rangle + \frac{LD^2}{2} - f(x^*) \\ &\leq f(x^*) + \frac{LD^2}{2} - f(x^*) \\ &= \frac{LD^2}{2}. \end{aligned}$$

□

Theorem D.4 (Theorem 3.3). Let $f : \mathcal{H} \rightarrow \mathbb{R}$ be L -smooth, convex, and μ -gradient dominated, and set $\gamma_t \leftarrow \min\{\eta_t \|\nabla f(x_t)\| / (L\|g_t\|), 1\}$ or $\gamma_t \leftarrow \arg \min_{\gamma \in [0,1]} f(x_t + \gamma g_t)$. Consider Algorithm 2 with the minor adjustment $x_{t+1} \leftarrow x_t + \gamma'_t(v_t - x_t)$ in Line 22 when $\gamma_t = 1$, where $v_t \leftarrow v_{k=0}$ is computed in Line 8 and $\gamma'_t \leftarrow \min\{\langle \nabla f(x_t), x_t - v_t \rangle / (L\|x_t - v_t\|^2), 1\}$ or $\gamma'_t \leftarrow \arg \min_{\gamma \in [0,1]} f(x_t + \gamma(v_t - x_t))$. Then for all $t \in \llbracket 0, T \rrbracket$,

$$f(x_t) - \min_C f \leq \frac{4LD^2}{t+2}.$$

Proof. We consider the line search-free strategies; the line search strategies follow since they achieve at least the same progress at every iteration. Let $\varepsilon_t := f(x_t) - \min_C f$ for all $t \in \llbracket 0, T \rrbracket$. We will show by induction that for all $t \in \llbracket 0, T \rrbracket$,

$$\varepsilon_t \leq \frac{4LD^2}{t+2}. \quad (23)$$

For the base case $t = 0$, let $x^* \in \arg \min_C f$. By smoothness of f , the definition of x_0 (Line 1), and the convexity of f , we have

$$\begin{aligned} \varepsilon_0 &= f(x_0) - f(x^*) \\ &\leq f(y) + \langle \nabla f(y), x_0 - y \rangle + \frac{L}{2} \|x_0 - y\|^2 - f(x^*) \\ &\leq f(y) + \langle \nabla f(y), x^* - y \rangle + \frac{LD^2}{2} - f(x^*) \\ &\leq f(x^*) + \frac{LD^2}{2} - f(x^*) \\ &= \frac{LD^2}{2} \end{aligned}$$

so (23) holds. Suppose that (23) holds for some $t \in \llbracket 0, T-1 \rrbracket$.

If $\gamma_t < 1$ then we can proceed as in the proof of Theorem 3.2 and obtain

$$\varepsilon_{t+1} \leq \varepsilon_t - \eta_t^2 \frac{\|\nabla f(x_t)\|^2}{2L}.$$

By Proposition 3.1(v), there exists $v_t \in \arg \min_{v \in \mathcal{V}} \langle \nabla f(x_t), v \rangle$ such that

$$\eta_t \geq \frac{\langle \nabla f(x_t), x_t - v_t \rangle}{\|\nabla f(x_t)\| \|x_t - v_t\|}.$$

By convexity of f and optimality of v_t ,

$$\begin{aligned} \varepsilon_t &= f(x_t) - f(x^*) \\ &\leq \langle \nabla f(x_t), x_t - x^* \rangle \\ &\leq \langle \nabla f(x_t), x_t - v_t \rangle. \end{aligned} \quad (24)$$

Thus,

$$\begin{aligned} \varepsilon_{t+1} &\leq \varepsilon_t - \frac{\langle \nabla f(x_t), x_t - v_t \rangle^2}{\|\nabla f(x_t)\|^2 \|x_t - v_t\|^2} \frac{\|\nabla f(x_t)\|^2}{2L} \\ &\leq \varepsilon_t - \frac{\varepsilon_t^2}{D^2} \frac{1}{2L} \\ &= \varepsilon_t \left(1 - \frac{\varepsilon_t}{2LD^2} \right). \end{aligned}$$

If $\varepsilon_t \leq 2LD^2/(t+2)$, then

$$\begin{aligned} \varepsilon_{t+1} &\leq \varepsilon_t \\ &\leq \frac{2LD^2}{t+2} \\ &\leq \frac{4LD^2}{t+3}. \end{aligned}$$

