Jahresbericht der Deutschen Mathematiker-Vereinigung (2023) 126:3–35
https://doi.org/10.1365/s13291-023-00275-x

SURVEY ARTICLE

The Frank-Wolfe Algorithm: A Short Introduction

Sebastian Pokutta[1]

Accepted: 30 November 2023 / Published online: 13 December 2023

Abstract
In this paper we provide an introduction to the Frank-Wolfe algorithm, a method for smooth convex optimization in the presence of (relatively) complicated constraints. We will present the algorithm, introduce key concepts, and establish important baseline results, such as e.g., primal and dual convergence. We will also discuss some of its properties, present a new adaptive step-size strategy as well as applications.

1 Introduction

Throughout this paper we will be concerned with *constrained optimization problems* of the form

$$\min_{x \in P} f(x), \tag{Opt}$$

where $P \subseteq \mathbb{R}^n$ is some convex feasible region capturing the constraints, e.g., a polyhedron arising from a system of linear inequalities or a spectrahedron, and f is the objective function satisfying some regularity property, e.g., smoothness and convexity. We also need to specify what access methods we have, both, to the function and the feasible region. A common setup is black box first-order access for f, allowing (only) the computation of gradients $\nabla f(x)$ for a given point x as well as the function value $f(x)$. For the access to the feasible region P, which we will assume to be compact in the following, there are several common models; we simplify the exposition here for the sake of brevity:

1. *Projection.* Access to the projection operator Π_P of P that, for a given point $x \in \mathbb{R}^n$ returns $\Pi_P(x) \doteq \operatorname{argmin}_{y \in P} \|x - y\|$, for some norm $\|.\|$ (or more generally Bregman divergences).
2. *Barrier function.* Access to a barrier function of the feasible region P that increases in value to infinity when approaching the boundary of P. A typical example is, the barrier function $-\sum_i \log(b_i - A_i x)$ for a linear inequality system $P \doteq \{x \mid Ax \leq b\}$.

✉ S. Pokutta
pokutta@zib.de

[1] Zuse Institute Berlin & TU Berlin, Berlin, Germany

3. *Linear Minimization.* Access to a Linear Minimization Oracle (LMO) that, given a linear objective $c \in \mathbb{R}^n$, returns $y \in \operatorname{argmin}_{x \in P} \langle c, x \rangle$.

Specialized approaches for specific cases, e.g., the simplex method (Dantzig [27, 28]) in the case of linear objectives which uses an explicit description of the feasible region also exist but here we concentrate on the aforementioned black box model. There are also proximal methods, which can be considered generalizations of projection-based methods and which we will not explicitly consider for the sake of brevity; see e.g., Nemirovski and Yudin [65], Nesterov [66, 67], Nocedal and Wright [68] for a discussion.

Traditionally, problems of the form (Opt) are solved by variants of *projection-based methods*. In particular first-order methods, such as variants of projected gradient descent are often chosen in large-scale contexts as they are comparatively cheap. For some feasible region P with projector Π_P (e.g., $\Pi_P(x) \doteq \operatorname{argmin}_{y \in P} \|x - y\|$) and smooth objective function f, *projected gradient descent (PGD)* updates typically take the form:

$$\begin{aligned} x_{t+\frac{1}{2}} &\leftarrow x_t - \gamma_t \nabla f(x_t) \\ x_{t+1} &\leftarrow \Pi_P(x_{t+\frac{1}{2}}), \end{aligned} \tag{PGD}$$

where γ_t is some step-size, e.g., $\gamma_t = 1/L$ if f is L-smooth (see Definition 2.1) and convex. In essence, a descent step is taken without considering the constraints, and then it is projected back into the feasible region (see Fig. 1). Projection-based first-order methods have been extensively studied, with comprehensive overviews available in, e.g., Nesterov [67], Nocedal and Wright [68]. Optimal methods and rates are known for most scenarios. Efficient execution of the projection operation is possible for simple constraints, such as box constraints or highly structured feasible regions, e.g., as discussed in Gupta et al. [43], Moondra et al. [63] for submodular base polytopes. However, when the feasible region grows in complexity, the projection operation can become the limiting factor. It often demands the solution of an auxiliary optimization problem—known as the *projection problem*—over the same feasible region for *every* descent step. This complexity renders the use of projection-based methods for many significant constrained problems quite challenging; in some cases relaxed projections which essentially compute separating hyperplanes can be used though.

Interior point methods (IPM) offer an alternative approach, see e.g., Boyd et al. [8], Potra and Wright [71]. To illustrate this approach, consider the goal of minimizing a linear function c over a polytope defined as $P \doteq \{x \mid Ax \leq b\}$. The typical updates in a path-following IPM resemble:

$$x_\mu \leftarrow \operatorname{argmin}_x \langle c, x \rangle + \mu \sum \log(b_i - A_i x), \tag{IPM}$$

where the value of $\mu \to 0$ according to some strategy for μ. Often, these steps are only approximately solved. IPMs, while potent with appealing theoretical guarantees, usually necessitate a barrier function that encapsulates the feasible region's description. In numerous critical scenarios, a concise feasible region description is either

unknown or proven to be non-existent. For instance, the matching polytope does not admit small linear programs, neither exact ones (Rothvoss [73]) nor approximate ones (Braun and Pokutta [9, 10], Sinha [74]). Additionally, achieving sufficient accuracy in the IPM step updates often requires second-order information, which can sometimes restrict its applicability.

Upon closely examining the two methods mentioned earlier, it is clear that both essentially transform the constrained problem (Opt) into an *unconstrained* one. They then either correct updates that violate constraints (as in PGD) or penalize nearing constraint violations (as in IPM). Yet, another category of techniques exists, termed *projection-free methods*, which focus directly on constrained optimization. Unlike their counterparts, these methods sidestep the need for costly projections or penalty strategies and maintain feasibility throughout the process. The most notable variants in this category are the *Frank-Wolfe (FW) methods*—going back to Frank and Wolfe [34]—which will be the focus of this article and which are also known as *conditional gradient (CG) methods* (Levitin and Polyak [57]).

