

Frank-Wolfe with a Nearest Extreme Point Oracle

Dan Garber

Technion - Israel Institute of Technology, Haifa, Israel

DANGAR@TECHNION.AC.IL

Noam Wolf

Technion - Israel Institute of Technology, Haifa, Israel

WOLFNOAM@CAMPUS.TECHNION.AC.IL

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

We consider variants of the classical Frank-Wolfe algorithm for constrained smooth convex minimization, that instead of access to the standard oracle for minimizing a linear function over the feasible set, have access to an oracle that can find an extreme point of the feasible set that is closest in Euclidean distance to a given vector. We first show that for many feasible sets of interest, such an oracle can be implemented with the same complexity as the standard linear optimization oracle. We then show that with such an oracle we can design new Frank-Wolfe variants which enjoy significantly improved complexity bounds in case the set of optimal solutions lies in the convex hull of a subset of extreme points with small diameter (e.g., a low-dimensional face of a polytope). In particular, for many 0–1 polytopes, under quadratic growth and strict complementarity conditions, we obtain the first linearly convergent variant with rate that depends only on the dimension of the optimal face and not on the ambient dimension.

1. Introduction

The Frank-Wolfe (FW) algorithm (aka the conditional gradient method) is a classical first-order method for minimizing a smooth and convex function $f(\cdot)$ over a convex and compact feasible set \mathcal{K} Frank and Wolfe (1956); Levitin and Polyak (1966); Jaggi (2013), where in this work we assume for simplicity that the underlying space is \mathbb{R}^d (though our results are applicable to any Euclidean vector space). This algorithm has regained significant interest within the machine learning and optimization communities in recent years due to the fact that, aside of access to a first-order oracle of the objective function, it only requires on each iteration to minimize a linear function over the feasible set which, in many cases of interest, is much more efficient than computing projections, as required by projected/proximal gradient methods. Another benefit of the method is that when the number of iterations is not too high, it produces iterates that are given as an explicit sparse convex combination of extreme points of the feasible set Jaggi (2013).

The well-known convergence rate of the method is $O(\beta D_{\mathcal{K}}^2/t)$, where β is the smoothness parameter of the objective, $D_{\mathcal{K}}$ is the Euclidean diameter of the set, and t is the iteration counter. It is well-known that this rate is not improvable even if the objective function is strongly convex (see for instance Lan (2013)), a property that is well known to allow for faster convergence rates, and in particular linear rates, for projected/proximal gradient methods Nesterov (2018); Beck (2017). Indeed, in recent years there is a significant research effort to design Frank-Wolfe variants with linear convergence rates under strong convexity or the weaker assumption of quadratic growth (see Definition 1 in the sequel), with most efforts focused on the case in which the feasible set is a convex and compact polytope GuéLat and Marcotte (1986); Garber and Hazan (2013, 2016); Lacoste-Julien

and Jaggi (2015); Beck and Shtern (2017); Garber and Meshi (2016); Pena et al. (2016); Pena and Rodriguez (2019); Diakonikolas et al. (2020); Garber (2020).

Despite of the above results, there is a significant disadvantage for both the standard Frank-Wolfe method and its linearly converging variants for polytopes (in comparison to projected/proximal gradient methods) that has not been addressed so far — the inherent dependency of the convergence rates on the diameter of the feasible set. First, while projected/proximal gradient methods enjoy the benefit of “warm-start” initialization, i.e., their convergence rates often depend on the distance of the initialization point from the optimal set (see for instance Nesterov (2018); Beck (2017)), for the Frank-Wolfe method we do not get such a dependence and even with a “warm-start” initialization, the rate depends on the diameter of the entire feasible set¹. Second, existing linearly convergent variants for polytopes depend on the diameter of the polytope, i.e., the convergence rate is of the form $\exp(-\Theta(D_{\mathcal{K}}^{-2}t))$, where $D_{\mathcal{K}}$ is the diameter (e.g., Garber and Hazan (2016); Lacoste-Julien and Jaggi (2015); Garber and Meshi (2016)), which is in stark contrast to proximal/projected gradient methods which, under strong convexity/quadratic growth of objective, enjoy a linear convergence rate with exponent that is independent of the diameter.

Such inferior dependence on the diameter is of great importance in many setups of interest for Frank-Wolfe-type methods. As a running example, consider polytopes that arise naturally from combinatorial structures such as the flow polytope (convex-hull of source-target paths in a directed acyclic graph), the spanning trees polytope of a graph, the perfect matchings polytope of a bipartite graph, or the base-polyhedron of a matroid. In all of these cases, solving the Frank-Wolfe linear optimization step can be done very efficiently using simple well-known combinatorial algorithms. However, for all of these polytopes the Euclidean diameter is in worst-case $\Theta(\sqrt{n})$, where n is the number of vertices in the above-mentioned graph-induced polytopes, and the size of the bases of the matroid in case of the base-polyhedron. Thus, with high-dimensional problems in mind, it is of clear interest to improve the complexity of Frank-Wolfe-type methods in terms of the diameter.

Our approach towards tackling this challenge is to consider Frank-Wolfe variants with a seemingly stronger oracle than the standard linear optimization oracle. As we shall discuss in the sequel, it turns out that in many setups of interest, this stronger oracle could be implemented with the same complexity as the standard linear optimization oracle.

Concretely, let us denote the set of extreme points of the feasible set by \mathcal{V} . The Frank-Wolfe method assumes the availability of an oracle that solves the following linear optimization problem over \mathcal{V} :

$$\arg \min_{\mathbf{v} \in \mathcal{V}} \mathbf{v}^{\top} \nabla f(\mathbf{x}), \quad (1)$$

where \mathbf{x} is some feasible point. We emphasize that the linear problem is solved over the set of extreme points (and not entire feasible set), since in most cases of interest indeed an efficient implementation of the linear optimization oracle will only consider the extreme points of the set (e.g., in all combinatorial polytopes mentioned above and for other settings of interest such as the nuclear norm ball of matrices or the set of trace-bounded positive semidefinite matrices Jaggi (2013)).

1. In Freund and Grigas (2016) it was shown that with a modified step-size, a “warm-start” could be leveraged to reduce the number of iterations required by Frank-Wolfe to reach a desired approximation error by an additive constant, however the resulting rate still depends on the diameter of the entire set.

In this paper we suggest using the following oracle which minimizes an ℓ_2 -regularized version of (1):

$$\arg \min_{\mathbf{v} \in \mathcal{V}} \mathbf{v}^\top \nabla f(\mathbf{x}) + \lambda \|\mathbf{v} - \mathbf{x}\|^2, \quad (2)$$

where $\|\cdot\|$ denotes the Euclidean norm and $\lambda > 0$. Note that (2) could be rewritten as

$$\arg \min_{\mathbf{v} \in \mathcal{V}} \left\| \mathbf{v} - \left(\mathbf{x} - \frac{1}{2\lambda} \nabla f(\mathbf{x}) \right) \right\|^2. \quad (3)$$

That is, we assume that given some point \mathbf{y} we can efficiently find the nearest extreme point (NEP) in the feasible set. We thus refer to an oracle for solving (3) as a NEP oracle. Here it is very important to emphasize that the optimization problems (2), (3) are only solved w.r.t. the set of extreme points — the set \mathcal{V} . This is very different from the regularized oracles suggested before in Migdalas (1994); Lan (2013) which solve certain regularized problems w.r.t. the entire feasible set, and thus have complexity similar to that of computing projection over the set, which is exactly the computational bottleneck that Frank-Wolfe-type methods aim to avoid. We also note that an oracle of the form (2) was implicitly already considered in Garber (2016), but there it was for the specific case in which the feasible set is the spectrahedron (unit-trace positive semidefinite matrices) and the aim was to obtain faster rates in terms of the iteration counter t .

Let us denote by \mathcal{X}^* the optimal set, that is $\mathcal{X}^* = \arg \min_{\mathbf{x} \in \mathcal{K}} f(\mathbf{x})$. We also denote $f^* = \min_{\mathbf{x} \in \mathcal{K}} f(\mathbf{x})$. Define

$$S^* \in \arg \min_{S \subseteq \mathcal{V}} \left\{ \max_{\mathbf{u}, \mathbf{v} \in S} \|\mathbf{v} - \mathbf{u}\| : \mathcal{X}^* \subseteq \text{conv}(S) \right\}, \quad D^* = \max_{\mathbf{u}, \mathbf{v} \in S^*} \|\mathbf{v} - \mathbf{u}\|. \quad (4)$$

That is, S^* is a subset of extreme points of minimum diameter whose convex-hull contains the optimal set \mathcal{X}^* , and D^* is the corresponding diameter. Also, when \mathcal{K} is a convex and compact polytope, we let \mathcal{F}^* denote the lowest-dimensional face of \mathcal{K} such that $\mathcal{X}^* \subseteq \mathcal{F}^*$ and we let $D_{\mathcal{F}^*}$ denote the Euclidean diameter of \mathcal{F}^* . As we shall discuss in detail in the sequel, in many cases of interest we have that $D^* \ll D_{\mathcal{K}}$ or $D_{\mathcal{F}^*} \ll D_{\mathcal{K}}$. For instance, when there is a unique optimal solution which is an extreme point we have that $D^* = 0$, or very often in case \mathcal{K} is a polytope and $\dim \mathcal{F}^* \ll d$ — a natural notion of sparsity for polytopes.

Our main contributions, in an informal and simplified presentation, are as follows:

1. We show that for many feasible sets of interest the NEP oracle (3) could be implemented with the same complexity as the standard optimization linear oracle (1). Cases of interest include 0–1 polytopes, nuclear norm balls of matrices, bound-trace positive semidefinite matrices, the unit spectral norm ball of matrices, and unstructured convex hulls. See Section 1.2.
2. We present a natural variant of the Frank-Wolfe method which replaces the linear optimization oracle with a NEP oracle and enjoys an improved rate of $O(\beta(D^{*2} + D_L^2)/t)$, where D_L is the diameter of the initial level set.² Assuming quadratic growth of the objective w.r.t. the feasible set, this rate improves to $O(\beta D^{*2}/t)$ plus a lower-order term of the form $\log(t)/t^2$. We also show such rates are not possible to obtain via Frank-Wolfe variants using a linear optimization oracle. See Theorems 2 and 4.

2. That is $D_L = \max\{\|\mathbf{u} - \mathbf{v}\| \mid \mathbf{u} \in \mathcal{K}, \mathbf{v} \in \mathcal{K}, f(\mathbf{u}) \leq f(\mathbf{x}_1), f(\mathbf{v}) \leq f(\mathbf{x}_1)\}$, where \mathbf{x}_1 is the initialization point.

3. In case \mathcal{K} is a polytope and quadratic growth holds, we present a linearly convergent FW variant which replaces the linear optimization oracle with a NEP oracle and enjoys a rate of the form $\exp(-t/(dD_{\mathcal{F}^*}^2))$, which improves upon the previous best $\exp(-t/(dD_{\mathcal{K}}^2))$ (e.g., Garber and Hazan (2013, 2016); Lacoste-Julien and Jaggi (2015)).³ For many polytopes of interest this gives the first linearly convergent algorithm with rate independent of the diameter of the entire polytope and whose per-iteration oracle complexity matches that of Frank-Wolfe. See Theorem 5.
4. In case \mathcal{K} is a polytope and both quadratic growth and strict complementarity (see definition in the sequel) hold, our linearly convergent variant converges with rate $\exp(-t/(\dim \mathcal{F}^* D_{\mathcal{F}^*}^2))$, where $\dim \mathcal{F}^*$ is the dimension of the optimal face \mathcal{F}^* . As a consequence, for many 0–1 polytopes we obtain the first FW variant whose convergence rate is independent of the dimension, provided that \mathcal{F}^* is low-dimensional. See Theorem 6.
5. We demonstrate that a NEP oracle could also lead to similar improvements to those in item 2 above in the stochastic setting, by analyzing a variant of the Stochastic Frank-Wolfe method. See Theorem 12.

1.1. Notation

We use boldface lowercase letters to denote vectors and lightface letters to denote scalars. When considering the space of matrices $\mathbb{R}^{m \times n}$ or that of symmetric $n \times n$ matrices \mathbb{S}^n , we use boldface uppercase letters to denote matrices. Throughout, we let $\|\cdot\|$ denote the Euclidean norm. For matrices we let $\|\cdot\|_2$ denote the spectral norm (i.e., largest singular value) and $\|\cdot\|_F$ denote the (Euclidean) Frobenius norm. For a set S of points in a Euclidean vector space we let $\text{conv}(S)$ denote their convex-hull. Given a point \mathbf{x} and convex and compact set S in a Euclidean vector space, we let $\text{dist}(\mathbf{x}, S)$ denote the Euclidean distance of \mathbf{x} from S .

1.2. Examples of feasible sets of interest

0–1 polytopes: 0–1 polytopes are polytopes which satisfy $\mathcal{V} \subseteq \{0, 1\}^d$, i.e., all vertices have either 0 or 1 entries. This family of polytopes captures many combinatorial polytopes of interest including the flow polytope, the spanning-tree polytope, the perfect matchings polytope of a bipartite graph, the base polyhedron of a matroid, the unit simplex, the hypercube $[0, 1]^d$, and many more.

For such polytopes, the NEP oracle could be implemented directly using a linear optimization oracle since for any vector \mathbf{y} we have

$$\arg \min_{\mathbf{v} \in \mathcal{V}} \|\mathbf{v} - \mathbf{y}\|^2 = \arg \min_{\mathbf{v} \in \mathcal{V}} \|\mathbf{v}\|^2 - 2\mathbf{v}^\top \mathbf{y} = \arg \min_{\mathbf{v} \in \mathcal{V}} \mathbf{v}^\top \mathbf{1} - 2\mathbf{v}^\top \mathbf{y} = \arg \min_{\mathbf{v} \in \mathcal{V}} \mathbf{v}^\top (\mathbf{1} - 2\mathbf{y}),$$

where $\mathbf{1}$ denotes the all-ones vector.

For many of these polytopes (e.g., flow polytope, perfect matchings polytope of a bipartite graph, hypercube, but not the simplex), the diameter of the optimal face scales with its dimension, e.g., $\text{diam}(\mathcal{F}^*) = \Theta(\sqrt{\dim \mathcal{F}^*})$. Thus, when the optimal set \mathcal{X}^* is contained within a low-dimensional face \mathcal{F}^* , we indeed have that $D^* \leq D_{\mathcal{F}^*} \ll D_{\mathcal{K}}$. Low-dimensionality of the optimal

3. Here for simplicity we hide dependencies on certain geometric quantities of the polytope which are standard for such results, as well as on the objective's condition number β/α , where α is the quadratic growth parameter.

face can be seen as a natural notion of sparsity (since it implies any optimal solution can be expressed as a sparse convex combination of extreme points), which is an important concept in many machine learning / statistical settings (see also discussions in the recent work [Garber \(2020\)](#)).