Else, $2LD^2/(t+2) < \varepsilon_t \leq 4LD^2/(t+2)$ so

$$\begin{aligned}\varepsilon_{t+1} &\leq \varepsilon_t \left(1 - \frac{\varepsilon_t}{2LD^2}\right) \\ &< \frac{4LD^2}{t+2} \left(1 - \frac{1}{2LD^2} \frac{2LD^2}{t+2}\right) \\ &= \frac{4LD^2}{t+2} \left(1 - \frac{1}{t+2}\right) \\ &\leq \frac{4LD^2}{t+3}\end{aligned}$$

so (23) holds for $t+1$.

Now consider the case $\gamma_t = 1$. Then by assumption, $x_{t+1} = x_t + \gamma'_t(v_t - x_t)$. By smoothness of f ,

$$\varepsilon_{t+1} \leq \varepsilon_t + \gamma'_t \langle \nabla f(x_t), v_t - x_t \rangle + \frac{L}{2} \gamma_t'^2 \|v_t - x_t\|^2.$$

If $\gamma'_t = \langle \nabla f(x_t), x_t - v_t \rangle / (L\|x_t - v_t\|^2)$ then

$$\begin{aligned}\varepsilon_{t+1} &\leq \varepsilon_t + \gamma'_t \langle \nabla f(x_t), v_t - x_t \rangle + \frac{L}{2} \gamma_t'^2 \|v_t - x_t\|^2 \\ &= \varepsilon_t - \frac{\langle \nabla f(x_t), x_t - v_t \rangle^2}{2L\|x_t - v_t\|^2} \\ &\leq \varepsilon_t \left(1 - \frac{\varepsilon_t}{2LD^2}\right)\end{aligned}$$

where we used (24), and we can conclude as before.

The final case to consider is $\gamma_t = 1$ and $\gamma'_t = 1$. Then $\langle \nabla f(x_t), x_t - v_t \rangle / (L\|x_t - v_t\|^2) \geq 1$ so

$$\begin{aligned}\varepsilon_{t+1} &\leq \varepsilon_t + \gamma'_t \langle \nabla f(x_t), v_t - x_t \rangle + \frac{L}{2} \gamma_t'^2 \|v_t - x_t\|^2 \\ &= \varepsilon_t + \langle \nabla f(x_t), v_t - x_t \rangle + \frac{L}{2} \|v_t - x_t\|^2 \\ &\leq \varepsilon_t + \frac{\langle \nabla f(x_t), v_t - x_t \rangle}{2} \\ &\leq \frac{\varepsilon_t}{2} \\ &\leq \frac{2LD^2}{t+2} \\ &\leq \frac{4LD^2}{t+3}\end{aligned}$$

where we used (24). Therefore (23) holds for $t+1$. □

Theorem D.5 (Theorem 3.4). *Let $f : \mathcal{H} \rightarrow \mathbb{R}$ be L -smooth, convex, and μ -gradient dominated, and set $\gamma_t \leftarrow \min\{\eta_t \|\nabla f(x_t)\| / (L\|g_t\|), 1\}$ or $\gamma_t \leftarrow \arg \min_{\gamma \in [0,1]} f(x_t + \gamma g_t)$. Assume that $|\{s \in \llbracket 0, t-1 \rrbracket \mid \gamma_s < 1, K_s > 1\}| \geq \omega t^p$ for all $t \in \llbracket 0, T-1 \rrbracket$, for some $\omega > 0$ and $p \in]0, 1]$. Then for all $t \in \llbracket 0, T \rrbracket$,*

$$f(x_t) - \min_c f \leq \frac{LD^2}{2} \exp\left(-\delta^2 \frac{\mu}{L} \omega t^p\right).$$

Proof. Let $t \in \llbracket 0, T \rrbracket$ and denote $N_t := |\{s \in \llbracket 0, t-1 \rrbracket \mid \gamma_s < 1, K_s > 1\}|$. We have $N_t \geq \omega t^p$ and, by