Historically, methods like the Frank-Wolfe algorithm garnered limited attention because of certain drawbacks, notably sub-optimal convergence rates. However, there was a notable resurgence in interest around 2013. This revival is largely attributed to shifting requirements and their other, now suddenly relevant properties. Notably, these methods are well suited to handle complicated constraints and possess a low iteration complexity. This makes them very effective in the context of large-scale machine learning problems (see, e.g., Lacoste-Julien et al. [54], Jaggi [48], Négiar et al. [64], Dahik [26], Jing et al. [49]), image processing (see, e.g., Joulin et al. [50], Tang et al. [75]), quantum physics (see, e.g., Gilbert [41], Designolle et al. [30]), submodular function maximization (see, e.g., Feldman et al. [33], Vondrák [79], Badanidiyuru and Vondrák [5], Mirzasoleiman et al. [60], Hassani et al. [45], Mokhtari et al. [61], Anari et al. [1], Anari et al. [2], Mokhtari et al. [62], Bach [4]), online learning (see, e.g., Hazan and Kale [46], Zhang et al. [86], Chen et al. [20], Garber and Kretzu [39], Kerdreux et al. [51], Zhang et al. [87]) and many more (see, e.g., Bolte et al. [6], Clarkson [22], Pierucci et al. [70], Harchaoui et al. [44], Wang et al. [81], Cheung and Li [21], Ravi et al. [72], Hazan and Minasyan [47], Dvurechensky et al. [32], Carderera and Pokutta [17], Macdonald et al. [58], Carderera et al. [18], Garber and Wolf [40], Bomze et al. [7], Wäldchen et al. [80], Chen and Sun [19], de Oliveira [29], Designolle et al. [30], Designolle et al. [31], Lacoste-Julien [52]). Moreover, there has been a proliferation of modifications to these methods, addressing many of their historical limitations (see, e.g., Freund et al. [35], Lacoste-Julien and Jaggi [53], Garber and Hazan [37, 38], Lan et al. [56], Braun et al. [12], Braun et al. [14], Braun et al. [13], Combettes and Pokutta [23], Tsuji et al. [78]) and there is an intricate connection between Frank-Wolfe methods and subgradients methods (Bach [3]); see Braun et al. [15] for a comprehensive exposition.

Rather than relying on potentially expensive projection operations (see Fig. 2), Frank-Wolfe methods use a so-called Linear Minimization Oracle (LMO). This subroutine only involves optimizing a linear function over the feasible region, often proving more cost-effective than traditional projections; see Combettes and Pokutta [24] for an in-depth comparison. The nuclear norm ball, along with matrix completion, is a prime example highlighting the difference in complexity. The core updates in

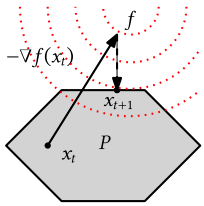

Fig. 1 Projection-based methods: may require (potentially expensive) projection back into P to ensure feasibility

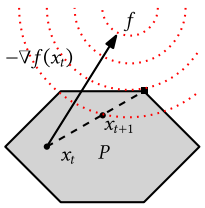

Fig. 2 Projection-free methods: ensure feasibility by forming convex combinations only

Fig. 3 New iterates are formed by convex combination with an extreme point approximating the gradient, ensuring feasibility

Algorithm 1: FW algorithm

1 $x_0 \in P$
2 **for** $t = 0$ **to** $T - 1$ **do**
3 $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
4 $\quad x_{t+1} \leftarrow (1 - \gamma_t)x_t + \gamma_t v_t$
5 **end for**

Frank-Wolfe methods often rely on the following fundamental update:

$$\begin{aligned} v_t &\leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle \\ x_{t+1} &\leftarrow (1 - \gamma_t)x_t + \gamma_t v_t, \end{aligned} \tag{FW}$$

where any solution to the argmin is suitable and γ_t follows some step-size strategy, e.g., $\gamma_t = \frac{2}{2+t}$. Essentially, the LMO identifies an alternate direction for descent. Subsequently, convex combinations of points are constructed within the feasible region to maintain feasibility. Viewed through the lens of complexity theory, the Frank-Wolfe methods reduce the optimization of a convex function f over P into the repeated optimization of evolving linear functions over P. A schematic of the most basic variant of the Frank-Wolfe algorithm is shown in Fig. 3.

For a more comprehensive exposition complementing this introductory article the interested reader is referred to Braun et al. [15]; the notation has been deliberately chosen to be matching whenever possible.

1.1 Outline

We start with some basic notions and notations in Sect. 2 and then present the original Frank-Wolfe algorithm along with some motivation in Sect. 3. We then proceed in Sect. 4 with establishing basic properties, such as e.g., convergence and also provide matching lower bounds. While this is primarily an overview article, we do provide a new adaptive step-size strategy in Sect. 4.5, which is also available in the

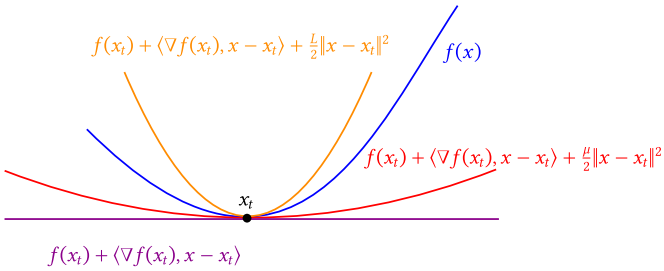

Fig. 4 (Strong) convexity and smoothness provide linear and quadratic approximations to f. Orange: quadratic upper bound via smoothness, Red: quadratic lower bound via strong convexity, Magenta: linear lower bound via convexity, Blue: function f

FrankWolfe.jl julia package. In Sect. 5 we then consider applications of the Frank-Wolfe algorithm and also discuss computational aspects in Sect. 6.

2 Preliminaries

In the following $\|\cdot\|$ will denote the 2-norm if not stated otherwise. Note however that in general other norms are possible and have been used in the context of Frank-Wolfe algorithms. Moreover, for simplicity we assume that f is differentiable, which is a standard assumption in the context of Frank-Wolfe algorithms although non-smooth variants are known (see, e.g., Braun et al. [15] for details).