The spectrahedron and trace-norm balls: The spectrahedron $\{\mathbf{X} \in \mathbb{S}^n : \mathbf{X} \succeq 0, \text{Tr}(\mathbf{X}) = \tau\} = \text{conv}\{\tau\mathbf{u}\mathbf{u}^\top \mid \mathbf{u} \in \mathbb{R}^n, \|\mathbf{u}\| = 1\}$ and the nuclear norm ball $\{\mathbf{X} \in \mathbb{R}^{m \times n} \mid \|\mathbf{X}\|_* \leq \tau\} = \text{conv}\{\tau\mathbf{u}\mathbf{v}^\top \mid \mathbf{u} \in \mathbb{R}^m, \mathbf{v} \in \mathbb{R}^n, \|\mathbf{u}\| = \|\mathbf{v}\| = 1\}$, where $\tau > 0$, are two highly popular convex relaxations for matrix rank constraint, and are ubiquitous in convex relaxations for low-rank matrix recovery problems (e.g., low-rank matrix completion). Optimization over these sets is one of the main reasons for the increasing popularity of Frank-Wolfe-type methods, since linear optimization over these sets amounts to a rank-one SVD computation — a task for which there exists very efficient iterative methods (e.g., power iterations, Lanczos algorithm), while Euclidean projection requires in general a full-rank SVD computation [Jaggi and Sulovský \(2010\)](#); [Garber \(2016\)](#); [Allen-Zhu et al. \(2017\)](#). Since all extreme points of these sets have the same Euclidean norm, the NEP oracle is equivalent to the linear optimization oracle — see *Norm-uniform sets* in the sequel, and hence could be implemented with the same complexity.

Consider now the case in which the feasible set is the spectrahedron and suppose that the optimal solution is unique and can be written in the form $\mathbf{X}^* = \sum_{i=1}^m \lambda_i \tau \mathbf{u}_i \mathbf{u}_i^\top$, where $(\lambda_1, \dots, \lambda_m)$ is in the unit simplex, $\|\mathbf{u}_i\| = 1, i = 1, \dots, m$, and for all $i \neq j$ we have $(\mathbf{u}_i^\top \mathbf{u}_j)^2 \geq 1 - \gamma$, for some $\gamma > 0$. In this case, it follows that $D^* \leq \max_{i,j} \|\tau \mathbf{u}_i \mathbf{u}_i^\top - \tau \mathbf{u}_j \mathbf{u}_j^\top\|_F \leq \sqrt{2\gamma}\tau$. Thus, for $\gamma \ll 1$ (which means \mathbf{X}^* admits a crude approximation via a rank-one matrix), we can indeed have $D^* \ll \sqrt{2}\tau = D_{\mathcal{K}}$. Clearly, a similar argument holds for the nuclear norm ball as well.

Unit spectral norm ball: Consider the set of matrices $\mathcal{K} = \{\mathbf{X} \in \mathbb{R}^{m \times n} \mid \|\mathbf{X}\|_2 \leq 1\} = \text{conv}\{\mathbf{U} \in \mathbb{R}^{m \times n} \mid \mathbf{U}^\top \mathbf{U} = \mathbf{I}\}$, $m \geq n$. Linear optimization over this set amounts to a SVD computation [Jaggi \(2013\)](#). Since, as in the previous example, all extreme points have the same Euclidean norm, the NEP oracle is equivalent to the linear optimization oracle (see *Norm-uniform sets* next). While for this set we have $D_{\mathcal{K}} = 2\sqrt{n}$, similarly to the example for the spectrahedron above, we can clearly have $D^* \ll 2\sqrt{n}$.

Norm-uniform sets: In case all extreme points have the same Euclidean norm i.e., $\|\mathbf{u}\| = \|\mathbf{v}\|$ for all $\mathbf{u}, \mathbf{v} \in \mathcal{V}$, we clearly have that the NEP oracle could be implemented using a single call to the standard linear optimization oracle since for any vector \mathbf{y} :

$$\arg \min_{\mathbf{v} \in \mathcal{V}} \|\mathbf{v} - \mathbf{y}\|^2 = \arg \min_{\mathbf{v} \in \mathcal{V}} \|\mathbf{v}\|^2 - 2\mathbf{v}^\top \mathbf{y} = \arg \min_{\mathbf{v} \in \mathcal{V}} \mathbf{v}^\top (-2\mathbf{y}).$$

Note that many of the above examples (e.g., perfect matchings polytope, base polyhedron of a matroid, spectrahedron, nuclear norm ball, unit spectral norm ball etc.) satisfy this property. Clearly, ℓ_1, ℓ_2 and ℓ_∞ balls in \mathbb{R}^d also satisfy this property.

Unstructured convex-hulls: In case the feasible set \mathcal{K} is given by its set of extreme points without further structure, i.e., $\mathcal{K} = \text{conv}\{\mathcal{V}\}$, where the set \mathcal{V} is explicitly given, linear optimization amounts to computing the linear function over all extreme points and taking the minimum. Clearly, we can also implement the NEP oracle with the same complexity.

1.3. The quadratic growth property

Most (but not all) results we report in this paper hold under the quadratic growth property which has similar consequences to strong convexity but is a weaker assumption.

Definition 1 (quadratic growth) *We say a function $f : \mathcal{K} \rightarrow \mathbb{R}$ satisfies the quadratic growth property with parameter $\alpha > 0$ w.r.t. a convex and compact set \mathcal{K} if for all $\mathbf{x} \in \mathcal{K}$ it holds that $\text{dist}(\mathbf{x}, \mathcal{X}^*)^2 \leq 2\alpha^{-1}(f(\mathbf{x}) - f^*)$.*

Quadratic growth is known to hold whenever $f(\mathbf{x})$ is of the form $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x}) + \mathbf{b}^\top \mathbf{x}$ for α_g -strongly convex $g : \mathbb{R}^m \rightarrow \mathbb{R}$, $\mathbf{A} \in \mathbb{R}^{m \times d}$, and \mathcal{K} is a convex and compact polytope (see for instance Beck and Shtern (2017); Garber (2019a)). Very recently, it was also established that for the important matrix domains — the spectrahedron and nuclear norm ball (mentioned above), for $f(\cdot)$ of this form which also satisfies a certain strict complementarity condition, quadratic growth also holds, see Ding et al. (2020b,a); Garber (2019b). Note that such structure of $f(\cdot)$ holds in particular for $g(\mathbf{z}) = \|\mathbf{z} - \mathbf{y}\|^2$ and \mathbf{A} being an underdetermined linear map which captures some of the most fundamental problems in statistics and machine learning.

2. Main Results

In this section we present our novel algorithms and corresponding convergence rates. The complete proofs, as well as additional results, are given in the appendix.

2.1. A NEP Oracle-based Frank-Wolfe Variant for General Convex Sets

Our first line of results concerns a straightforward adaptation of the standard Frank-Wolfe method, where the call to the linear optimization oracle is replaced with a call to the NEP oracle, see Algorithm 1 below. We note that we use a modified step-size $\tilde{\eta}_t$ for the convex combination on each iteration (and not the one used to compute the new extreme point \mathbf{v}_t) in order to guarantee that our method is a decent method which is important for achieving the improved dependencies on the initialization point.

Algorithm 1 Frank-Wolfe with Nearest Extreme Point Oracle

- 1: Input: sequence of step-sizes $\{\eta_t\}_{t \geq 1} \subset [0, 1]$
 - 2: $\mathbf{x}_1 \leftarrow$ some arbitrary point in \mathcal{V}
 - 3: **for** $t = 1, 2, \dots$ **do**
 - 4: $\mathbf{v}_t \leftarrow \arg \min_{\mathbf{v} \in \mathcal{V}} \mathbf{v}^\top \nabla f(\mathbf{x}_t) + \frac{\beta \eta_t}{2} \|\mathbf{x}_t - \mathbf{v}\|^2$
 {equivalent to $\mathbf{v}_t \leftarrow \arg \min_{\mathbf{v} \in \mathcal{V}} \|\mathbf{v} - (\mathbf{x}_t - (\beta \eta_t)^{-1} \nabla f(\mathbf{x}_t))\|^2$ }
 - 5: pick $\tilde{\eta}_t \in [0, 1]$ such that $f((1 - \tilde{\eta}_t)\mathbf{x}_t + \tilde{\eta}_t \mathbf{v}_t) \leq \min\{f((1 - \eta_t)\mathbf{x}_t + \eta_t \mathbf{v}_t), f(\mathbf{x}_t)\}$
 - 6: $\mathbf{x}_{t+1} \leftarrow (1 - \tilde{\eta}_t)\mathbf{x}_t + \tilde{\eta}_t \mathbf{v}_t$
 - 7: **end for**
-

Theorem 2 *Using Algorithm 1 with step-size $\eta_t = \frac{2}{t+1}$ we have*

$$\forall t \geq 2 : \quad f(\mathbf{x}_t) - f^* \leq \frac{2\beta(D^{*2} + D_L^2)}{t+1},$$

where D_L is the diameter of the initial level set (see Footnote 2).

Moreover, if $f(\cdot)$ has the quadratic growth property over \mathcal{K} with parameter $\alpha > 0$, then

$$\forall t \geq 2: \quad f(\mathbf{x}_t) - f^* \leq \frac{2\beta D^{*2}}{t+1} + \frac{8\beta^2}{\alpha} (D^{*2} + \min\{2\alpha^{-1}(f(\mathbf{x}_1) - f^*), D_L^2\}) \frac{\log(t)}{t^2}.$$

Theorem 2 shows that as opposed to the standard convergence rate of the Frank-Wolfe method which, regardless of the initialization, is $O(\beta D_{\mathcal{K}}^2/t)$, our NEP oracle-based variant has a rate of the form $O(\beta D^{*2}/t)$ plus an additional natural term that depends on the quality of the initialization point. In particular, under quadratic growth, this additional term decays at a fast rate of $O(\log t/t^2)$, and thus even without good initialization, our algorithm has significantly improved complexity whenever $D^* \ll D_{\mathcal{K}}$.

2.1.1. COMPLEMENTARY LOWER BOUND FOR LINEAR OPTIMIZATION-BASED FW VARIANTS

We now present a complementary result showing that the rates reported in Theorem 2 are impossible to obtain in general for Frank-Wolfe variants using only a linear optimization oracle.

Definition 3 (Frank-Wolfe-type method (see also Garber (2020))) *An iterative algorithm for the optimization problem $\min_{\mathbf{x} \in \mathcal{K}} f(\mathbf{x})$, where \mathcal{K} is convex and compact and $f(\cdot)$ is smooth and convex, is a Frank-Wolfe-type method if on each iteration t , it performs a single call to the linear optimization oracle of \mathcal{K} w.r.t. the point $\nabla f(\mathbf{x}_t)$, i.e., computes some $\mathbf{v}_t \in \arg \min_{\mathbf{v} \in \mathcal{K}} \mathbf{v}^\top \nabla f(\mathbf{x}_t)$, where \mathbf{x}_t is the current iterate, and produces the next iterate \mathbf{x}_{t+1} by taking some convex combination of the points in $\{\mathbf{x}_1, \mathbf{v}_1, \dots, \mathbf{v}_t\}$, where \mathbf{x}_1 is the initialization point.*

Theorem 4 *Let $\mathcal{K} = [0, 1]^d$, and fix a positive integer $m < d$. Let $\mathbf{x}^* = \frac{1}{2} \sum_{i=1}^m \mathbf{e}_i$, i.e., \mathbf{x}^* has $\frac{1}{2}$ for the first m coordinates and 0 for the rest. Now consider the minimization of the function $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{x} - \mathbf{x}^*\|^2$ over \mathcal{K} starting at $\mathbf{x}_1 = \mathbf{e}_{m+1}$. Then, for any Frank-Wolfe-type method there exists a sequence of answers returned by the linear optimization oracle such that for any $t \leq \lfloor \sqrt{d-m-1} \rfloor$, the t th iterate of the algorithm \mathbf{x}_t satisfies $f(\mathbf{x}_t) - f^* \geq \frac{1}{4}$.*

Note that for the problem described in Theorem 4 it holds that $f(\mathbf{x}_1) - f^* = \frac{1}{2} + \frac{m}{8}$, $D^* = \sqrt{m}$, and $\alpha = \beta = 1$. Thus, when $m \ll \sqrt{d}$, we have that $O(m)$ iterations suffice for Algorithm 1 to obtain approximation error $< 1/4$, while any Frank-Wolfe-type method with a linear optimization oracle will require $\Omega(\sqrt{d})$ iterations. Moreover, all currently existing upper-bounds for Frank-Wolfe-type methods actually require $\Omega(d)$ iterations (since $D_{\mathcal{K}}^2 = d$ in this case).

2.2. A NEP Oracle-based Linearly-Convergent Variant for Polytopes

We now turn to discuss our NEP oracle-based linearly convergent algorithm for the case in which the feasible set \mathcal{K} is a convex and compact polytope. Our algorithm (see Algorithm 2) is an adaptation of the *fully-corrective* Frank-Wolfe variant Jaggi (2013), where the linear optimization step is replaced in a straightforward manner with the new NEP oracle (similarly to Algorithm 1). On each iteration of the algorithm we optimize either the quadratic-upper bound on the objective due to smoothness (Option 1) or the objective itself (Option 2) over the convex-hull of all previously found vertices.

We make some comments regarding the efficient solution of the quadratic optimization problem in lines 7 and 9 of Algorithm 2. Regarding line 7, when the number of vertices in the decomposition of the current iterate k satisfies $k \ll d$, using a preprocessing step (i.e., explicitly computing

Algorithm 2 Linearly Convergent Frank-Wolfe for Polytopes with a NEP Oracle

- 1: Input: a sequence $\{\rho_t\}_{t \geq 1} \subset [0, 1]$
 - 2: $\mathbf{x}_1 \leftarrow$ some arbitrary point in \mathcal{V}
 - 3: **for** $t = 1, 2, \dots$ **do**
 - 4: Let $\sum_{i=1}^k \lambda_i \mathbf{v}_i$ be an explicitly maintained decomposition of \mathbf{x}_t to vertices
 - 5: $\mathbf{v}_{k+1} \leftarrow \arg \min_{\mathbf{u} \in \mathcal{V}} \mathbf{u}^\top \nabla f(\mathbf{x}_t) + \beta \rho_t \|\mathbf{u} - \mathbf{x}_t\|^2$
 - 6: **Option 1:**
 - 7: $\mathbf{x}_{t+1} \leftarrow \arg \min_{\mathbf{x} \in \text{conv}(\mathbf{v}_1, \dots, \mathbf{v}_{k+1})} \mathbf{x}^\top \nabla f(\mathbf{x}_t) + \frac{\beta}{2} \|\mathbf{x} - \mathbf{x}_t\|^2$
 - 8: **Option 2 (fully-corrective):**
 - 9: $\mathbf{x}_{t+1} \leftarrow \arg \min_{\mathbf{x} \in \text{conv}(\mathbf{v}_1, \dots, \mathbf{v}_{k+1})} f(\mathbf{x})$
 - 10: **end for**
-

$\mathbf{v}_i^\top \nabla f(\mathbf{x}_t)$, $\mathbf{v}_i^\top \mathbf{x}_t$, $\mathbf{v}_i^\top \mathbf{v}_j$, $i = 1, \dots, k+1, j = 1, \dots, k+1$), this problem could be reformulated as convex quadratic minimization over the k -dimensional unit simplex. Such problems can be efficiently solved via fast first-order methods. The problem in line 9 can also be reformulated as a convex problem over the k -dimensional unit simplex however, solving it via first-order methods requires more evaluations of the objective's gradients.