Proposition 3.1(v), if $K_t > 1$ then $\eta_t \geq \delta$. By Theorem 3.2,

$$\begin{aligned}
f(x_t) - \min_{\mathcal{C}} f &\leq \frac{LD^2}{2} \prod_{s=0}^{t-1} \left(1 - \eta_s^2 \frac{\mu}{L}\right)^{\mathbb{1}_{\{\gamma_s < 1\}}} \left(1 - \frac{\|g_s\|}{2\|v_s - x_s\|}\right)^{\mathbb{1}_{\{\gamma_s = 1\}}} \\
&\leq \frac{LD^2}{2} \prod_{\substack{s=0 \\ \gamma_s < 1 \\ K_s > 1}}^{t-1} \left(1 - \eta_s^2 \frac{\mu}{L}\right) \\
&\leq \frac{LD^2}{2} \left(1 - \delta^2 \frac{\mu}{L}\right)^{N_t} \\
&\leq \frac{LD^2}{2} \exp\left(-\delta^2 \frac{\mu}{L} N_t\right) \\
&\leq \frac{LD^2}{2} \exp\left(-\delta^2 \frac{\mu}{L} \omega t^p\right).
\end{aligned}$$

□

Corollary D.6 (Corollary 3.6). *In order to achieve ε -convergence, the number of linear minimizations performed over \mathcal{C} is*

$$\begin{cases} \mathcal{O}\left(\frac{LD^2 \min\{K, 1/\delta\}}{\varepsilon}\right) & \text{in the worst-case scenario} \\ \mathcal{O}\left(\min\left\{K, \frac{1}{\delta}\right\} \left(\frac{1}{\omega\delta^2} \frac{L}{\mu} \ln\left(\frac{1}{\varepsilon}\right)\right)^{1/p}\right) & \text{in the practical scenario.} \end{cases}$$

Proof. Let $t \in \llbracket 0, T-1 \rrbracket$. By Proposition 3.1(v), $\eta_t \geq (K_t - 1)\delta$, and since $\eta_t \leq 1$, we have $K_t \leq 1/\delta + 1$. Thus, $K_t \leq \min\{K, 1/\delta + 1\}$. We conclude by estimating T via Theorem 3.3 or Theorem 3.4: $T = \mathcal{O}(LD^2/\varepsilon)$ or $T = \mathcal{O}\left(\left(\frac{1}{\omega\delta^2} \frac{L}{\mu} \ln\left(\frac{1}{\varepsilon}\right)\right)^{1/p}\right)$ respectively. □

D.4 Computational experiments

Fact D.7 (Fact 4.1). *Consider \mathbb{R}^n and let $\tau > 0$. Then $\mathcal{B}_1(\tau) = \{[z]_{1:n} - [z]_{n+1:2n} \mid z \in \tau\Delta_{2n}\}$.*

Proof. Let $x \in \mathcal{B}_1(\tau)$. Define $\delta := (\tau - \|x\|_1)/2n \geq 0$ and $z \in \mathbb{R}^{2n}$ by

$$[z]_i := \begin{cases} [x]_i + \delta & \text{if } [x]_i \geq 0 \\ \delta & \text{if } [x]_i < 0 \end{cases} \quad \text{and} \quad [z]_{n+i} := \begin{cases} \delta & \text{if } [x]_i \geq 0 \\ -[x]_i + \delta & \text{if } [x]_i < 0 \end{cases}$$

for all $i \in \llbracket 1, n \rrbracket$. Then $x = [z]_{1:n} - [z]_{n+1:2n}$ and $z \geq 0$. Furthermore,

$$\begin{aligned}
1^\top z &= \sum_{i=1}^n ([z]_i + [z]_{n+i}) \\
&= \sum_{i=1}^n (|[x]_i| + 2\delta) \\
&= \|x\|_1 + 2n\delta \\
&= \tau
\end{aligned}$$

so $z \in \tau\Delta_{2n}$.

For the reverse direction, let $z \in \tau\Delta_{2n}$ and $x := [z]_{1:n} - [z]_{n+1:2n}$. Then

$$\begin{aligned}\|x\|_1 &= \sum_{i=1}^n |[z]_i - [z]_{n+i}| \\ &\leq \sum_{i=1}^n ([z]_i + [z]_{n+i}) \\ &= \tau\end{aligned}$$

so $x \in \mathcal{B}_1(\tau)$.

□