For our analysis we will heavily rely on the following key concepts:

Definition 2.1 (Convexity and Strong Convexity) Let $f\colon P \to \mathbb{R}$ be a differentiable function. Then f is *convex* if

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle \quad \text{for all } x, y \in P. \tag{2.1}$$

Moreover, f is μ-*strongly convex* if

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2 \quad \text{for all } x, y \in P. \tag{2.2}$$

Definition 2.2 (Smoothness) Let $f\colon P \to \mathbb{R}$ be a differentiable function. Then f is L-*smooth* if

$$f(y) - f(x) \leq \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 \quad \text{for all } x, y \in P. \tag{2.3}$$

The smoothness and (strong) convexity inequalities from above allow us to obtain upper and lower bounds on the function f. Convexity and strong convexity provide respectively linear and quadratic lower bounds on the function f at a given point x while smoothness provides a quadratic upper bound as shown in Fig. 4.

For completeness we note that, both, L-smoothness and μ-strong convexity can also be expressed without relying on function values of f only using gradients ∇f. This is in particular useful in the context of an adaptive step-size strategy that we will discuss in Sect. 4.5 as it significantly improves numerical stability of the estimates.

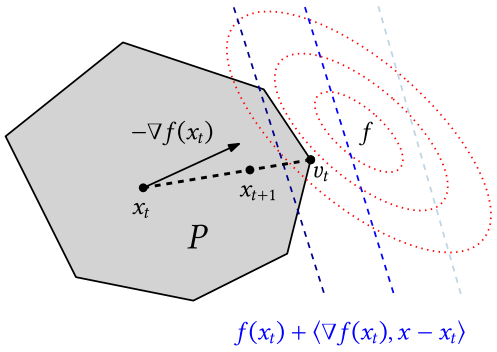

Fig. 5 The Frank–Wolfe step: To minimize a convex function f over a polytope P, a linear approximation of f is constructed at x_t as $f(x_t) + \langle \nabla f(x_t), x - x_t \rangle$. The Frank–Wolfe vertex v_t minimizes this approximation. The step transitions from x_t to x_{t+1} by moving towards v_t as determined by a step-size rule. Contour lines of f in red and linear approximation blue (Color figure online)

Remark 2.3 (Smoothness and Strong Convexity via Gradients) Let $f\colon P \to \mathbb{R}$ be a differentiable function. Then f is *L-smooth* if

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle \leq L \|y - x\|^2 \quad \text{for all } x, y \in P, \tag{2.4}$$

and similarly f is *μ-strongly convex* if

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle \geq \mu \|y - x\|^2 \quad \text{for all } x, y \in P. \tag{2.5}$$

There is also the closely related and seemingly stronger property of L-Lipschitz continuous gradient ∇f, however in the case that P is full-dimensional and f is convex it is known to be equivalent to L-smoothness (see Nesterov [67, Theorem 2.1.5] for the unbounded case, i.e., where $P = \mathbb{R}^n$ and Braun et al. [15, Lemma 1.7] for P being arbitrary convex domain). In particular, for twice differentiable convex functions f, we can also capture smoothness and strong convexity in terms of the Hessian via $\|\nabla^2 f\| \leq L$ and via the largest eigenvalue of $\nabla^2 f$ being upper bounded by $L \geq 0$ and the smallest eigenvalue being lower bounded by $\mu \geq 0$, respectively; the first inequality is useful for numerical estimation of L.

In the following the domain P will be a compact convex set and we assume that we have access to a so-called *Linear Minimization Oracle (LMO)* for P, which upon being provided with a linear objective function c returns a minimizer $v = \operatorname{argmin}_{x \in P} \langle c, x \rangle$ as formalized in Algorithm 2. Note that v is not necessarily unique and without loss of generality we assume that v is an extreme point of P; these extreme points are also often called *atoms* in the context of Frank-Wolfe algorithms. For the compact convex set P the *diameter* D of P is defined as $D \doteq \max_{x,y \in P} \|x - y\|$.

Regarding the function f we assume that we have access to gradients and function evaluations which is formalized as *First-Order Oracle* (denoted as FOO), which given a point $x \in P$, returns the function value $f(x)$ and gradient $\nabla f(x)$ at x; see Algorithm 3. In the following we let x^* be one of the optimal solutions to $\min_{x \in P} f(x)$ and define further $f^* \doteq f(x^*)$. Moreover, if not stated otherwise we consider $f\colon P \to \mathbb{R}$. See Fig. 6 for the two oracles.

Algorithm 2: Linear Minimization Oracle over P (LMO)

Input: Linear objective c
Output: $v \in \operatorname{argmin}_{x \in P} \langle c, x \rangle$

Algorithm 3: First-Order Oracle for f (FOO)

Input: Point $x \in P$
Output: $\nabla f(x)$ and $f(x)$

Fig. 6 Access to function f and feasible region P is via two functions that we assume to have (oracle) access to

Algorithmus 4 : Frank–Wolfe algorithm

Input: Initial atom $x_0 \in P$, smooth and convex objective function f
Output: Iterates $x_1, \ldots \in P$

1 **for** $t = 0$ **to** $\ldots$ **do**
2 $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
3 $\quad \gamma_t \leftarrow \frac{2}{2+t}$
4 $\quad x_{t+1} \leftarrow (1 - \gamma_t) x_t + \gamma_t v_t$
5 **end for**

3 The Frank-Wolfe Algorithm

We will now introduce the original variant of the Frank-Wolfe (FW) algorithm due to Frank and Wolfe [34], which is often also referred to as Conditional Gradients (Levitin and Polyak [57]). Although many advanced variants with enhanced properties and improved convergence in specific problem configurations exist today, we will focus on the original version for clarity and to underscore the fundamental concepts.

Suppose we are interested in minimizing a smooth and convex function f over some compact convex feasible set P. A natural strategy would be to follow the negative of the gradient $\nabla f(x)$ at a given point x. However, how far can we go into that direction before we hit the boundary of the feasible region? Moreover, even if we would know how far we can go, i.e., we would potentially truncate steps to not leave the feasible region, even then the resulting algorithm might not be converging to an optimal solution. In fact, the arguably most well-known strategy, the *projected gradient descent* method does not simply stop at the boundary but follows the negative of the gradient according to some step-size, disregarding the constraints, and then *projects back* onto the feasible region. This last step can be very costly: if we do not have an efficient formulation or algorithm for the projection problem, solving this projection problem can be a (relatively expensive) optimization problem in itself. In contrast, the basic idea of the Frank-Wolfe algorithm is to *not* follow the negative of the gradient but to follow an alternative direction of descent, which is well-enough aligned with the negative of the gradient, ensures enough primal progress, and for which we can easily ensure feasibility by means of computing convex combinations. This is done via the aforementioned Linear Minimization Oracle, with which we can optimize the negative of the gradient over the feasible region P and then take the obtained vertex to form an alternative direction of descent. The overall process is outlined in Fig. 5 and in Algorithm 4 we provide the Frank-Wolfe algorithm, which only requires access to (Opt) via the LMO (see Algorithm 2) to access the feasible region and via the FOO (see Algorithm 3) to access the function.