In order to present our guarantees for Algorithm 2, we require some additional notation. Following Garber and Hazan (2016); Garber (2020) we assume \mathcal{K} is a convex and compact polytope in the form $\mathcal{K} := \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{A}_1 \mathbf{x} = \mathbf{b}_1, \mathbf{A}_2 \mathbf{x} \leq \mathbf{b}_2\}$, $\mathbf{A}_1 \in \mathbb{R}^{m_1 \times d}$, $\mathbf{A}_2 \in \mathbb{R}^{m_2 \times d}$. For a face \mathcal{F} of \mathcal{K} we define:

$$\dim \mathcal{F} := d - \dim \text{span}\{\{\mathbf{A}_1(1), \dots, \mathbf{A}_1(m_1)\} \cup \{\mathbf{A}_2(i) : i \in [m_2], \forall \mathbf{x} \in \mathcal{F} : \mathbf{A}_2(i)^\top \mathbf{x} = \mathbf{b}_2(i)\}\}.$$

We let $\mathcal{F}^* \subseteq \mathcal{K}$ denote the lowest-dimensional face of \mathcal{P} containing the set of optimal solutions, i.e., $\mathcal{X}^* \subseteq \mathcal{F}^*$. In the following we write $\mathcal{F}^* = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{A}_1^* \mathbf{x} = \mathbf{b}_1^*, \mathbf{A}_2^* \mathbf{x} \leq \mathbf{b}_2^*\}$.⁴ We let \mathbb{A}^* denote the set of all $\dim \mathcal{F}^* \times d$ matrices whose rows are linearly independent rows chosen from the rows of \mathbf{A}_2^* . We define $\psi = \max_{\mathbf{M} \in \mathbb{A}^*} \|\mathbf{M}\|_2$ and $\xi = \min_{\mathbf{v} \in \mathcal{V} \cap \mathcal{F}^*} \min_i \{\mathbf{b}_2^*(i) - \mathbf{A}_2^*(i)^\top \mathbf{v} \mid \mathbf{b}_2^*(i) > \mathbf{A}_2^*(i)^\top \mathbf{v}\}$. Finally, we let $D_{\mathcal{F}^*}$ denote the diameter of the optimal face.

Theorem 5 *Suppose that \mathcal{K} is a convex and compact polytope and quadratic growth holds with parameter $\alpha > 0$. Let $C \geq f(\mathbf{x}_1) - f^*$ and $M \geq \max\{\frac{\beta}{\alpha}(4 + 8d\mu^2 D_{\mathcal{F}^*}^2), \frac{1}{2}\}$, where $\mu = \frac{\psi}{\xi}$.⁵*

Using Algorithm 2 with parameter $\rho_t = \frac{\min\{\sqrt{\frac{2Cd\mu^2}{\alpha}} \exp\left(-\frac{1}{4M}(t-1)\right), 1\}}{2M}$ for all $t \geq 1$ one has,

$$\forall t \geq 1 : \quad f(\mathbf{x}_t) - f^* \leq C \exp\left(-\frac{t-1}{4M}\right).$$

4. The rows of \mathbf{A}_1^* are exactly the rows of \mathbf{A}_1 plus rows of \mathbf{A}_2 which correspond to constraints that are tight for all points in \mathcal{F}^* and the vector \mathbf{b}_1^* is defined accordingly. The rows of \mathbf{A}_2^* are the rows of \mathbf{A}_2 which correspond to constraints that are satisfied by some of the points in \mathcal{F}^* but not by others, and the vector \mathbf{b}_2^* is defined accordingly.

5. Note that for polytopes such as the flow polytope, the perfect matchings polytope of a bipartite graph, the $[0, 1]^d$ hypercube and the unit simplex it holds that $\mu = 1$.

Theorem 5 improves upon the state-of-the-art complexity bounds for Frank-Wolfe-type methods for general polytopes Garber and Hazan (2016)⁶ by replacing the dependency on $D_{\mathcal{K}}^2$ with $D_{\mathcal{F}^*}^2$. For instance, for the $[0, 1]^d$ hypercube, when $\dim \mathcal{F}^* \ll d$ we have $D_{\mathcal{F}^*}^2 = \Theta(\dim \mathcal{F}^*) \ll D_{\mathcal{K}}^2 = \Theta(d)$.

We note that while Theorem 5 relies on a pre-defined sequence of step-sizes $\{\rho_t\}_{t \geq 1}$ which can be difficult to tune in practice, in Theorem 11 (Section C.2) we prove that the same rate can be obtained using an adaptive step-size by applying a logarithmic-scale search on each iteration t to choose a value for ρ_t which gives the largest decrease in function value.

2.2.1. IMPROVED DEPENDENCE ON DIMENSION UNDER STRICT COMPLEMENTARITY

While for many polytopes Theorem 5 implies significant improvement in the dependence on the dimension whenever $\dim \mathcal{F}^* \ll d$, still the exponent has explicit dependence on the dimension d .

In a very recent work Garber (2020) it was shown that even when the optimal face is low-dimensional, without further assumptions, Frank-Wolfe-type methods (as defined in Definition 3) cannot avoid such dependence. It was also shown that under a strict complementarity condition (see Assumption 1), it is possible to improve the explicit dependence on the dimension d to only dependence on the dimension of the optimal face \mathcal{F}^* . The strict complementarity assumption, in the context of analyzing Frank-Wolfe-type methods, was suggested by Wolfe himself Wolfe (1970), and it was also instrumental in the early work GuéLat and Marcotte (1986) on linearly-converging Frank-Wolfe methods, but not in the more modern ones such as Garber and Hazan (2013, 2016); Lacoste-Julien and Jaggi (2015). Garber (2020) motivated this assumption by proving it implies the robustness of the optimal face \mathcal{F}^* to small perturbations in the objective function $f(\cdot)$.

Assumption 1 (strict complementarity) *There exist $\delta > 0$ such that for all $\mathbf{x}^* \in \mathcal{X}^*$ and $\mathbf{v} \in \mathcal{V}$: if $\mathbf{v} \in \mathcal{V} \setminus \mathcal{F}^*$ then $(\mathbf{v} - \mathbf{x}^*)^\top \nabla f(\mathbf{x}^*) \geq \delta$; otherwise, if $\mathbf{v} \in \mathcal{V} \cap \mathcal{F}^*$ then $(\mathbf{v} - \mathbf{x}^*)^\top \nabla f(\mathbf{x}^*) = 0$.*

Theorem 6 *Suppose that in addition to the assumptions of Theorem 5, Assumption 1 also holds with some parameter $\delta > 0$, and let $C \geq f(\mathbf{x}_1) - f^*$, $M_1 \geq \max\{\frac{4\beta}{\alpha} + 8\beta D_{\mathcal{F}^*}^2 \max\{2\kappa, \delta^{-1}\}, \frac{1}{2}\}$, and $M_2 \geq \{\frac{4\beta}{\alpha} + 16\beta\kappa D_{\mathcal{F}^*}^2, \frac{1}{2}\}$, where $\kappa = \frac{2\mu^2 \dim \mathcal{F}^*}{\alpha}$.*

Using Algorithm 2 with parameters $\rho_t = \frac{\min\{\sqrt{2 \max\{2\kappa, \delta^{-1}\} C \exp(-\frac{1}{4M_1}(t-1))}, 1\}}{2M_1}$ for all $t \geq 1$, one has,

$$\forall t \geq 1 : f(\mathbf{x}_t) - f^* \leq C \exp\left(-\frac{t-1}{4M_1}\right). \quad (5)$$

Furthermore, in case that $2\kappa \leq \delta^{-1}$ and $\dim \mathcal{F}^ > 0$ (i.e. $\kappa > 0$), denoting $\tau \geq 4M_1 \log(\frac{C}{\delta^2 \kappa}) + 1$ and for all $t \geq \tau$, using $\rho_t = \frac{\min\{2\delta\kappa \exp(-\frac{t-\tau}{8M_2}), 1\}}{2M_2}$ instead of the above value, one has,*

$$\forall t \geq \tau : f(\mathbf{x}_t) - f^* \leq \delta^2 \kappa \exp\left(-\frac{t-\tau}{4M_2}\right). \quad (6)$$

6. We note that i. while other results on linearly converging FW variants have complexity bounds that are stated using different quantities, such as the *pyramidal width* in Lacoste-Julien and Jaggi (2015), their worst-case complexity bounds do not improve over Garber and Hazan (2016), and ii. while Garber and Meshi (2016) presented a FW variant with improved dependence on the dimension (but without improvement in dependence on $D_{\mathcal{K}}$), their result applies only to a very restricted family of polytopes and in particular a strict subset of the 0–1 polytopes.

Theorem 6 improves upon Garber (2020) by replacing the dependence on $D_{\mathcal{K}}$ with $D_{\mathcal{F}^*}$. Thus, for many polytopes of interest, when $\dim \mathcal{F}^* \ll d$, Theorem 6 has no explicit dependence on the ambient dimension d , but only on the dimension of the low-dimensional optimal face \mathcal{F}^* . In particular, as discussed, for many polytopes of interest, the NEP oracle could be implemented using a single call to the linear optimization oracle, and as a result, for many 0–1 polytopes, under quadratic growth and strict complementarity, we obtain the first linearly convergent algorithm with the same per-iteration oracle-complexity as Frank-Wolfe and with dimension-independent convergence rate whenever $\dim \mathcal{F}^* \ll d$.

2.3. A NEP Oracle-based Frank-Wolfe Variant for Stochastic Optimization

Our last result concerns a standard stochastic optimization setting in which $f(\cdot)$ is given by a stochastic first-order oracle with stochastic gradients upper-bounded in ℓ_2 norm by some $G > 0$.

Stochastic Frank-Wolfe-type methods have also received notable attention in recent years, including the use of the conditional-gradient sliding approach and various variance reduction methods, see for instance Lan and Zhou (2016); Hazan and Luo (2016). Here however, in order to demonstrate the benefit of our NEP oracle to this setting as well, we only focus on the most basic method known as the Stochastic Frank-Wolfe (SFW) algorithm, which replaces the exact gradient in the Frank-Wolfe method with a mini-batched stochastic gradient (see Hazan and Luo (2016)). Also, here for simplicity we only focus on the case in which $f(\cdot), \mathcal{K}$ satisfy the quadratic growth property.

Our algorithm, which is a straight-forward adaptation of the SFW algorithm is given as Algorithm 3 (replacing the call to the linear optimization oracle with a call to the NEP oracle) in Appendix 2.3.

Theorem 7 *Suppose f, \mathcal{K} satisfy the quadratic growth property with some $\alpha > 0$. Using Algorithm 3 with step-size $\eta_t = \frac{2}{t+1}$ and mini-batch sizes that satisfy*

$$m_t \geq \max \left\{ \left(\frac{G(t+1)}{\beta D_{\mathcal{K}}} \right)^2, \min \left\{ \left(\frac{GD_{\mathcal{K}}(t+1)}{\beta D^{*2}} \right)^2, \left(\frac{\alpha G(t+1)^2}{8\beta^2 D_{\mathcal{K}}} \right)^2 \right\} \right\},$$

for any $0 < \epsilon \leq \frac{16\beta^2 D_{\mathcal{K}}^2}{\alpha}$, expected approximation error ϵ is achieved after $\tilde{O}(\max\{\frac{\beta D^{*2}}{\epsilon}, \frac{\beta D_{\mathcal{K}}}{\sqrt{\alpha\epsilon}}\})$ calls to the NEP oracle and $\tilde{O}(\beta G^2 \max\{\frac{D_{\mathcal{K}}^2 D^{*2}}{\epsilon^3}, \frac{D_{\mathcal{K}}}{\alpha^{3/2} \epsilon^{3/2}}, \frac{D_{\mathcal{K}}^3}{\alpha^{1/2} \epsilon^{5/2}}\})$ stochastic gradient evaluations, where \tilde{O} suppresses poly-logarithmic terms in $\frac{\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}$.

Let us compare Theorem 7 with the standard SFW method which requires $O(\beta G^2 D_{\mathcal{K}}^4 / \epsilon^3)$ stochastic gradients and $O(\beta D_{\mathcal{K}}^2 / \epsilon)$ calls to the linear optimization oracle Hazan and Luo (2016), and the state-of-the-art — the stochastic conditional gradient sliding (SCGS) method Lan and Zhou (2016) which requires optimal $O(G^2 / (\alpha\epsilon))$ stochastic gradients and $O(\beta D_{\mathcal{K}}^2 / \epsilon)$ calls to the linear optimization oracle. The improvement over SFW is quite clear, both in terms of the stochastic oracle and the optimization oracle (at least when α is not trivially small). While SCGS has clear advantage in terms of the stochastic oracle complexity⁷, we see that when the main concern is the optimization oracle complexity and $D^* \ll D_{\mathcal{K}}$, already the simple stochastic scheme in Algorithm 3 can have significant advantage over SCGS.

7. This is not surprising since, as opposed to SCGS, SFW cannot leverage the quadratic growth to improve the stochastic gradient complexity.

3. Proof Ideas

In this section we describe the main novel components in the analysis of Algorithms 1,2 — novel bounds on the per-iteration error reduction which are independent of the diameter of the set.

3.1. Error reduction for Algorithm 1

Lemma 8 *Using Algorithm 1 with any choice of step-sizes $\{\eta_t\}_{t=1}^\infty$ one has,*

$$\forall t \geq 1 : \quad f(\mathbf{x}_{t+1}) - f^* \leq (1 - \eta_t)(f(\mathbf{x}_t) - f^*) + \frac{\beta\eta_t^2}{2}(\text{dist}(\mathbf{x}_t, \mathcal{X}^*)^2 + D^{*2}).$$

Proof Fix some iteration $t \geq 1$ and let \mathbf{x}^* be the optimal solution closest to \mathbf{x}_t . From the β -smoothness of $f(\cdot)$, the optimality of \mathbf{v}_t , and the definition of the set S^* in (4), we have that,

$$\begin{aligned} f(\mathbf{x}_{t+1}) &\leq \min_{\mathbf{v} \in S^*} f(\mathbf{x}_t) + \eta_t(\mathbf{v} - \mathbf{x}_t)^\top \nabla f(\mathbf{x}_t) + \frac{\beta\eta_t^2}{2} \|\mathbf{v} + \mathbf{x}^* - \mathbf{x}^* - \mathbf{x}_t\|^2 \\ &= \min_{\mathbf{v} \in S^*} f(\mathbf{x}_t) + \eta_t(\mathbf{v} - \mathbf{x}_t)^\top \nabla f(\mathbf{x}_t) + \frac{\beta\eta_t^2}{2} (\|\mathbf{x}_t - \mathbf{x}^*\|^2 \\ &\quad + 2(\mathbf{x}_t - \mathbf{x}^*)^\top (\mathbf{x}^* - \mathbf{v}) + \|\mathbf{v} - \mathbf{x}^*\|^2) \\ &\leq \min_{\mathbf{v} \in S^*} f(\mathbf{x}_t) + \eta_t(\mathbf{v} - \mathbf{x}_t)^\top \nabla f(\mathbf{x}_t) \\ &\quad + \frac{\beta\eta_t^2}{2} (\text{dist}(\mathbf{x}_t, \mathcal{X}^*)^2 + 2(\mathbf{x}_t - \mathbf{x}^*)^\top (\mathbf{x}^* - \mathbf{v}) + D^{*2}) \\ &\stackrel{(a)}{\leq} f(\mathbf{x}_t) + \eta_t(\mathbf{x}^* - \mathbf{x}_t)^\top \nabla f(\mathbf{x}_t) \\ &\quad + \frac{\beta\eta_t^2}{2} (\text{dist}(\mathbf{x}_t, \mathcal{X}^*)^2 + 2(\mathbf{x}_t - \mathbf{x}^*)^\top (\mathbf{x}^* - \mathbf{x}^*) + D^{*2}) \\ &\leq f(\mathbf{x}_t) - \eta_t(f(\mathbf{x}_t) - f^*) + \frac{\beta\eta_t^2}{2} (\text{dist}(\mathbf{x}_t, \mathcal{X}^*)^2 + D^{*2}), \end{aligned}$$

where (a) holds since $\mathbf{x}^* \in \text{conv}(S^*)$ and since $\mathbf{v}^\top \nabla f(\mathbf{x}_t) + \beta\eta_t(\mathbf{x}_t - \mathbf{x}^*)^\top (\mathbf{x}^* - \mathbf{v})$ is linear in \mathbf{v} .