As can be seen, assuming access to the two oracles, the actual implementation is very straight-forward: a simple computation of a convex combination, which ensures that we do not leave the feasible region. We made the deliberate choice in Line 3 of Algorithm 4 to use the most basic step-size strategy $\gamma_t = \frac{2}{2+t}$, the so-called *open loop* or *agnostic* step-size, as this makes the algorithm *parameter-independent*, i.e., not requiring any function parameters or parameter estimations. In the worst-case, this step-size is not dominated by more elaborate strategies (such as, e.g., line search or short steps), however in many important special cases there are better choices. As this is crucial we will discuss this a little more in-depth in Sect. 4.5 and will also provide a new variant of an adaptive step-size strategy.

Another important property is that the algorithm is *affine invariant*, i.e., problem rescaling etc. does not affect the algorithm's performance, compared to most other methods including PGD (notable exceptions exist, e.g., Newton's method). This makes the algorithm also very robust (especially with the open loop step-sizes) often offering superior numerical stability.

Finally, we would like to mention that at iteration t the iterate x_t is a convex combination of at most $t+1$ extreme points (or atoms) of P. This will allow us later to obtain sparsity vs. approximation trade-offs in Sect. 4.1.

4 Properties

We will now establish key properties of Algorithm 4. We start with convergence properties and will then establish matching lower bounds as well as other properties.

4.1 Convergence

We will now prove the convergence of the Frank-Wolfe algorithm (Algorithm 4). Convergence proofs for these methods typically use two key ingredients, which we will introduce in the following.

Lemma 4.1 (Primal gap, Dual gap, and Frank-Wolfe gap) *Let f be a convex function and P a compact convex set and consider* (Opt). *For all $x \in P$ it holds*:

$$\underbrace{f(x) - f(x^*)}_{\text{primal gap at } x} \leq \underbrace{\langle \nabla f(x), x - x^* \rangle}_{\text{dual gap at } x} \leq \underbrace{\max_{v \in P} \langle \nabla f(x), x - v \rangle}_{\text{Frank-Wolfe gap at } x}. \tag{FW-gap}$$

Proof The first inequality follows from convexity and the second inequality follows from maximality. □

The Frank-Wolfe gap plays a crucial role in the theory of Frank-Wolfe methods as it provides an easily computable optimality certificate and suboptimality gap measure. An extreme point $v \in \operatorname{argmax}_{z \in P} \langle \nabla f(x), x - z \rangle$, is typically referred to as *Frank-Wolfe vertex for* $\nabla f(x)$. The Frank-Wolfe gap also naturally appear in the first-order optimality condition for (Opt), which states that $x^* \in P$ is optimal for (Opt) if and only if the Frank-Wolfe gap at x^* is equal to 0. Note that in the constrained case it does not necessarily hold that $\nabla f(x^*) = 0$.

Lemma 4.2 (First-order Optimality Condition) *Let $x^* \in P$. Then x^* is an optimal solution to* (Opt) *if and only if*

$$\langle \nabla f(x^*), x^* - v \rangle \leq 0$$

for all $v \in P$. In particular, we have that the Frank-Wolfe gap $\max_{v \in P} \langle \nabla f(x^*), x^* - v \rangle = 0$.

The second property that is crucial is smoothness as it allows us to lower bound the primal progress we can derive from a step of the Frank-Wolfe algorithm.

Lemma 4.3 (Primal progress from smoothness) *Let f be an L-smooth function and let $x_{t+1} = (1 - \gamma_t)x_t + \gamma_t v_t$ with $x_t, v_t \in P$. Then we have*

$$f(x_t) - f(x_{t+1}) \geq \gamma_t \langle \nabla f(x_t), x_t - v_t \rangle - \gamma_t^2 \frac{L}{2} \|x_t - v_t\|^2. \tag{4.1}$$

Proof The statement follows directly from the smoothness inequality (2.3)

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

choosing $x \leftarrow x_t$ and $y \leftarrow x_{t+1}$, plugging in the definition of x_{t+1}, and rearranging. This gives the desired inequality

$$f(x_t) - f(x_{t+1}) \geq \gamma_t \langle \nabla f(x_t), x_t - v_t \rangle - \gamma_t^2 \frac{L}{2} \|x_t - v_t\|^2. \qquad \square$$

With these two key ingredients (Lemma 4.1 and Lemma 4.3) we can now establish the basic convergence rate of the Frank-Wolfe algorithm:

Theorem 4.4 (Primal convergence of the Frank-Wolfe algorithm) *Let f be an L-smooth convex function and let P be a compact convex set of diameter D. Consider the iterates of Algorithm 4. Then the following holds*:

$$f(x_t) - f(x^*) \leq \frac{2LD^2}{t+2},$$

and hence for any accuracy $\varepsilon > 0$ we have $f(x_t) - f(x^*) \leq \varepsilon$ for all $t \geq \frac{2LD^2}{\varepsilon}$.

Proof The convergence proof of the Frank-Wolfe algorithm follows an approach that is quite representative for convergence results in that area. The proof follows the template outlined in Braun et al. [15] and mimics closely the proof in Jaggi [48].

Our starting point is the inequality from Lemma 4.3

$$\begin{aligned} f(x_t) - f(x_{t+1}) &\geq \gamma_t \langle \nabla f(x_t), x_t - v_t \rangle - \gamma_t^2 \frac{L}{2} \|x_t - v_t\|^2 \\ &\geq \gamma_t (f(x_t) - f(x^*)) - \gamma_t^2 \frac{L}{2} \|x_t - v_t\|^2. \qquad \text{(Lemma 4.1)} \end{aligned}$$

Subtracting $f(x^*)$ on both sides, bounding $\|x_t - v_t\| \leq D$, and rearranging leads to

$$f(x_{t+1}) - f(x^*) \leq (1-\gamma_t)(f(x_t) - f(x^*)) + \gamma_t^2 \frac{LD^2}{2}. \tag{4.2}$$

This contraction relates the primal gap at x_{t+1} with the primal gap at x_t. We conclude the proof by induction. First observe that for $t = 0$ by (4.2) it follows

$$f(x_1) - f(x^*) \leq \frac{LD^2}{2} \leq \frac{2LD^2}{2}.$$

Now consider $t \geq 1$. We have

$$\begin{aligned}
&f(x_{t+1}) - f(x^*) \\
&\quad \leq (1-\gamma_t)(f(x_t) - f(x^*)) + \gamma_t^2 \frac{LD^2}{2} \\
&\quad \leq \frac{t}{2+t}(f(x_t) - f(x^*)) + \frac{4}{(2+t)^2}\frac{LD^2}{2} && \text{(definition of } \gamma_t) \\
&\quad \leq \frac{t}{2+t}\frac{2LD^2}{2+t} + \frac{4}{(2+t)^2}\frac{LD^2}{2} && \text{(induction hypothesis)} \\
&\quad = \frac{2LD^2}{t+3}\left(\frac{(3+t)(1+t)}{(2+t)^2}\right) \leq \frac{2LD^2}{t+3}, && ((3+t)(1+t) \leq (2+t)^2)
\end{aligned}$$

which completes the proof. □

The theorem above provides a convergence guarantee for the primal gap. However, it relies on knowledge of the diameter D and Lipschitz constant L for estimating the number of required iterations to reach a certain target accuracy ε. We can also consider the Frank-Wolfe gap $\max_{v_t \in P} \langle \nabla f(x_t), x_t - v_t \rangle$, which upper bounds the primal gap $f(x_t) - f(x^*)$ via Lemma 4.1. While this gap is not monotonously decreasing (similar to the primal gap in the case of the open loop step-size) it is readily available in each iteration and hence can be used as a stopping criterion, i.e., we stop the algorithm when $\max_{v_\tau \in P} \langle \nabla f(x_\tau), x_\tau - v_\tau \rangle \leq \varepsilon$, not requiring a priori knowledge about D and L. For the running minimum we can establish a convergence rate similar to that in Theorem 4.4; see Jaggi [48], see also Braun et al. [15, Theorem 2.2 and Remark 2.3].

Theorem 4.5 (Frank-Wolfe gap convergence of the Frank-Wolfe algorithm) *Let f be an L-smooth convex function and let P be a compact convex set of diameter D. Consider the iterates of Algorithm 4. Then the running minimum of the Frank–Wolfe gaps up to iteration t satisfies*:

$$\min_{0 \leq \tau \leq t} \max_{v_\tau \in P} \langle \nabla f(x_\tau), x_\tau - v_\tau \rangle \leq \frac{6.75\, LD^2}{t+2}$$

Another important property of the Frank-Wolfe algorithm is that it maintains convex combinations of extreme points and in each iteration at most one new extreme point is added. This leads to a natural accuracy vs. sparsity trade-off, where sparsity broadly refers to having convex combinations with a small number of vertices. This property is very useful and has been exploited repeatedly to prove mathematical results via applying the convergence guarantee of the Frank-Wolfe algorithm; we will see such an example further below in Sect. 5.1

4.2 A Matching Lower Bound

In this section we will now provide a matching lower bound example due to Lan [55], Jaggi [48] that will require $\Omega(LD^2/\varepsilon)$ LMO calls to achieve an accuracy of ε for an L-smooth function f and a feasible region of diameter D. This lower bound holds for *any* algorithm that accesses the feasible region solely through an LMO and shows that in general the convergence rate of the Frank-Wolfe algorithm in Theorem 4.4 cannot be improved. We consider

$$\min_{x \in \operatorname{conv}\{e_1,\dots,e_n\}} \|x\|^2,$$

i.e., we minimize the standard quadratic $f(x) = \|x\|^2$ over the probability simplex $P \doteq \operatorname{conv}\{e_1, \dots, e_n\}$, where the e_i denote the standard basis vectors in $\mathbb{R}^n$, i.e., we have $L = 2$ and $D = \sqrt{2}$ and any other combination of values for L and D can be obtained via rescaling. As f is strongly convex it has a unique optimal solution, which is easily seen to be $x^* = (\frac{1}{n}, \dots, \frac{1}{n})$ with optimal objective function value $f(x^*) = \frac{1}{n}$. Note that the optimal solution lies in the relative interior of P, one of the earliest cases in which improved convergence rates for Frank-Wolfe methods have been obtained (GuéLat and Marcotte [42]).

If we now run the Frank-Wolfe algorithm from any extreme point x_0 of P, then after $t < n$ iterations, we have made t LMO calls, and hence have picked up at most $t+1$ of the n standard basis vectors. This is the only information available to us about the feasible region and by convexity the only feasible points the algorithm can create are convex combinations x_t of these picked up extreme points. Thus it holds

$$f(x_t) \geq \min_{\substack{x \in \operatorname{conv} \mathcal{S} \\ \mathcal{S} \subseteq \{e_1,\dots,e_n\} \\ |\mathcal{S}| \leq t+1}} f(x) = \frac{1}{t+1}.$$

Therefore the primal gap after t iterations satisfies $f(x_t) - f(x^*) \geq 1/(t+1) - 1/n$ and thus with the choice $n \gg 1/\varepsilon$ we need $\Omega(1/\varepsilon)$ LMO calls to guarantee a primal gap of at most ε. Finally, observe that this example also provides an inherent sparsity vs. optimality tradeoff: if we aim for a solution with sparsity t, then the primal gap can be as large as $f(x_t) - f(x^*) \geq 1/(t+1) - 1/n$.