Subtracting f^* from both sides concludes the proof. \blacksquare

3.2. Error reduction for Algorithm 2

We now proceed to the main lemma in the proof of Theorems 5, 6 which will allow us to bound the improvement on each iteration of Algorithm 2.

Lemma 9 *Fix some iteration $t \geq 1$ of Algorithm 2 (when either Option 1 or 2 are used) and consider a convex decomposition of \mathbf{x}_t into vertices: $\mathbf{x}_t = \sum_{i=1}^k \lambda_i \mathbf{v}_i$, and fix some $\eta \in [0, 1]$. Suppose there exists $R \leq 1$ such that some $\mathbf{x}^* \in \mathcal{X}^*$ can be written as $\mathbf{x}^* = \sum_{i=1}^k (\lambda_i - \Delta_i^*) \mathbf{v}_i + \sum_{i=1}^k \Delta_i^* \mathbf{z}$ for $\Delta_i^* \in [0, \lambda_i]$, $\mathbf{z} \in \mathcal{F}^*$ and $\sum_{i=1}^k \Delta_i^* \leq R$. Then, using $\rho_t = \eta R$ one has that,*

$$f(\mathbf{x}_{t+1}) - f^* \leq (1 - \eta)(f(\mathbf{x}_t) - f^*) + \eta^2 \beta (2\|\mathbf{x}_t - \mathbf{x}^*\|^2 + 4R^2 D_{\mathcal{F}^*}^2).$$

This lemma indeed allows us to obtain linear rates since, as it was shown in [Garber and Hazan \(2016\)](#); [Garber \(2020\)](#), informally speaking, we can take $R = O(\|\mathbf{x}_t - \mathbf{x}^*\|) = O(\sqrt{f(\mathbf{x}_t) - f^*})$.

Proof First, note that we can assume w.l.o.g. that $\sum_{i=1}^k \Delta_i^* = R$ (see [Observation 1](#) in [Appendix C.1](#)).

Now let $\mathbf{p}_t = \mathbf{x}^* - R\mathbf{z} + R\mathbf{v}_{k+1}$ and $\mathbf{y}_{t+1} = (1 - \eta)\mathbf{x}_t + \eta\mathbf{p}_t$. Note that since $\mathbf{p}_t = \sum_{i=1}^k (\lambda_i - \Delta_i^*)\mathbf{v}_i + \sum_{i=1}^k \Delta_i^*\mathbf{v}_{k+1}$, we have that $\mathbf{p}_t \in \text{conv}(\mathbf{v}_1, \dots, \mathbf{v}_{k+1})$ and thus $\mathbf{y}_{t+1} \in \text{conv}(\mathbf{v}_1, \dots, \mathbf{v}_{k+1})$. Thus by the β -smoothness of $f(\cdot)$ and the optimality of \mathbf{x}_{t+1} (as defined in either line 7 or 9 of [Algorithm 2](#)), we have that,

$$\begin{aligned}
 f(\mathbf{x}_{t+1}) &\leq f(\mathbf{x}_t) + (\mathbf{y}_{t+1} - \mathbf{x}_t)^\top \nabla f(\mathbf{x}_t) + \frac{\beta}{2} \|\mathbf{y}_{t+1} - \mathbf{x}_t\|^2 \\
 &= f(\mathbf{x}_t) + \eta(\mathbf{x}^* - R\mathbf{z} + R\mathbf{v}_{k+1} - \mathbf{x}_t)^\top \nabla f(\mathbf{x}_t) + \frac{\eta^2 \beta}{2} \|\mathbf{x}^* - R\mathbf{z} + R\mathbf{v}_{k+1} - \mathbf{x}_t\|^2 \\
 &\leq f(\mathbf{x}_t) + \eta(\mathbf{x}^* - \mathbf{x}_t)^\top \nabla f(\mathbf{x}_t) + \eta R(\mathbf{v}_{k+1} - \mathbf{z})^\top \nabla f(\mathbf{x}_t) \\
 &\quad + \eta^2 \beta (\|\mathbf{x}^* - R\mathbf{z} - (1 - R)\mathbf{x}_t\|^2 + R^2 \|\mathbf{v}_{k+1} - \mathbf{x}_t\|^2) \\
 &\leq f(\mathbf{x}_t) + \eta(\mathbf{x}^* - \mathbf{x}_t)^\top \nabla f(\mathbf{x}_t) \\
 &\quad + \eta^2 \beta (\|\mathbf{x}^* - R\mathbf{z} - (1 - R)\mathbf{x}_t\|^2 + R^2 \|\mathbf{w}^* - \mathbf{x}_t\|^2)
 \end{aligned} \tag{7}$$

where in the last inequality we let \mathbf{w}^* be a vertex in \mathcal{F}^* such that $\mathbf{w}^{*\top} \nabla f(\mathbf{x}_t) \leq \mathbf{z}^\top \nabla f(\mathbf{x}_t)$ and we use the fact that since $\rho_t = \eta R$, we have that $\mathbf{v}_{k+1} \in \arg \min_{\mathbf{u} \in \mathcal{V}} \mathbf{u}^\top \nabla f(\mathbf{x}_t) + \beta \eta R \|\mathbf{u} - \mathbf{x}_t\|^2$. Note that

$$\|\mathbf{w}^* - \mathbf{x}_t\|^2 \leq 2\|\mathbf{w}^* - \mathbf{x}^*\|^2 + 2\|\mathbf{x}_t - \mathbf{x}^*\|^2 \leq 2\|\mathbf{x}_t - \mathbf{x}^*\|^2 + 2D_{\mathcal{F}^*}^2. \tag{8}$$

Also,

$$\begin{aligned}
 \|\mathbf{x}^* - R\mathbf{z} - (1 - R)\mathbf{x}_t\|^2 &= \|(1 - R)\mathbf{x}^* - (1 - R)\mathbf{x}_t + R\mathbf{x}^* - R\mathbf{z}\|^2 \\
 &\leq 2(1 - R)^2 \|\mathbf{x}_t - \mathbf{x}^*\|^2 + 2R^2 D_{\mathcal{F}^*}^2.
 \end{aligned} \tag{9}$$

Plugging-in (8),(9) into (7), we have

$$\begin{aligned}
 f(\mathbf{x}_{t+1}) &\leq f(\mathbf{x}_t) + \eta(\mathbf{x}^* - \mathbf{x}_t)^\top \nabla f(\mathbf{x}_t) \\
 &\quad + \eta^2 \beta (2R^2 \|\mathbf{x}_t - \mathbf{x}^*\|^2 + 2R^2 D_{\mathcal{F}^*}^2 + 2(1 - R)^2 \|\mathbf{x}_t - \mathbf{x}^*\|^2 + 2R^2 D_{\mathcal{F}^*}^2) \\
 &\leq f(\mathbf{x}_t) + \eta(\mathbf{x}^* - \mathbf{x}_t)^\top \nabla f(\mathbf{x}_t) + \eta^2 \beta (2\|\mathbf{x}_t - \mathbf{x}^*\|^2 + 4R^2 D_{\mathcal{F}^*}^2),
 \end{aligned}$$

where we have used the fact that $\max_{R \in [0,1]} R^2 + (1 - R)^2 = 1$.

Finally, using the gradient inequality and subtracting f^* from both sides we get the proof. \blacksquare

4. Experiments

In this section we present numerical evidence that demonstrate the benefits of our NEP oracle-based algorithms. Due to lack of space, additional results are deferred to [Appendix E](#).

We conducted two experiments. In the first experiment we consider minimizing a random least-squares objective over the unit hypercube $[0, 1]^d$. In the second experiment we consider the task of

video co-localization taken from [Lacoste-Julien and Jaggi \(2015\)](#) which takes the form of minimizing a convex quadratic objective over the flow polytope. Note that in both experiments the feasible set is a convex and compact polytope. In both experiments we compare the performances of the original Frank-Wolfe algorithm (FW), the Frank-Wolfe with away-steps (AFW) variant [Lacoste-Julien and Jaggi \(2015\)](#), the fully-corrective Frank-Wolfe variant (FC) [Jaggi \(2013\)](#)⁸, our Algorithm 1 (NEP FW), and our Algorithm 2 with the fully-corrective option (NEP FC). For the video co-localization experiment we also included the pairwise Frank-Wolfe variant (PFW) [Lacoste-Julien and Jaggi \(2015\)](#) and the DICG Frank-Wolfe variant (with line-search) [Garber and Meshi \(2016\)](#).

In both experiments, when implementing the fully corrective variants (FC, NEP FC), we used a constant number of FISTA [Beck and Teboulle \(2009\)](#) iterations in order to compute the next iterate (i.e., finding an approximate solution to the problem in line 9 of Algorithm 2). For our NEP oracle-based algorithms which require the smoothness parameter β , we set it precisely according to the data (i.e., largest eigenvalue of the Hessian, which is fixed since the objectives are quadratic).

4.1. Hypercube-constrained least-squares

We consider the problem $\min_{\mathbf{x} \in [0,1]^d} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$, where we take \mathbf{A} to be a $m \times d$ matrix, $m = 175$, $d = 200$, with standard Gaussian entries and we set $\mathbf{b} = \mathbf{A}\mathbf{x}^*$, where \mathbf{x}^* is constructed by first choosing a random vertex of the hypercube and then changing its first 5 entries to 0.5. Thus, \mathbf{x}^* , which is also an optimal solution, lies on a face of dimension 5 of the hypercube. The initialization point for all algorithms is taken to be $\mathbf{0}$. For both standard Frank-Wolfe and Algorithm 1 we used the theoretical step-size $\frac{2}{t+1}$ (an alternative is to use line-search but for both variants it seems to give inferior results on this problem). For Algorithm 1 we skipped line 5 since it had no observable impact on performance. For Algorithm 2, on each iteration t we took $\rho_t \in \{2^{a/4} \rho_{t-1}\}_{a=-4}^4$ which achieves the lowest function value, where initially we set $\rho_0 = 0.5$. For the FC and NEP FC variants, on each iteration t we used 50 iterations of FISTA to compute the next iterate \mathbf{x}_{t+1} . Also, the smoothness constant L_t of the FISTA objective was chosen to be fixed throughout all iterations and its was empirically tuned, resulting in $2 \cdot 10^4$ for Algorithm 2 and $5 \cdot 10^4$ for FC.

Figure 1 shows the results averaged over 50 i.i.d. runs (where in each run we sample a fresh matrix \mathbf{A} and an optimal solution \mathbf{x}^*). As can be seen, NEP FC significantly outperforms all other algorithms both with respect to the number of iterations and runtime. Also, it can be seen that the simple addition of the NEP oracle in the NEP FW method leads to substantial improvement in performance compared to the standard Frank-Wolfe method. Moreover, with respect to runtime, NEP FW outperforms all other methods except for NEP FC.

4.2. Video co-localization

For our second experiment we use a formulation of the *video co-localization* task as a convex quadratic problem over the flow polytope (which is a 0–1 polytope), a formulation that was originally proposed in [Joulin et al. \(2014\)](#). We used the same dataset and initialization point used in [Lacoste-Julien and Jaggi \(2015\)](#) and [Garber and Meshi \(2016\)](#). The dimension of the problem is $d = 660$ and the optimal solution has 66 non-zero coordinates and no coordinate is equal to 1, which implies that the optimal face is indeed low-dimensional.

8. Fully-corrective Frank-Wolfe is equivalent to using the fully-corrective option in Algorithm 2 with $\rho_t = 0$ i.e. with a linear optimization oracle instead of the NEP oracle.

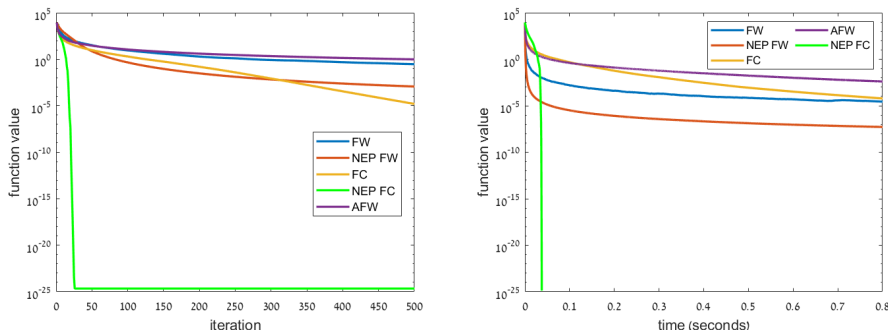


Figure 1: Comparison of Frank-Wolfe variants on the hypercube-constrained least-squares problem. The results are the average of 50 i.i.d. runs.

As opposed to the previous experiment, here for both the standard Frank-Wolfe method and Algorithm 1 we used line-search to set the step-size since for both it gives better results than the fixed $\frac{2}{t+1}$ step-size. For Algorithm 1 we use the theoretical constant $\eta_t = \frac{2}{t+1}$ for the regularization weight when calling the NEP oracle. For Algorithm 2 we used $\rho_t = (1/\sqrt{2})^{t+1}$. For both FC and NEP FC variants, on each iteration t we used 10 iterations of FISTA to compute \mathbf{x}_{t+1} , where as in the previous experiment, the FISTA smoothness parameter was fixed throughout all iterations and tuned empirically, resulting in a value of 0.25 for both variants.

Since the optimal value of the objective is not known we find it approximately using 1000 iteration of DICG Garber and Meshi (2016) (which results in a duality gap of 10^{-15}).

The results are given in Figure 2. As it can be seen, NEP FC outperforms all other algorithms, both with respect to the number of iterations and running time. Although it may seem that the difference between NEP-FC and FC is not significant, the ratio between the time it takes FC to reach an approximation error of 10^{-12} and the time it takes NEP FC to reach the same error is 1.21. Furthermore, it can be seen that the simple addition of the NEP oracle in the NEP FW method leads to substantial improvement in performance compared to the standard Frank-Wolfe method, and that with respect to running time, NEP FW outperforms the linearly-converging variants AFW and PFW.

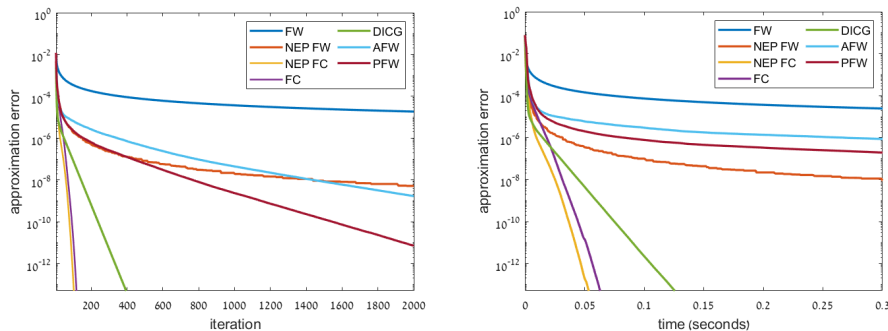


Figure 2: Comparison of Frank-Wolfe variants on the video co-localization problem. The times shown are the averages of 200 runs.