However, several remarks are in order to put this example into perspective. First of all, the lower bound example only holds up to the dimension n of the problem and that is for good reason. Once we pass the dimension threshold, the lower bound is not instructive any more and other step-size strategies might achieve linear rates for $t \geq n$,

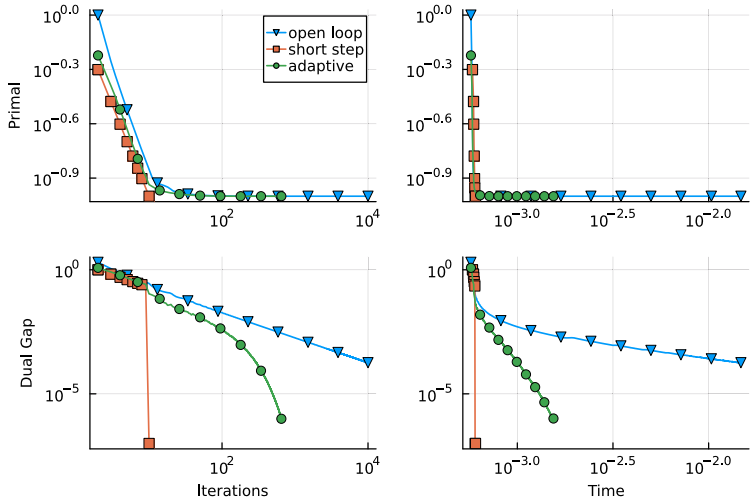

Fig. 7 Minimizing $f(x) = \|x\|^2$ over the probability simplex of dimension $n = 10$ with an iteration limit of $k = 10^4$. It can be seen that once the iteration t crosses the dimension threshold n the short step strategy immediately recovers the optimal solution

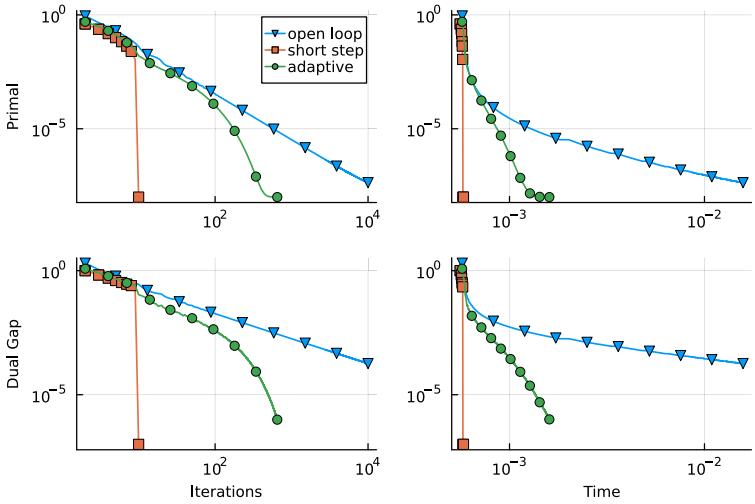

Fig. 8 Same parameters as in Fig. 7, however with the modified objective $f(x) = \|x - (\frac{1}{n}, \ldots, \frac{1}{n})\|^2$. Note, that $(\frac{1}{n}, \ldots, \frac{1}{n})$ is the optimal solution for minimizing $f(x) = \|x\|^2$ over the probability simplex. Dual convergence is identical while primal convergence differs

and in particular if the step-size is the short step rule (see also Sect. 4.5) with exact smoothness L we are optimal after exactly $t = n - 1$ iterations; see Figs. 7 and 8 for computational examples. Moreover, here we considered convergence rates independent of additional problem parameters. Introducing such parameters might provide more granular convergence rates under mild assumptions as shown, e.g., in Garber

[36]. There is also a different lower bound of $\Omega(1/\varepsilon)$ by Wolfe [85] (see also Braun et al. [15, Theorem 2.8]) that is based around the so-called zigzagging phenomenon of the Frank-Wolfe algorithm and that holds beyond the dimension threshold. However, it only holds for step-size strategies—grossly simplifying—that are at least as good as the short step strategy and interestingly the open loop step-size strategy is not subject to this lower bound. This is no coincidence, as there are cases (Wirth et al. [82, 84]) where the open loop step-size can achieve a convergence rate of $\mathcal{O}(1/\varepsilon^2)$ for instances that satisfy the condition of the lower bound of Wolfe [85]. Finally, there is a universal lower bound (Braun et al. [15, Proposition 2.9]) that matches the improved $\mathcal{O}(1/\varepsilon^2)$ rate for the open loop step-size:

Proposition 4.6 *Let f be an L-smooth and convex function over a compact convex set P. Then for $t \geq 1$, the iterates of the Frank-Wolfe algorithm* (*Algorithm 4*) *with any step sizes γ_τ satisfy*

$$f(x_t) - f(x^*) \geq \prod_{\tau=1}^{t-1} (1 - \gamma_\tau) \cdot \langle \nabla f(x^*), x_1 - x^* \rangle, \tag{4.3}$$

and in particular for the open loop step-size rule $\gamma_\tau = 2/(\tau + 2)$ we have

$$f(x_t) - f(x^*) \geq \frac{2}{t(t+1)} \cdot \langle \nabla f(x^*), x_1 - x^* \rangle. \tag{4.4}$$

Finally, in actual computations these lower bounds are rarely an issue as instances often possess additional structure and adaptive step-size strategies (see Sect. 4.5) provide excellent computational performance without requiring any knowledge of problem parameters.

4.3 Nonconvex Objectives

The Frank-Wolfe algorithm can also be used to obtain locally optimal solutions if f is nonconvex but smooth. In this case, $x \in P$ is *locally optimal* (or *first-order critical*) if and only if the Frank-Wolfe gap at x is 0, i.e., $\max_{v \in P} \langle \nabla f(x), x - v \rangle = 0$. We will present a simple argument to establish convergence to a locally optimal solution, however the argument can be improved as done in Lacoste-Julien [52], which was also the first to establish convergence for smooth nonconvex objectives. In particular, our argument will use a constant step-size $\gamma_t = \gamma \doteq \frac{1}{\sqrt{T+1}}$ which has the advantage that it is parameter-free, but we need to decide on the number of iterations T ahead of time and the convergence guarantee only holds for the last iteration T in contrast to so-called *anytime guarantees* that hold in each iteration $t = 0, \ldots, T$. Nonetheless, the core of the argument is identical and more clearly isolated that way.

Theorem 4.7 (Convergence for nonconvex objectives) *Let f be an L-smooth but not necessarily convex function and P be a compact convex set of diameter D. Let $T \in \mathbb{N}$, then the iterates of the Frank-Wolfe algorithm* (*Algorithm 4*) *with the step-size*

$\gamma_t = \gamma \doteq \frac{1}{\sqrt{T+1}}$ *satisfy*:

$$G_T \doteq \min_{0 \le t \le T} \max_{v_t \in P} \langle \nabla f(x_t), x_t - v_t \rangle \le \frac{\max\{2h_0, LD^2\}}{\sqrt{T+1}},$$

where $h_0 \doteq f(x_0) - f(x^*)$ *is the primal gap at* x_0.

Proof Our starting point is the primal progress bound at iterate x_t from Lemma 4.3

$$f(x_t) - f(x_{t+1}) \ge \gamma \langle \nabla f(x_t), x_t - v_t \rangle - \gamma^2 \frac{L}{2} \|x_t - v_t\|^2.$$

Summing up the above along the iterations $t = 0, \ldots, T$ and rearranging gives

$$\begin{aligned}\gamma \sum_{t=0}^{T} \langle \nabla f(x_t), x_t - v_t \rangle &\le f(x_0) - f(x_{T+1}) + \gamma^2 \sum_{t=0}^{T} \frac{L}{2} \|x_t - v_t\|^2 \\ &\le f(x_0) - f(x^*) + \gamma^2 \sum_{t=0}^{T} \frac{LD^2}{2} = h_0 + \gamma^2 (T+1) \frac{LD^2}{2}.\end{aligned}$$

We divide by $\gamma(T+1)$ on both sides to arrive at our final estimation

$$G_T \le \frac{1}{T+1} \sum_{t=0}^{T} \langle \nabla f(x_t), x_t - v_t \rangle \le \frac{h_0}{\gamma(T+1)} + \gamma \frac{LD^2}{2}, \tag{4.5}$$

and for $\gamma = \frac{1}{\sqrt{T+1}}$ this yields

$$G_T \le \frac{1}{T+1} \sum_{t=0}^{T} \langle \nabla f(x_t), x_t - v_t \rangle \le \frac{2h_0 + LD^2}{2\sqrt{T+1}} \le \frac{\max\{2h_0, LD^2\}}{\sqrt{T+1}}, \tag{4.6}$$

which completes the proof. □

Note that G_T can be observed throughout the algorithm's run and can be used as a stopping criterion. Moreover, the convergence rate of $\mathcal{O}(1/\sqrt{T})$ is optimal; see Braun et al. [15] for a discussion. If we have knowledge about h_0, L, and D then the above estimation can be slightly improved while maintaining a constant step size rule. We revisit (4.5) and optimize for γ, to obtain $\gamma = \sqrt{\frac{2h_0}{LD^2(T+1)}}$ and hence:

$$G_T \le \frac{1}{T+1} \sum_{t=0}^{T} \langle \nabla f(x_t), x_t - v_t \rangle \le \sqrt{\frac{2h_0 LD^2}{T+1}} \le \frac{\max\{2h_0, LD^2\}}{\sqrt{T+1}}. \tag{4.7}$$

In the right most estimation the two bounds from (4.6) and (4.7) are identical, which is due to the relatively weak estimation of the very last inequality. In fact, the difference between (4.6) und (4.7) is that in the former we have the arithmetic mean

between $2h_0$ and LD^2 as bound, i.e., $G_T \leq \frac{2h_0+LD^2}{2}\frac{1}{\sqrt{T+1}}$, whereas in (4.7) we have the geometric mean of the two terms, i.e., $G_T \leq \sqrt{2h_0LD^2}\frac{1}{\sqrt{T+1}}$; by the AMGM inequality the latter is smaller than the former. In both cases, we can also turn the guarantees into anytime guarantees (with minor changes in constants) by using the step-size rules $\gamma_t = 1/\sqrt{t+1}$ and $\gamma_t = \sqrt{\frac{2h_0}{LD^2(t+1)}}$, respectively, and then using the bound $\sum_{t=0}^{T-1}\frac{1}{\sqrt{t+1}} \leq 2\sqrt{T}-1$. Then telescoping works analogously to the above with minor adjustments. Finally, note that in all estimation we do not only provide a guarantee for the running minimum of the Frank-Wolfe gap but their averages in fact and the former is a consequence of the latter.

4.4 Dual Prices

Another very useful property of the Frank-Wolfe algorithm (and also its more complex extensions) is that we readily obtain dual prices for active constraints, as long as the LMO provides dual prices. Similar to linear optimization, the dual price of a constraint captures the (local) rate of change of the objective if the constraint is relaxed. This is in particular useful in, e.g., portfolio optimization applications and energy problems, where marginal prices of constraints can guide decisions of real-world decision makers. Here we will only consider dual prices at the optimal solution x^* and we will only cover the basic case without any degeneracy. However we can also derive dual prices for approximately optimal solutions and we refer the interested reader to Braun and Pokutta [11] for an in-depth discussion.

Suppose that the feasible region P is actually a polytope of the form $P = \{z : Az \leq b\}$ with $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^n$. Let $x \in P$ be arbitrary. By strong duality we have that $v \in P$ is a minimizer for the linear program $\min_{\{z:Az\leq b\}}\langle \nabla f(x), z\rangle$ if and only if there exist dual prices $0 \leq \lambda \in \mathbb{R}^m$, so that

$$\nabla f(x) = -\lambda A \qquad \text{and} \qquad \langle \nabla f(x), v\rangle = \min_{\{z:Az\leq b\}}\langle \nabla f(x), z\rangle = -\langle \lambda, b\rangle, \tag{LP-duality}$$

i.e., the dual prices together with the constraints certify optimality. It is well known that the second equation can be equivalently replaced by a complementary slackness condition that states $\langle \lambda, b - Av\rangle = 0$; it can be readily seen that (LP-duality) implies $\langle \lambda, b - Av\rangle = 0$ by rearranging and the other direction follows similarly. Now consider a primal-dual pair (v, λ) that satisfies (LP-duality). By definition v is also a Frank-Wolfe vertex for $\nabla f(x)$, so that we immediately obtain

$$\langle \nabla f(x), x - v\rangle = -\langle \lambda A, x - v\rangle = \langle \lambda, b - Ax\rangle,$$

i.e., the *Frank-Wolfe gap at x is equal to the complementarity gap for x given λ*; if the latter would be 0 then complementary slackness would hold or equivalently the Frank-Wolfe gap would be 0 and (x, λ) would be an optimal primal-dual pair. This can be now directly be related to $\min_{\{z:Az\leq b\}} f(z)$ via *Slater's (strong duality) condition of optimality*: x is optimal for $\min_{\{z:Az\leq b\}} f(z)$ if and only if x is optimal for $\min_{\{z:Az\leq b\}}\langle \nabla f(x), z\rangle$. This implies that if x is an optimal solution to

$\min_{\{z:Az\leq b\}} f(z)$ then (x, λ) will also satisfy (LP-duality). Hence for an optimal solution x, the dual prices λ valid for v are also valid for x.