Acknowledgments

This research was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 1108/18).

References

- Zeyuan Allen-Zhu, Elad Hazan, Wei Hu, and Yuanzhi Li. Linear convergence of a frank-wolfe type algorithm over trace-norm balls. In *Advances in Neural Information Processing Systems*, pages 6191–6200, 2017.
- Amir Beck. *First-order methods in optimization*. SIAM, 2017.
- Amir Beck and Shimrit Shtern. Linearly convergent away-step conditional gradient for non-strongly convex functions. *Mathematical Programming*, 164(1-2):1–27, 2017.
- Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
- Jelena Diakonikolas, Alejandro Carderera, and Sebastian Pokutta. Locally accelerated conditional gradients. In *International Conference on Artificial Intelligence and Statistics*, pages 1737–1747. PMLR, 2020.
- Lijun Ding, Jicong Fan, and Madeleine Udell. k fw: A frank-wolfe style algorithm with stronger subproblem oracles. *arXiv preprint arXiv:2006.16142*, 2020a.
- Lijun Ding, Yingjie Fei, Qiantong Xu, and Chengrun Yang. Spectral frank-wolfe algorithm: Strict complementarity and linear convergence. *arXiv preprint arXiv:2006.01719*, 2020b.
- Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.
- Robert M Freund and Paul Grigas. New analysis and results for the frank-wolfe method. *Mathematical Programming*, 155(1-2):199–230, 2016.
- Dan Garber. Faster projection-free convex optimization over the spectrahedron. In *Advances in Neural Information Processing Systems*, pages 874–882, 2016.
- Dan Garber. Logarithmic regret for online gradient descent beyond strong convexity. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 295–303. PMLR, 16–18 Apr 2019a. URL <http://proceedings.mlr.press/v89/garber19b.html>.
- Dan Garber. Linear convergence of frank-wolfe for rank-one matrix recovery without strong convexity. *arXiv preprint arXiv:1912.01467*, 2019b.
- Dan Garber. Revisiting frank-wolfe for polytopes: Strict complementarity and sparsity. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

- Dan Garber and Elad Hazan. Playing non-linear games with linear oracles. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 2013.
- Dan Garber and Elad 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.
- Dan Garber and Ofer Meshi. Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1001–1009, 2016.
- Jacques GuéLat and Patrice Marcotte. Some comments on Wolfe’s ‘away step’. *Mathematical Programming*, 35(1), 1986.
- Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271, 2016.
- Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning, ICML*, 2013.
- Martin Jaggi and Marek Sulovský. A simple algorithm for nuclear norm regularized problems. In *ICML*, 2010.
- Armand Joulin, Kevin Tang, and Li Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *European Conference on Computer Vision*, pages 253–268. Springer, 2014.
- Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in neural information processing systems*, pages 496–504, 2015.
- Guanghui Lan. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550*, 2013.
- Guanghui Lan and Yi Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.
- Evgeny S Levitin and Boris T Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6:1–50, 1966.
- Athanasios Migdalas. A regularization of the frank—wolfe method and unification of certain non-linear programming methods. *Mathematical Programming*, 65(1-3):331–345, 1994.
- Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.
- Javier Pena and Daniel Rodríguez. Polytope conditioning and linear convergence of the frank–wolfe algorithm. *Mathematics of Operations Research*, 44(1):1–18, 2019.
- Javier Pena, Daniel Rodríguez, and Negar Soheili. On the von neumann and frank–wolfe algorithms with away steps. *SIAM Journal on Optimization*, 26(1):499–512, 2016.
- Philip Wolfe. *Integer and nonlinear programming*. North-Holland, 1970.

Appendix A. Proof of Theorem 2

For clarity, we first restate the theorem and then prove it.

Theorem 2 Using Algorithm 1 with step-size $\eta_t = \frac{2}{t+1}$ we have

$$\forall t \geq 2: \quad f(\mathbf{x}_t) - f^* \leq \frac{2\beta(D^{*2} + D_L^2)}{t+1},$$

where D_L is the diameter of the initial level set (see Footnote 2).

Moreover, if $f(\cdot)$ has the quadratic growth property over \mathcal{K} with parameter $\alpha > 0$, then

$$\forall t \geq 2: \quad f(\mathbf{x}_t) - f^* \leq \frac{2\beta D^{*2}}{t+1} + \frac{8\beta^2(D^{*2} + \min\{2\alpha^{-1}(f(\mathbf{x}_1) - f^*), D_L^2\}) \log(t)}{t^2}.$$

Proof Using Lemma 8 with our choice of step size η_t , we have that for all $t \geq 1$,

$$f(\mathbf{x}_{t+1}) - f^* \leq \left(1 - \frac{2}{t+1}\right)(f(\mathbf{x}_t) - f^*) + \frac{2\beta D^{*2}}{(t+1)^2} + \frac{2\beta \text{dist}(\mathbf{x}_t, \mathcal{X}^*)^2}{(t+1)^2}.$$

Thus, from Lemma 10 we have that for any $t \geq 2$,

$$\begin{aligned} f(\mathbf{x}_t) - f^* &\leq \frac{1}{t^2} \sum_{k=1}^{t-1} 2\beta(D^{*2} + \text{dist}(\mathbf{x}_k, \mathcal{X}^*)^2) \\ &\leq \frac{1}{t-1} \sum_{k=1}^{t-1} 2\beta D^{*2} + \frac{1}{t^2} \sum_{k=1}^{t-1} 2\beta \text{dist}(\mathbf{x}_k, \mathcal{X}^*)^2 \\ &= \frac{2\beta D^{*2}}{t+1} + \frac{1}{t^2} \sum_{k=1}^{t-1} 2\beta \text{dist}(\mathbf{x}_k, \mathcal{X}^*)^2 \end{aligned} \quad (10)$$

$$\leq \frac{2\beta D^{*2}}{t+1} + \frac{1}{t-1} \sum_{k=1}^{t-1} 2\beta \text{dist}(\mathbf{x}_k, \mathcal{X}^*)^2. \quad (11)$$

Since Algorithm 1 is a decent method (i.e., the function value never increases from one iteration to the next), and so all iterates as well as the optimal set \mathcal{X}^* are contained within the initial level set L , for any $t \geq 1$ we can bound $\text{dist}(\mathbf{x}_t, \mathcal{X}^*) \leq D_L$.

Thus, using (11) we have that,

$$\forall t \geq 2: \quad f(\mathbf{x}_t) - f^* \leq \frac{2\beta D^{*2}}{t+1} + \frac{2\beta D_L^2}{t+1}.$$

If we additionally assume quadratic growth, we can use again the fact that Algorithm 1 is a decent method, in order bound $\text{dist}(\mathbf{x}_t, \mathcal{X}^*)^2 \leq \frac{2}{\alpha}(f(\mathbf{x}_1) - f^*)$ which results in the bound

$$\forall t \geq 2: \quad f(\mathbf{x}_t) - f^* \leq \frac{2\beta D^{*2}}{t+1} + \frac{2\beta \min\{2\alpha^{-1}(f(\mathbf{x}_1) - f^*), D_L^2\}}{t+1}. \quad (12)$$

Thus, denoting $M = 2\beta(D^{*2} + \min\{\frac{2}{\alpha}(f(\mathbf{x}_1) - f^*), D_L^2\})$, using again the quadratic growth of $f(\cdot)$ and (10) we have that for any $t \geq 2$,

$$\begin{aligned} f(\mathbf{x}_t) - f^* &\leq \frac{2\beta D^{*2}}{t+1} + \frac{4\beta}{\alpha t^2} \sum_{k=1}^{t-1} (f(\mathbf{x}_k) - f^*) \stackrel{(a)}{\leq} \frac{2\beta D^{*2}}{t+1} + \frac{4\beta}{\alpha t^2} \sum_{k=1}^{t-1} \frac{M}{k+1} \\ &= \frac{2\beta D^{*2}}{t+1} + \frac{4\beta}{\alpha} M \left(\sum_{k=1}^t \frac{1}{k} - 1 \right) \leq \frac{2\beta D^{*2}}{t+1} + \frac{4\beta}{\alpha} \frac{M \log(t)}{t^2}, \end{aligned} \quad (13)$$

where (a) follows from the definition of M and the bound in (12).

Thus, plugging-in the value of M in (13) we indeed have that,

$$\forall t \geq 2: \quad f(\mathbf{x}_t) - f^* \leq \frac{2\beta D^{*2}}{t+1} + \frac{8\beta^2 (D^{*2} + \min\{2\alpha^{-1}(f(\mathbf{x}_1) - f^*), D_L^2\}) \log(t)}{t^2}.$$

■

Lemma 10 *Let $\{a_t, b_t\}_{t=1}^{\infty}$ be non-negative scalars such that,*

$$\forall t \geq 1: \quad a_{t+1} \leq \left(1 - \frac{2}{t+1}\right) a_t + \frac{b_t}{(t+1)^2}.$$

Then, we have that for any $t \geq 2$,

$$a_t \leq \frac{1}{t^2} \sum_{k=1}^{t-1} b_k.$$

In particular, when $\{b_t\}_{t=1}^{\infty}$ is upper-bounded by $M > 0$ we have that,

$$\forall t \geq 2: \quad a_t \leq \frac{M}{t+1}.$$

Proof First we define a sequence $\{\hat{a}_t\}_{t=1}^{\infty}$ such that $\hat{a}_1 = a_1$ and $\forall t \geq 1: \hat{a}_{t+1} = \left(1 - \frac{2}{t+1}\right) \hat{a}_t + \frac{b_t}{(t+1)^2}$. Note that $\hat{a}_2 = \frac{b_1}{4}$ and that for any $t \geq 1$, $a_t \leq \hat{a}_t$.

Thus, since $\{\hat{a}_t\}_{t=2}^{\infty}$ is of the form $\hat{a}_{t+1} = c_t \hat{a}_t + \bar{b}_t$ (a first-order non-homogeneous recurrence relation) we have that for any $t \geq 2$,

$$\hat{a}_t = \left(\prod_{i=2}^{t-1} c_i \right) \left(\hat{a}_2 + \sum_{k=2}^{t-1} \frac{\bar{b}_k}{\prod_{i=2}^k c_i} \right).$$

Thus, noting that for any $k \geq 2$,

$$\prod_{i=2}^k \left(1 - \frac{2}{i+1}\right) = \prod_{i=2}^k \frac{i-1}{i+1} = \frac{2}{k(k+1)},$$

we have that for $t \geq 3$,

$$\begin{aligned} a_t &\leq \hat{a}_t = \frac{2}{t(t-1)} \left(\hat{a}_2 + \sum_{k=2}^{t-1} \frac{k(k+1)}{2} \cdot \frac{b_k}{(k+1)^2} \right) \\ &\stackrel{(a)}{=} \frac{2}{t(t-1)} \sum_{k=1}^{t-1} \frac{k(k+1)}{2} \cdot \frac{b_k}{(k+1)^2} = \frac{1}{t(t-1)} \sum_{k=1}^{t-1} \frac{kb_k}{k+1} \\ &\leq \frac{1}{t(t-1)} \cdot \frac{t-1}{t} \sum_{k=1}^{t-1} b_k = \frac{1}{t^2} \sum_{k=1}^{t-1} b_k, \end{aligned}$$

where (a) holds since $\hat{a}_2 = \frac{b_1}{4}$ (note that for that reason the bound also holds for $t = 2$).

Finally, if $\{b_t\}_{t=1}^\infty$ is upper-bounded by some $M > 0$, we have that for any $t \geq 2$,

$$a_t \leq \frac{1}{t^2} \sum_{k=1}^{t-1} b_k \leq \frac{(t-1)M}{(t+1)(t-1)} = \frac{M}{t+1}.$$

■

Appendix B. Proof of Theorem 4

For clarity, we first restate the theorem and then prove it.

Theorem 4 *Let $\mathcal{K} = [0, 1]^d$, and fix a positive integer $m < d$. Let $\mathbf{x}^* = \frac{1}{2} \sum_{i=1}^m \mathbf{e}_i$, i.e., \mathbf{x}^* has $\frac{1}{2}$ for the first m coordinates and 0 for the rest. Now consider the minimization of the function $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{x} - \mathbf{x}^*\|^2$ over \mathcal{K} starting at $\mathbf{x}_1 = \mathbf{e}_{m+1}$. Then, for any Frank-Wolfe-type method there exists a sequence of answers returned by the linear optimization oracle such that for any $t \leq \lfloor \sqrt{d-m-1} \rfloor$, the t th iterate of the algorithm \mathbf{x}_t satisfies $f(\mathbf{x}_t) - f^* \geq \frac{1}{4}$.*

Proof Clearly, the unique optimal solution is \mathbf{x}^* and $f^* = f(\mathbf{x}^*) = 0$. Let $k = \lfloor \sqrt{d-m-1} \rfloor$ and let S_0, S_1, \dots, S_k be a partition of the last $d-m-1$ coordinates (i.e., of the set $\{m+2 \leq i \leq d : i \in \mathbb{N}\}$) such that each $S_i, i = 1, \dots, k$ contains exactly k coordinates. Consider now the iterates of some Frank-Wolfe-type method. Observe that for any $t \geq 1$, since the last $d-m$ coordinates of $\nabla f(\mathbf{x}_t) = \mathbf{x}_t - \mathbf{x}^*$ are the same as those of \mathbf{x}_t , the last $d-m$ coordinates of a valid answer returned by the linear optimization oracle can contain 0 in coordinates in which \mathbf{x}_t is non-zero, and either 0 or 1 in coordinates in which \mathbf{x}_t is 0. Thus, a valid sequence of answers returned by the linear optimization oracle on iterations $1 \leq t \leq k$ may set on each iteration $t \in \{1, \dots, k\}$ the oracle's output \mathbf{v}_t to contain 1 in the coordinates in S_t and 0 in the rest of the last $d-m$ coordinates (i.e., the coordinates in $(\bigcup_{i=0}^k S_i \setminus S_t) \cup \{m+1\}$).

For any vector $\mathbf{x} \in \mathbb{R}^d$ we let $\mathbf{x}^{(d-m)}$ denote the restriction of \mathbf{x} to the last $d-m$ coordinates. Now, fix some $t \leq k+1$. Since $\mathbf{x}_t \in \text{conv}(\mathbf{x}_1, \mathbf{v}_1, \dots, \mathbf{v}_k)$, there exist some $\mathbf{w} \in \text{conv}(\mathbf{v}_1, \dots, \mathbf{v}_k)$ and $\gamma \in [0, 1]$ such that $\mathbf{x}_t = (1-\gamma)\mathbf{x}_1 + \gamma\mathbf{w}$. Note that since $\mathbf{w}^{(d-m)}$ is a convex combination of k orthogonal vectors $(\mathbf{v}_1^{(d-m)}, \dots, \mathbf{v}_k^{(d-m)})$, each having Euclidean norm \sqrt{k} , we have that $\|\mathbf{w}^{(d-m)}\|^2 \geq 1$. Thus, using the fact that $\mathbf{x}^{*(d-m)} = \mathbf{0}$ and that $\mathbf{w}^{(d-m)}$ is orthogonal to $\mathbf{x}_1^{(d-m)}$,

we have that,

$$\begin{aligned} f(\mathbf{x}_t) - f^* &= \frac{1}{2} \|\mathbf{x}_t - \mathbf{x}^*\|^2 \geq \frac{1}{2} \|\mathbf{x}_t^{(d-m)}\|^2 = \frac{1}{2} \|(1-\gamma)\mathbf{x}_1^{(d-m)} + \gamma\mathbf{w}^{(d-m)}\|^2 \\ &= \frac{1}{2}(1-\gamma)^2 \|\mathbf{x}_1^{(d-m)}\|^2 + \frac{1}{2}\gamma^2 \|\mathbf{w}^{(d-m)}\|^2 \geq \frac{1}{2}(1-\gamma)^2 + \frac{1}{2}\gamma^2 \geq \frac{1}{4}, \end{aligned}$$

where the last inequality holds since $\min_{\gamma \in [0,1]} (1-\gamma)^2 + \gamma^2 = \frac{1}{2}$.

Thus, we indeed have that for any $t \leq k = \lfloor \sqrt{d-m-1} \rfloor$, $f(\mathbf{x}_t) - f^* \geq \frac{1}{4}$. ■

Appendix C. Results and proofs missing from Section 2.2

In Section C.1 we prove Theorem 5, in Section C.2 we prove that the linear convergence rate in Theorem 5 also holds if instead of using a predefined sequence $\{\rho_t\}_{t \geq 1}$, we use an adaptive-step size strategy, and in Section C.3 we prove Theorem 6.

C.1. Proof of Theorem 5

We first prove the following technical observation which was used in the proof of Lemma 9 and then prove the theorem.

Observation 1 *Suppose that \mathcal{K} is a convex and compact polytope and let $\mathbf{x} \in \mathcal{K}$ which is given by a convex combination of vertices $\mathbf{x} = \sum_{i=1}^k \lambda_i \mathbf{v}_i$. Suppose there exists $R \leq 1$ such that some $\mathbf{x}^* \in \mathcal{X}^*$ can be written as $\mathbf{x}^* = \sum_{i=1}^k (\lambda_i - \Delta_i^*) \mathbf{v}_i + \sum_{i=1}^k \Delta_i^* \mathbf{z}$ for $\Delta_i^* \in [0, \lambda_i]$, $\mathbf{z} \in \mathcal{F}^*$ and $\sum_{i=1}^k \Delta_i^* \leq R$. Then, \mathbf{x}^* can also be written as, $\mathbf{x}^* = \sum_{i=1}^k (\lambda_i - \bar{\Delta}_i) \mathbf{v}_i + \sum_{i=1}^k \bar{\Delta}_i \bar{\mathbf{z}}$ where, $\bar{\Delta}_i \in [0, \lambda_i]$, $\bar{\mathbf{z}} \in \mathcal{F}^*$ and $\sum_{i=1}^k \bar{\Delta}_i = R$.*

Proof Denote $R' = \sum_{i=1}^k \Delta_i^*$ and assume $R' < R$. Let $\gamma = \frac{R-R'}{1-R'}$ and note that since $R' < R \leq 1$, $\gamma \in [0, 1]$. We will show that $\bar{\Delta}_i = (1-\gamma)\Delta_i^* + \gamma\lambda_i$ and $\bar{\mathbf{z}} = \frac{(1-\gamma)R'}{R}\mathbf{z} + \frac{\gamma}{R}\mathbf{x}^*$ satisfy the required conditions.

Indeed, since $\gamma \in [0, 1]$, we have that $\bar{\Delta}_i \in [0, \lambda_i]$, and since

$$(1-\gamma)R' + \gamma = \frac{R'(1-R)}{1-R'} + \frac{R-R'}{1-R'} = R, \quad (14)$$

we have that, $\bar{\mathbf{z}} = \frac{(1-\gamma)R'}{R}\mathbf{z} + \frac{\gamma}{R}\mathbf{x}^*$ is a convex combination of points in \mathcal{F}^* and thus in \mathcal{F}^* itself.

Moreover, using (14) we have that,

$$\sum_{i=1}^k \bar{\Delta}_i = (1-\gamma) \sum_{i=1}^k \Delta_i^* + \gamma \sum_{i=1}^k \lambda_i = (1-\gamma)R' + \gamma = R.$$

Finally, we indeed have that,

$$\begin{aligned} \mathbf{x}^* &= (1-\gamma)\mathbf{x}^* + \gamma\mathbf{x}^* = \sum_{i=1}^k (1-\gamma)(\lambda_i - \Delta_i^*)\mathbf{v}_i + (1-\gamma)R'\mathbf{z} + \gamma\mathbf{x}^* \\ &= \sum_{i=1}^k (\lambda_i - \bar{\Delta}_i)\mathbf{v}_i + R\bar{\mathbf{z}}, \end{aligned}$$

completing the proof. \blacksquare

Before continuing to the proof of Theorem 5 we first restate it.

Theorem 5 *Suppose that \mathcal{K} is a convex and compact polytope and quadratic growth holds with parameter $\alpha > 0$. Let $C \geq f(\mathbf{x}_1) - f^*$ and $M \geq \max\{\frac{\beta}{\alpha}(4 + 8d\mu^2 D_{\mathcal{F}^*}^2), \frac{1}{2}\}$, where $\mu = \frac{\psi}{\xi}$. Using Algorithm 2 with parameter $\rho_t = \frac{\min\{\sqrt{\frac{2Cd\mu^2}{\alpha} \exp(-\frac{1}{4M}(t-1))}, 1\}}{2M}$ for all $t \geq 1$ one has,*

$$\forall t \geq 1 : \quad f(\mathbf{x}_t) - f^* \leq C \exp\left(-\frac{t-1}{4M}\right),$$

Proof The proof is by induction on t . For $t = 1$ the bound holds by the definition of the constant C .

Now suppose the bound holds for some $t \geq 1$. We will show that it holds for $t + 1$. Let $\mathbf{x}_t^* \in \mathcal{X}^*$ be the closest optimal solution to \mathbf{x}_t and let $\sum_{i=1}^k \lambda_i \mathbf{v}_i$ be the decomposition of \mathbf{x}_t used in Algorithm 2.

By Lemma 5.5 from Garber and Hazan (2016) there exist $\Delta_i^* \in [0, \lambda_i]$, $i = 1, \dots, k$ and $\mathbf{z} \in \mathcal{F}^*$, such that $\mathbf{x}_t^* = \sum_{i=1}^k (\lambda_i - \Delta_i^*) \mathbf{v}_i + \sum_{i=1}^k \Delta_i^* \mathbf{z}$, and $\sum_{i=1}^k \Delta_i^* \leq \min\{\sqrt{d}\mu \|\mathbf{x}_t - \mathbf{x}_t^*\|, 1\}$, which by the quadratic growth of $f(\cdot)$ together with the induction assumption, implies that $\sum_{i=1}^k \Delta_i^* \leq \min\{\sqrt{\frac{2Cd\mu^2}{\alpha} \exp(-\frac{1}{4M}(t-1))}, 1\}$, where M is as defined in the theorem.

Therefore, taking $R = \min\{\sqrt{\frac{2Cd\mu^2}{\alpha} \exp(-\frac{1}{4M}(t-1))}, 1\}$ and $\eta = \frac{1}{2M}$, we have that $\rho_t = \eta R$ and $\sum_{i=1}^k \Delta_i^* \leq R$ and thus, we can use Lemma 9 which implies that,

$$\begin{aligned} f(\mathbf{x}_{t+1}) - f^* &\leq (1 - \eta)(f(\mathbf{x}_t) - f^*) + \eta^2 \beta (2\|\mathbf{x}_t - \mathbf{x}_t^*\|^2 + 4R^2 D_{\mathcal{F}^*}^2) \\ &\stackrel{(a)}{\leq} (1 - \eta) C e^{-\frac{t-1}{4M}} + \eta^2 \beta \left(\frac{4C}{\alpha} e^{-\frac{t-1}{4M}} + \frac{8Cd\mu^2 D_{\mathcal{F}^*}^2}{\alpha} e^{-\frac{t-1}{4M}} \right) \\ &\stackrel{(b)}{\leq} (1 - \eta) C e^{-\frac{1}{4M}(t-1)} + M\eta^2 C e^{-\frac{1}{4M}(t-1)} \\ &\stackrel{(c)}{=} \left(1 - \frac{1}{4M}\right) C e^{-\frac{1}{4M}(t-1)} \stackrel{(d)}{\leq} C e^{-\frac{1}{4M}t}, \end{aligned}$$

where (a) holds by the induction assumption and the quadratic growth of $f(\cdot)$, (b) holds due to the definition of M , (c) holds since $\eta = \frac{1}{2M}$, and (d) holds since $1 - x \leq e^{-x}$. \blacksquare

C.2. Linear convergence with adaptive step-sizes

We now prove that, in principle, the linear rate of Theorem 5 can be achieved with an adaptive choice of the parameter ρ_t , instead of the predefined value listed in Theorem 5. Theorem 11 demonstrates that it suffices to do a log-scale search over ρ_t , i.e., check values $\rho_t = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots$, and take the one which leads to the largest decrease in function value. Note that according to the theorem and since \mathcal{K} is compact, if the target accuracy we are looking to obtain is some $\epsilon > 0$, we need not consider values of ρ_t below some $O(\epsilon)$. Thus, the overall number of search steps will be logarithmic in $1/\epsilon$, and the overall increase in complexity due to the use of such adaptive step-sizes will be an $O(\log 1/\epsilon)$ factor.

Theorem 11 *Suppose that all of the assumptions of Theorem 5 hold and denote $h_1 = f(\mathbf{x}_1) - f^*$ and $M^* = \max\{\frac{\beta}{\alpha}(4 + 8d\mu^2 D_{\mathcal{F}^*}^2), \frac{1}{2}\}$. Consider running Algorithm 2 and for all $t \geq 1$ define $\rho_t^* = \frac{\min\{\sqrt{d}\mu \text{dist}(\mathbf{x}_t, \mathcal{X}^*), 1\}}{2M^*}$. Suppose that for all $t \geq 1$ we use some value $\rho_t \in [\frac{\rho_t^*}{2}, \rho_t^*]$. Then, with these parameters one has,*

$$\forall t \geq 1 : f(\mathbf{x}_t) - f^* \leq h_1 \exp\left(-\frac{3(t-1)}{16M^*}\right).$$

Proof Fix some iteration $t \geq 1$. Using the same notation and arguments as in the proof of Theorem 5, by Lemma 5.5 from Garber and Hazan (2016) we can take $R = \min\{\sqrt{d}\mu \|\mathbf{x}_t - \mathbf{x}_t^*\|, 1\}$ (note that $\|\mathbf{x}_t - \mathbf{x}_t^*\| = \text{dist}(\mathbf{x}_t, \mathcal{X}^*)$ as \mathbf{x}_t^* denotes the closest optimal solution to \mathbf{x}_t). Thus, taking $\eta = \rho_t/R$ and noting that by the definition of ρ_t in the theorem, $\eta \in [\frac{1}{4M^*}, \frac{1}{2M^*}]$, by Lemma 9 we have that,

$$\begin{aligned} f(\mathbf{x}_{t+1}) - f^* &\leq (1 - \eta)(f(\mathbf{x}_t) - f^*) + \eta^2 \beta (2\|\mathbf{x}_t - \mathbf{x}_t^*\|^2 + 4R^2 D_{\mathcal{F}^*}^2) \\ &\stackrel{(a)}{\leq} (1 - \eta)(f(\mathbf{x}_t) - f^*) + \eta^2 \beta \left(\frac{4}{\alpha} + \frac{8d\mu^2 D_{\mathcal{F}^*}^2}{\alpha}\right) (f(\mathbf{x}_t) - f^*) \\ &\stackrel{(b)}{=} (1 - \eta + M^* \eta^2)(f(\mathbf{x}_t) - f^*) \stackrel{(c)}{\leq} \left(1 - \frac{3}{16M^*}\right) (f(\mathbf{x}_t) - f^*), \end{aligned}$$

where (a) follows from the quadratic growth of $f(\cdot)$ and the definition of R , (b) follows from the definition of M^* , and (c) holds since $\eta \in [\frac{1}{4M^*}, \frac{1}{2M^*}]$.

Using the fact that $1 - x \leq e^{-x}$, we have that,

$$f(\mathbf{x}_t) - f^* \leq h_1 \left(1 - \frac{3}{16M^*}\right)^{t-1} \leq h_1 \exp\left(-\frac{3(t-1)}{16M^*}\right). \quad \blacksquare$$

C.3. Proof of Theorem 6

The proof goes along the same lines as the proof of Theorem 5, but this time, since we assume δ -strict complementarity, we can use Lemma 2 from Garber (2020) instead of Lemma 5.5 from Garber and Hazan (2016) in order to upper-bound the amount of probability mass we need to move from the convex decomposition of the point \mathbf{x}_t in order to reach the closest optimal solution \mathbf{x}^* , yielding a dimension-independent linear convergence rate.

For clarity, before continuing to the proof of the theorem we first restate it.

Theorem 6 *Suppose that in addition to the assumptions of Theorem 5, Assumption 1 also holds with some parameter $\delta > 0$, and let $C \geq f(\mathbf{x}_1) - f^*$, $M_1 \geq \max\{\frac{4\beta}{\alpha} + 8\beta D_{\mathcal{F}^*}^2 \max\{2\kappa, \delta^{-1}\}, \frac{1}{2}\}$ and $M_2 \geq \{\frac{4\beta}{\alpha} + 16\beta\kappa D_{\mathcal{F}^*}^2, \frac{1}{2}\}$, where $\kappa = \frac{2\mu^2 \dim \mathcal{F}^*}{\alpha}$.*

Using Algorithm 2 with parameters $\rho_t = \frac{\min\{\sqrt{2 \max\{2\kappa, \delta^{-1}\} C \exp(-\frac{1}{4M_1}(t-1)), 1\}}{2M_1}$ for all $t \geq 1$, one has,

$$\forall t \geq 1 : f(\mathbf{x}_t) - f^* \leq C \exp\left(-\frac{t-1}{4M_1}\right). \quad (5)$$

Furthermore, in case that $2\kappa \leq \delta^{-1}$ and $\dim \mathcal{F}^* > 0$ (i.e. $\kappa > 0$), denoting $\tau \geq 4M_1 \log(\frac{C}{\delta^2\kappa}) + 1$ and for all $t \geq \tau$, using $\rho_t = \frac{\min\{2\delta\kappa \exp(-\frac{t-\tau}{8M_2}), 1\}}{2M_2}$ instead of the above value, one has,

$$\forall t \geq \tau: \quad f(\mathbf{x}_t) - f^* \leq \delta^2\kappa \exp\left(-\frac{t-\tau}{4M_2}\right). \quad (6)$$

Proof We first prove the rate in (5) by induction on t . For $t = 1$ the bound holds by the definition of the constant C .

Now, suppose the bound holds for some $t \geq 1$. We will show that it holds for $t + 1$. Let $\mathbf{x}_t^* \in \mathcal{X}^*$ be the closest optimal solution to $\mathbf{x}_t = \sum_{i=1}^k \lambda_i \mathbf{v}_i$ and denote $h_t = f(\mathbf{x}_t) - f^*$. Then, by Lemma 2 from Garber (2020), there exist $\Delta_i^* \in [0, \lambda_i]$, $i = 1, \dots, k$, and $\mathbf{z} \in \mathcal{F}^*$, such that $\mathbf{x}_t^* = \sum_{i=1}^k (\lambda_i - \Delta_i^*) \mathbf{v}_i + \sum_{i=1}^k \Delta_i^* \mathbf{z}$ and $\sum_{i=1}^k \Delta_i^* \leq \min\{1, \delta^{-1}h_t + \sqrt{\kappa h_t}\}$.

Since when $\delta^{-1}h_t > \sqrt{\kappa h_t}$ we have that $\sum_{i=1}^k \Delta_i^* \leq \min\{1, 2\delta^{-1}h_t\} \leq \min\{1, \delta^{-1/2}\sqrt{2h_t}\}$, and otherwise we have that $\sum_{i=1}^k \Delta_i^* \leq \min\{1, 2\sqrt{\kappa h_t}\}$, by taking the maximum of the two we have that $\sum_{i=1}^k \Delta_i^* \leq \min\{1, \max\{\sqrt{2\kappa}, \delta^{-1/2}\}\sqrt{2h_t}\}$. Thus, taking $\eta = \frac{1}{2M_1}$ and $R = \min\{1, \sqrt{2 \max\{2\kappa, \delta^{-1}\}} C \exp(-\frac{t-1}{4M_1})\}$, where M_1 is as defined in the theorem, we have that $\rho_t = \eta R$ and $\sum_{i=1}^k \Delta_i^* \leq R$, and therefore we can use Lemma 9 which implies that,

$$\begin{aligned} h_{t+1} &\leq (1-\eta)h_t + \eta^2\beta(2\|\mathbf{x}_t - \mathbf{x}_t^*\|^2 + 4R^2D_{\mathcal{F}^*}^2) \\ &\stackrel{(a)}{\leq} (1-\eta)Ce^{-\frac{1}{4M_1}(t-1)} + \eta^2\beta\left(\frac{4}{\alpha}Ce^{-\frac{1}{4M_1}(t-1)} + 8\max\{2\kappa, \delta^{-1}\}D_{\mathcal{F}^*}^2Ce^{-\frac{1}{4M_1}(t-1)}\right) \\ &\stackrel{(b)}{\leq} (1-\eta)Ce^{-\frac{1}{4M_1}(t-1)} + M_1\eta^2Ce^{-\frac{1}{4M_1}(t-1)} \\ &\stackrel{(c)}{=} \left(1 - \frac{1}{4M_1}\right)Ce^{-\frac{1}{4M_1}(t-1)} \stackrel{(d)}{\leq} e^{-\frac{1}{4M_1}t}, \end{aligned}$$

where (a) holds by the induction assumption and the quadratic growth of $f(\cdot)$, (b) holds due to the definition of M_1 , (c) holds since $\eta = \frac{1}{2M_1}$, and (d) holds since $1 - x \leq e^{-x}$.

Now we turn to prove the rate in (6). The proof goes along the same lines as the proof of (5), but now we have from (5) that for τ as defined in the theorem, it holds that $h_\tau \leq \delta^2\kappa$, and since when $h_t \leq \delta^2\kappa$ we have that $\sum_{i=1}^k \Delta_i^* \leq \min\{1, \delta^{-1}h_t + \sqrt{\kappa h_t}\} \leq \min\{1, 2\sqrt{\kappa h_t}\}$, we can replace C with $\delta^2\kappa$ and $\max\{2\kappa, \delta^{-1}\}$ with 2κ in the above arguments, and get the desired bound for all $t \geq \tau$. \blacksquare

Appendix D. Results and proofs missing from Section 2.3

Our NEP Oracle-based Stochastic Frank-Wolfe variant is given in Algorithm 3

In Section D.1 we prove a theorem on the convergence rate of Algorithm 3. Then, in Section D.2 we prove Theorem 7.

Throughout this section, for all $t \geq 1$ we denote $\bar{h}_t = \mathbb{E}[f(\mathbf{x}_t) - f^*]$.

Algorithm 3 Stochastic Frank-Wolfe with a Nearest Extreme Point Oracle

- 1: Input: a sequence of step-sizes $\{\eta_t\} \subset [0, 1]$, a sequence of mini-batch sizes $\{m_t\} \subset \mathbb{N}$
 - 2: $\mathbf{x}_1 \leftarrow$ some arbitrary point in \mathcal{V}
 - 3: **for** $t = 1 \dots$ **do**
 - 4: Compute $\tilde{\nabla}_t$ as the average of m_t iid unbiased estimators of $\nabla f(\mathbf{x}_t)$
 - 5: $\mathbf{v}_t \leftarrow \arg \min_{\mathbf{v} \in \mathcal{V}} \mathbf{v}^\top \tilde{\nabla}_t + \frac{\beta \eta_t}{2} \|\mathbf{x}_t - \mathbf{v}\|^2$
 - 6: $\mathbf{x}_{t+1} \leftarrow (1 - \eta_t)\mathbf{x}_t + \eta_t \mathbf{v}_t$
 - 7: **end for**
-

D.1. Convergence rate of Algorithm 3

Theorem 12 Assume $f(\cdot), \mathcal{K}$ satisfy the quadratic growth property with parameter $\alpha > 0$. Then, using Algorithm 3 with step-size $\eta_t = \frac{2}{t+1}$ and mini-batch sizes that satisfy

$$m_t \geq \max \left\{ \left(\frac{G(t+1)}{\beta D_{\mathcal{K}}} \right)^2, \min \left\{ \left(\frac{GD_{\mathcal{K}}(t+1)}{\beta D^{*2}} \right)^2, \left(\frac{\alpha G(t+1)^2}{8\beta^2 D_{\mathcal{K}}} \right)^2 \right\} \right\},$$

one has,

$$\forall t \geq 2: \quad \mathbb{E}[f(\mathbf{x}_t) - f^*] \leq \frac{4\beta D^{*2}}{t+1} + \frac{32\beta^2 D_{\mathcal{K}}^2 \log(t)}{t^2}.$$

We first prove a lemma on the improvement (in expectation) on each iteration of Algorithm 3, and then we prove the theorem.

Lemma 13 Fix some iteration $t \geq 1$ of Algorithm 3 and let $d_t \geq \mathbb{E}[\text{dist}(\mathbf{x}_t, \mathcal{X}^*)^2]$. Then, using a step-size $\eta_t \in [0, 1]$ and mini-batch size $m_t \geq \left(\frac{2GD_{\mathcal{K}}}{\beta \eta_t \min\{D^{*2} + d_t, D_{\mathcal{K}}^2\}} \right)^2$ one has,

$$\bar{h}_{t+1} \leq (1 - \eta_t)\bar{h}_t + \beta \eta_t^2 (\min\{D^{*2} + d_t, D_{\mathcal{K}}^2\}).$$

Proof Let \mathbf{x}^* be the closest optimal solution to \mathbf{x}_t . We will upper-bound $\tilde{\nabla}_t^\top \mathbf{v}_t + \frac{\beta \eta_t}{2} \|\mathbf{v}_t - \mathbf{x}_t\|^2$ in two ways.

On one hand, consider an extreme point $\mathbf{u}^* \in \arg \min_{\mathbf{v} \in \mathcal{V}} \tilde{\nabla}_t^\top \mathbf{v}$. By the optimality of \mathbf{v}_t we have that,

$$\tilde{\nabla}_t^\top \mathbf{v}_t + \frac{\beta \eta_t}{2} \|\mathbf{v}_t - \mathbf{x}_t\|^2 \leq \tilde{\nabla}_t^\top \mathbf{u}^* + \frac{\beta \eta_t}{2} \|\mathbf{u}^* - \mathbf{x}_t\|^2 \leq \tilde{\nabla}_t^\top \mathbf{x}^* + \frac{\beta \eta_t D_{\mathcal{K}}^2}{2}.$$

On the other hand, since \mathbf{x}^* is a convex combination of extreme points from the set S^* , as in the proof of Lemma 8, there must exist some $\mathbf{w}^* \in S^*$ such that,

$$\tilde{\nabla}_t^\top \mathbf{w}^* + \beta \eta_t (\mathbf{w}^* - \mathbf{x}^*)^\top (\mathbf{x}^* - \mathbf{x}_t) \leq \tilde{\nabla}_t^\top \mathbf{x}^*.$$

Thus, denoting $\bar{d}_t = \text{dist}(\mathbf{x}_t, \mathcal{X}^*)$, by the optimality of \mathbf{v}_t we have that,

$$\begin{aligned} \tilde{\nabla}_t^\top \mathbf{v}_t + \frac{\beta\eta_t}{2} \|\mathbf{v}_t - \mathbf{x}_t\|^2 &\leq \tilde{\nabla}_t^\top \mathbf{w}^* + \frac{\beta\eta_t}{2} \|\mathbf{w}^* - \mathbf{x}_t\|^2 = \\ \tilde{\nabla}_t^\top \mathbf{w}^* + \frac{\beta\eta_t}{2} \|\mathbf{w}^* - \mathbf{x}^* + \mathbf{x}^* - \mathbf{x}_t\|^2 &= \\ \tilde{\nabla}_t^\top \mathbf{w}^* + \frac{\beta\eta_t}{2} \|\mathbf{w}^* - \mathbf{x}^*\|^2 + \beta\eta_t (\mathbf{w}^* - \mathbf{x}^*)^\top (\mathbf{x}^* - \mathbf{x}_t) + \frac{\beta\eta_t}{2} \|\mathbf{x}_t - \mathbf{x}^*\|^2 &\leq \\ \tilde{\nabla}_t^\top \mathbf{x}^* + \frac{\beta\eta_t D^{*2}}{2} + \frac{\beta\eta_t \bar{d}_t^2}{2}. \end{aligned}$$

Thus, denoting $\bar{M}_t = \min\{D^{*2} + \bar{d}_t^2, D_{\mathcal{K}}^2\}$, we have that,

$$\tilde{\nabla}_t^\top \mathbf{v}_t + \frac{\beta\eta_t}{2} \|\mathbf{v}_t - \mathbf{x}_t\|^2 \leq \tilde{\nabla}_t^\top \mathbf{x}^* + \frac{\beta\eta_t \bar{M}_t}{2}. \quad (15)$$

Now, from the β -smoothness of $f(\cdot)$ we have that,

$$\begin{aligned} f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) &\leq \nabla f(\mathbf{x}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{\beta}{2} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 \\ &= \eta_t \nabla f(\mathbf{x}_t)^\top (\mathbf{v}_t - \mathbf{x}_t) + \frac{\beta\eta_t^2}{2} \|\mathbf{v}_t - \mathbf{x}_t\|^2 \\ &= \eta_t (\nabla f(\mathbf{x}_t) - \tilde{\nabla}_t)^\top (\mathbf{v}_t - \mathbf{x}_t) + \eta_t \tilde{\nabla}_t^\top (\mathbf{v}_t - \mathbf{x}_t) + \frac{\beta\eta_t^2}{2} \|\mathbf{v}_t - \mathbf{x}_t\|^2 \\ &\stackrel{(a)}{\leq} \eta_t (\nabla f(\mathbf{x}_t) - \tilde{\nabla}_t)^\top (\mathbf{v}_t - \mathbf{x}_t) + \eta_t \tilde{\nabla}_t^\top (\mathbf{x}^* - \mathbf{x}_t) + \frac{\beta\eta_t^2 \bar{M}_t}{2} \\ &= \eta_t \nabla f(\mathbf{x}_t)^\top (\mathbf{x}^* - \mathbf{x}_t) + \eta_t (\nabla f(\mathbf{x}_t) - \tilde{\nabla}_t)^\top (\mathbf{v}_t - \mathbf{x}^*) + \frac{\beta\eta_t^2 \bar{M}_t}{2} \\ &\stackrel{(b)}{\leq} \eta_t (f(\mathbf{x}^*) - f(\mathbf{x}_t)) + \eta_t D_{\mathcal{K}} \|\tilde{\nabla}_t - \nabla f(\mathbf{x}_t)\| + \frac{\beta\eta_t^2 \bar{M}_t}{2}, \end{aligned}$$

where (a) follows from (15), and (b) follows from the convexity of $f(\cdot)$ and the Cauchy-Schwarz inequality.

Now, using Jensen's inequality we have that $\mathbb{E}[\|\tilde{\nabla}_t - \nabla f(\mathbf{x}_t)\|] \leq \sqrt{\mathbb{E}[\|\tilde{\nabla}_t - \nabla f(\mathbf{x}_t)\|^2]} \leq \frac{G}{\sqrt{m_t}}$, which by our choice of m_t , is at most $\frac{\beta\eta_t \min\{D^{*2} + d_t, D_{\mathcal{K}}^2\}}{2D_{\mathcal{K}}}$. Thus, taking expectation and noting that $\mathbb{E}[\bar{M}_t] = \mathbb{E}[\min\{D^{*2} + \bar{d}_t^2, D_{\mathcal{K}}^2\}] \leq \min\{D^{*2} + d_t, D_{\mathcal{K}}^2\}$, we have that

$$\bar{h}_{t+1} - \bar{h}_t \leq -\eta_t \bar{h}_t + \beta\eta_t^2 (\min\{D^{*2} + d_t, D_{\mathcal{K}}^2\}).$$

Finally, rearranging, we have that

$$\bar{h}_{t+1} \leq (1 - \eta_t) \bar{h}_t + \beta\eta_t^2 (\min\{D^{*2} + d_t, D_{\mathcal{K}}^2\}).$$

■

Proof [Proof of Theorem 12] First note that for any $t \geq 1$,

$$m_t \geq \left(\frac{G(t+1)}{\beta D_{\mathcal{K}}} \right)^2 = \left(\frac{2GD_{\mathcal{K}}}{\beta\eta_t \min\{D_{\mathcal{K}}^2, D^{*2} + D_{\mathcal{K}}^2\}} \right)^2.$$

Thus, we can use Lemma 13 with the trivial bound $D_{\mathcal{K}}^2 \geq \mathbb{E}[\text{dist}(\mathbf{x}_t, \mathcal{X}^*)^2]$ which, with our choice of step size η_t , implies that,

$$\forall t \geq 1: \quad \bar{h}_{t+1} \leq \left(1 - \frac{2}{t+1}\right)\bar{h}_t + \frac{4\beta D_{\mathcal{K}}^2}{(t+1)^2},$$

which by Lemma 10 (with $M = 4\beta D_{\mathcal{K}}^2$) gives us the bound,

$$\forall t \geq 2: \quad \bar{h}_t \leq \frac{4\beta D_{\mathcal{K}}^2}{t+1}. \quad (16)$$

Now, for any $t \geq 1$, let $d_t = \frac{8\beta D_{\mathcal{K}}^2}{\alpha(t+1)}$. By the quadratic growth of $f(\cdot)$ and (16), we have that $\mathbb{E}[\text{dist}(\mathbf{x}_t, \mathcal{X}^*)^2] \leq d_t$. Thus, since the mini-batch sizes satisfy for all $t \geq 1$,

$$\begin{aligned} m_t &\geq \max \left\{ \left(\frac{G(t+1)}{\beta D_{\mathcal{K}}} \right)^2, \min \left\{ \left(\frac{GD_{\mathcal{K}}(t+1)}{\beta D^{*2}} \right)^2, \left(\frac{\alpha G(t+1)^2}{8\beta^2 D_{\mathcal{K}}} \right)^2 \right\} \right\} \\ &= \max \left\{ \left(\frac{2GD_{\mathcal{K}}}{\beta \eta_t D_{\mathcal{K}}^2} \right)^2, \min \left\{ \left(\frac{2GD_{\mathcal{K}}}{\beta \eta_t D^{*2}} \right)^2, \left(\frac{2GD_{\mathcal{K}}}{\beta \eta_t d_t} \right)^2 \right\} \right\} \\ &= \left(\frac{2GD_{\mathcal{K}}}{\beta \eta_t \min\{D_{\mathcal{K}}^2, \max\{D^{*2}, d_t\}\}} \right)^2 \geq \left(\frac{2GD_{\mathcal{K}}}{\beta \eta_t \min\{D_{\mathcal{K}}^2, D^{*2} + d_t\}} \right)^2, \end{aligned}$$

we can use Lemma 13 with $d_t = \frac{8\beta D_{\mathcal{K}}^2}{\alpha(t+1)}$, which, by our choice of step size η_t implies that,

$$\forall t \geq 1: \quad \bar{h}_{t+1} \leq \left(1 - \frac{2}{t+1}\right)\bar{h}_t + \frac{4\beta}{(t+1)^2} \left(D^{*2} + \frac{8\beta D_{\mathcal{K}}^2}{\alpha(t+1)}\right).$$

Thus, by Lemma 10, for any $t \geq 2$, we have that,

$$\begin{aligned} \bar{h}_t &\leq \frac{1}{t^2} \sum_{k=1}^{t-1} 4\beta \left(D^{*2} + \frac{8\beta D_{\mathcal{K}}^2}{\alpha(k+1)}\right) \leq \frac{\frac{1}{t-1} \sum_{k=1}^{t-1} 4\beta D^{*2}}{t+1} + \frac{32\beta^2 D_{\mathcal{K}}^2}{\alpha} \cdot \frac{\sum_{k=1}^t \frac{1}{k} - 1}{t^2} \\ &\leq \frac{4\beta D^{*2}}{t+1} + \frac{32\beta^2 D_{\mathcal{K}}^2 \log(t)}{t^2}, \end{aligned}$$

concluding the proof. ■

D.2. Proof of Theorem 7

For clarity, we first restate the theorem and then prove it.

Theorem 7 *Suppose f, \mathcal{K} satisfy the quadratic growth property with some $\alpha > 0$. Using Algorithm 3 with step-size $\eta_t = \frac{2}{t+1}$ and mini-batch sizes that satisfy*

$$m_t = \max \left\{ \left(\frac{G(t+1)}{\beta D_{\mathcal{K}}} \right)^2, \min \left\{ \left(\frac{GD_{\mathcal{K}}(t+1)}{\beta D^{*2}} \right)^2, \left(\frac{\alpha G(t+1)^2}{8\beta^2 D_{\mathcal{K}}} \right)^2 \right\} \right\},$$

for any $0 < \epsilon \leq \frac{16\beta^2 D_{\mathcal{K}}^2}{\alpha}$, expected approximation error ϵ is achieved after $\tilde{O}\left(\max\left\{\frac{\beta D^{*2}}{\epsilon}, \frac{\beta D_{\mathcal{K}}}{\sqrt{\alpha\epsilon}}\right\}\right)$ calls to the NEP oracle and $\tilde{O}\left(\beta G^2 \max\left\{\frac{D_{\mathcal{K}}^2 D^{*2}}{\epsilon^3}, \frac{D_{\mathcal{K}}}{\alpha^{3/2} \epsilon^{3/2}}, \frac{D_{\mathcal{K}}^3}{\alpha^{1/2} \epsilon^{5/2}}\right\}\right)$ stochastic gradient evaluations, where \tilde{O} suppresses poly-logarithmic terms in $\frac{\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}$.

Proof Let $0 < \epsilon \leq \frac{16\beta^2 D_{\mathcal{K}}^2}{\alpha}$ and note that for $t \geq 8\beta D^{*2}\epsilon^{-1}$, it holds that $\frac{4\beta D^{*2}}{t+1} \leq \frac{\epsilon}{2}$. Now, let $M > 0$ such that $\sqrt{\frac{2M}{\epsilon} \log(\frac{2M}{\epsilon})} \geq 2$, and observe that for all $t \geq \sqrt{\frac{2M}{\epsilon} \log(\frac{2M}{\epsilon})}$ it holds that,

$$\frac{M \log(t)}{t^2} \stackrel{(a)}{\leq} \frac{M \log(\sqrt{\frac{2M}{\epsilon} \log(\frac{2M}{\epsilon})})}{\frac{2M}{\epsilon} \log(\frac{2M}{\epsilon})} \stackrel{(b)}{\leq} \frac{M \log(\frac{2M}{\epsilon})}{\frac{2M}{\epsilon} \log(\frac{2M}{\epsilon})} = \frac{\epsilon}{2},$$

where (a) holds since $\frac{M \log(t)}{t^2}$ is monotonically decreasing in t for $t \geq 2$, and (b) holds since $\log(\frac{2M}{\epsilon}) \leq \frac{2M}{\epsilon}$.

Thus, taking $M = \frac{32\beta^2 D_{\mathcal{K}}^2}{\alpha}$, note that since $0 < \epsilon \leq \frac{16\beta^2 D_{\mathcal{K}}^2}{\alpha} = \frac{M}{2}$, we have that, $\sqrt{\frac{2M}{\epsilon} \log(\frac{2M}{\epsilon})} \geq \sqrt{4 \log(4)} > 2$, and thus, we have that,

$$\forall t \geq \sqrt{\frac{64\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon} \log\left(\frac{64\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)} : \frac{32\beta^2 D_{\mathcal{K}}^2 \log(t)}{t^2} \leq \frac{\epsilon}{2}.$$

Thus, denoting $T = \left\lceil \max\left\{\frac{8\beta D^{*2}}{\epsilon}, \sqrt{\frac{64\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon} \log\left(\frac{64\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)}\right\} \right\rceil$, using Theorem 12 we have that for all $t \geq T$,

$$\bar{h}_t \leq \frac{4\beta D^{*2}}{t+1} + \frac{32\beta^2 D_{\mathcal{K}}^2 \log(t)}{t^2} \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

Thus, we indeed reach ϵ expected approximation error in $O\left(\max\left\{\frac{\beta D^{*2}}{\epsilon}, \frac{\beta D_{\mathcal{K}} \log^{1/2}\left(\frac{\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)}{\sqrt{\alpha\epsilon}}\right\}\right)$ calls to the linear oracle.

Now, let n_{grad} be the number of stochastic gradient used until iteration T . Note that we have that,

$$\begin{aligned} n_{grad} &= \sum_{t=1}^{T-1} m_t \leq T m_{T-1} = T \max\left\{\left(\frac{GT}{\beta D_{\mathcal{K}}}\right)^2, \min\left\{\left(\frac{GD_{\mathcal{K}}T}{\beta D^{*2}}\right)^2, \left(\frac{\alpha GT^2}{8\beta^2 D_{\mathcal{K}}}\right)^2\right\}\right\} \\ &= \max\left\{\frac{G^2 T^3}{\beta^2 D_{\mathcal{K}}^2}, \min\left\{\frac{G^2 D_{\mathcal{K}}^2 T^3}{\beta^2 D^{*4}}, \frac{\alpha^2 G^2 T^5}{64\beta^4 D_{\mathcal{K}}^2}\right\}\right\}. \end{aligned} \quad (17)$$

Thus, when $\frac{8\beta D^{*2}}{\epsilon} \geq \sqrt{\frac{64\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon} \log\left(\frac{64\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)}$ we have that, $T = \lceil \frac{8\beta D^{*2}}{\epsilon} \rceil = O\left(\frac{\beta D^{*2}}{\epsilon}\right)$, and thus using (17), we have that,

$$\begin{aligned} n_{grad} &\leq \max\left\{\frac{G^2 T^3}{\beta^2 D_{\mathcal{K}}^2}, \frac{G^2 D_{\mathcal{K}}^2 T^3}{\beta^2 D^{*4}}\right\} \stackrel{(a)}{=} \frac{G^2 D_{\mathcal{K}}^2 T^3}{\beta^2 D^{*4}} = O\left(\frac{G^2 D_{\mathcal{K}}^2}{\beta^2 D^{*4}} \cdot \frac{\beta^3 D^{*6}}{\epsilon^3}\right) \\ &= O\left(\frac{\beta G^2 D_{\mathcal{K}}^2 D^{*2}}{\epsilon^3}\right), \end{aligned}$$

where (a) holds since $D^* \leq D_{\mathcal{K}}$.

Otherwise, we have that, $T = \left\lceil \sqrt{\frac{64\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon} \log\left(\frac{64\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)} \right\rceil = O\left(\frac{\beta D_{\mathcal{K}} \log^{1/2}\left(\frac{\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)}{\sqrt{\alpha\epsilon}}\right)$, and thus, using (17) we have that,

$$\begin{aligned} n_{grad} &\leq \max \left\{ \frac{G^2 T^3}{\beta^2 D_{\mathcal{K}}^2}, \frac{\alpha^2 G^2 T^5}{64\beta^4 D_{\mathcal{K}}^2} \right\} \\ &= O \left(\max \left\{ \frac{G^2}{\beta^2 D_{\mathcal{K}}^2} \cdot \frac{\beta^3 D_{\mathcal{K}}^3 \log^{3/2}\left(\frac{\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)}{\alpha^{3/2} \epsilon^{3/2}}, \frac{\alpha^2 G^2}{\beta^4 D_{\mathcal{K}}^2} \cdot \frac{\beta^5 D_{\mathcal{K}}^5 \log^{5/2}\left(\frac{\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)}{\alpha^{5/2} \epsilon^{5/2}} \right\} \right) \\ &= O \left(\max \left\{ \frac{\beta G^2 D_{\mathcal{K}} \log^{3/2}\left(\frac{\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)}{\alpha^{3/2} \epsilon^{3/2}}, \frac{\beta G^2 D_{\mathcal{K}}^3 \log^{5/2}\left(\frac{\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)}{\alpha^{1/2} \epsilon^{5/2}} \right\} \right). \end{aligned}$$

Thus, we indeed achieve ϵ expected approximation error after

$$O \left(\beta G^2 \max \left\{ \frac{D_{\mathcal{K}}^2 D^{*2}}{\epsilon^3}, \frac{D_{\mathcal{K}} \log^{3/2}\left(\frac{\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)}{\alpha^{3/2} \epsilon^{3/2}}, \frac{D_{\mathcal{K}}^3 \log^{5/2}\left(\frac{\beta^2 D_{\mathcal{K}}^2}{\alpha\epsilon}\right)}{\alpha^{1/2} \epsilon^{5/2}} \right\} \right)$$

stochastic gradient evaluations. ■

Appendix E. Additional numerical results

E.1. Hypercube-constrained least-squares

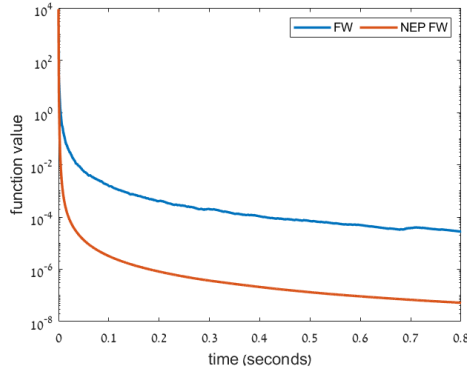


Figure 3: Comparison between the standard Frank-Wolfe method (FW) and our NEP FW variant on the hypercube-constrained least-squares problem. These are the same results as in Figure 1 but focusing only on these two variants.

E.2. Video co-localization

In Figure 6 we present the performance of the algorithms measured by the duality gap (as was done in Lacoste-Julien and Jaggi (2015) and Garber and Meshi (2016)). It can be seen that although

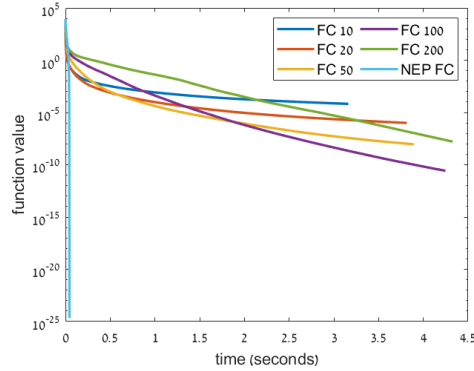


Figure 4: Comparison between our NEP FC variant with 50 iterations of FISTA per iteration and the FC variant with various choices for the number of FISTA iterations (the number of FISTA iterations is shown besides the algorithm’s name) on the hypercube-constrained least-squares problem. The results are the average of 50 i.i.d. runs (where in each run we sample fresh A, x^*).

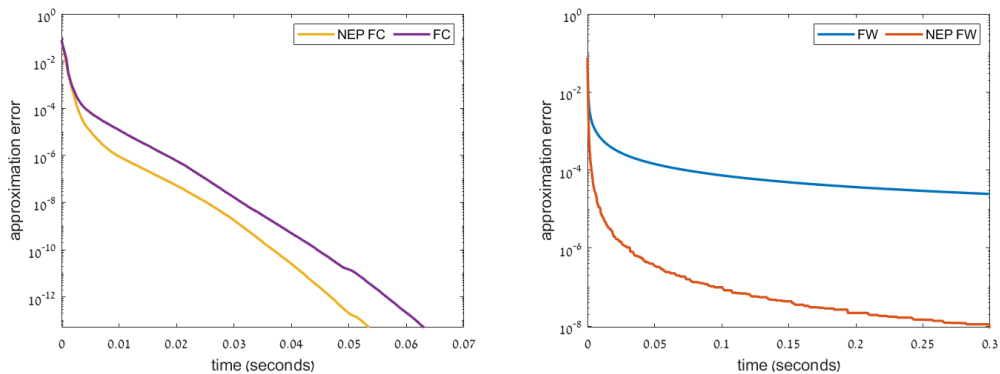


Figure 5: Comparison of Frank-Wolfe variants on the video co-localization problem. The times shown are the averages of 200 runs. These are the same results as in Figure 2, but focusing on FC vs. NEP FC (left panel) and FW vs. NEP FW (right panel).

NEP FC still outperforms all other algorithms, it only gives a slight improvement over FC and that NEP FW only slightly outperforms FW, and is outperformed by AFW and PFW both with respect to time and number of iterations. Here we remind the reader that while we proved the theoretical superiority of our NEP oracle-based algorithms w.r.t. the (primal) approximation error, we did not give any improved bounds w.r.t. the duality gap, and we leave it for future work to settle the question whether or not the use of a NEP oracle could lead to provably faster dual convergence.

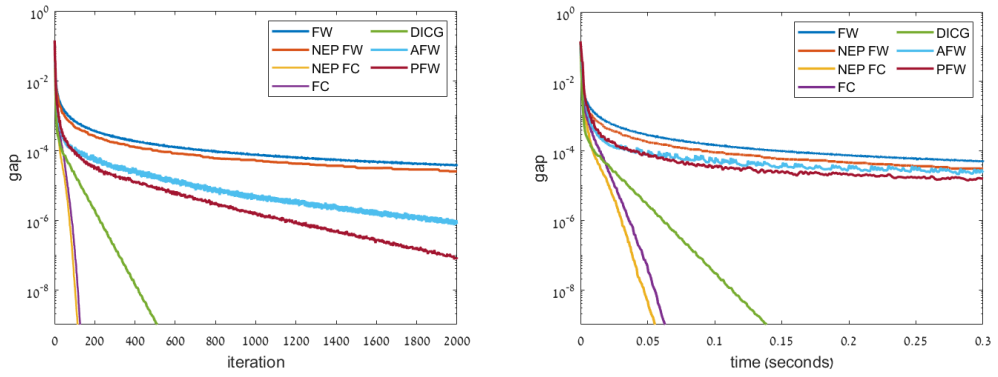


Figure 6: Comparison of Frank-Wolfe variants on the video co-localization problem in terms of the duality gap. The times shown are the averages of 200 runs.