Given that the LMO for polytopes is often realized via linear programming solvers that compute dual prices as by-product, we readily obtain dual prices λ for the optimal solution x^* via the Frank–Wolfe vertex v for $\nabla f(x^*)$.

4.5 Adaptive Step-Sizes

The primal progress of a Frank-Wolfe step is driven by the smoothness inequality. Suppose f is L-smooth, then using the definition of the Frank-Wolfe step, i.e., $x_{t+1} = (1-\gamma_t)x_t + \gamma_t v_t$ and Lemma 4.3 provides:

$$f(x_t) - f(x_{t+1}) \geq \gamma_t \langle \nabla f(x_t), x_t - v_t \rangle - \gamma_t^2 \frac{L}{2} \|x_t - v_t\|^2. \tag{4.8}$$

Now rather than plugging in the open loop step-size, we can view the right-hand side as an expression in one variable γ_t and maximize the right-hand side. This leads to the optimal choice

$$\gamma_t = \frac{\langle \nabla f(x_t), x_t - v_t \rangle}{L\|x_t - v_t\|^2} \qquad \text{and} \qquad f(x_t) - f(x_{t+1}) \geq \frac{\langle \nabla f(x_t), x_t - v_t \rangle^2}{2L\|x_t - v_t\|^2}. \tag{4.9}$$

Technically we can only form convex combinations if $\gamma_t \in [0, 1]$, so that we have to truncate $\gamma_t := \min\left\{\frac{\langle \nabla f(x_t), x_t - v_t\rangle}{L\|x_t - v_t\|^2}, 1\right\}$; observe that $\gamma_t \geq 0$ holds always as the we have that the Frank-Wolfe gap $\langle \nabla f(x_t), x_t - v_t \rangle \geq 0$. This step-size choice is often referred to as *short step step-size* and is the Frank-Wolfe equivalent to steepest descent. In the case that the truncation is active, it holds that $\langle \nabla f(x_t), x_t - v_t \rangle \geq L\|x_t - v_t\|^2$ and together with (4.8) it follows that we are in a regime where we converge linearly with

$$f(x_t) - f(x_{t+1}) \geq \langle \nabla f(x_t), x_t - v_t \rangle / 2 \geq (f(x_t) - f(x^*))/2,$$

i.e., the primal progress is at least half of the Frank-Wolfe gap and hence at least half of the primal gap.

The short step strategy avoids the overhead of line searches, however unfortunately it requires knowledge of the smoothness constant L or at least reasonably tight upper bounds of such. This issue is what Pedregosa et al. [69] addressed in a very nice paper by dynamically approximating L. Rather than performing a traditional line search on the function value, the approximation of L leads only to a slightly slower convergence rate by a constant factor, has only small overhead, and does not suffer the additive resolution issue of traditional line searches, where one can only get as accurate as the line search ε. In particular, this adaptive strategy allows to adapt to the potentially better local smoothness of f, rather than relying on a worst-case estimate; see Braun et al. [15] for an in-depth discussion.

In a nutshell, what Pedregosa et al. [69] do is perform a multiplicative search for L until the smoothness inequality

$$f(x_t) - f(x_{t+1}) \geq \gamma_t \langle \nabla f(x_t), x_t - v_t \rangle - \gamma_t^2 \frac{M}{2} \|x_t - v_t\|^2. \tag{adaptive}$$

Algorithmus 5 : (modified) Adaptive step-size strategy

Input: Objective function f, smoothness estimate $\widetilde{L}$, feasible points x, v with $\langle \nabla f(x), x - v \rangle \geq 0$, progress parameters $\eta \leq 1 < \tau$
Output: Updated estimate $\widetilde{L}^*$, step-size γ

```
M ← ηL̃
loop
    γ ← min{⟨∇f(x), x − v⟩ / (M‖x − v‖²), 1}        {compute short step for estimation M}
    if ⟨∇f(x + γ(v − x)), x − v⟩ ≥ 0 then
        L̃* ← M
        return L̃*, γ
    end if
    M ← τM
end loop
```

holds for the approximation M of L with $\gamma_t = \min\left\{\frac{\langle \nabla f(x_t), x_t - v_t\rangle}{M\|x_t - v_t\|^2}, 1\right\}$ being the short step.

Unfortunately, checking (adaptive) in practice can be numerically very challenging as we mix function evaluations, gradient evaluations, and quadratic norm terms. Rather we present a new variant of the adaptive step-size strategy, where we rely on a different test for accepting the estimation M of L:

$$\langle \nabla f(x_{t+1}), x_t - v_t \rangle \geq 0, \tag{altAdaptive}$$

where $x_{t+1} = (1 - \gamma_t)x_t + \gamma_t v_t$ as before with $\gamma_t = \min\left\{\frac{\langle \nabla f(x_t), x_t - v_t\rangle}{M\|x_t - v_t\|^2}, 1\right\}$ being the short step for the estimation M, i.e., we only test (inner products with) the gradient ∇f at different points. Moreover, this test might provide additional primal progress as we discuss below. This leads to the adaptive step-size strategy given in Algorithm 5, which is numerically very stable, however requires gradient computations (rather than function evaluations).

We first show now that our condition (altAdaptive) implies the same primal progress as (adaptive) and then we will show that (altAdaptive) holds for L if f is L-smooth. As such all results of Pedregosa et al. [69] apply readily to the modified variant in Algorithm 5. To demonstrate the convergence behavior of the various step-size strategies we ran a simple test problem with results presented in Fig. 9. We see that the adaptive strategy approximates the short step very well and both significantly outperform the open loop strategy.

In the following we present the slightly more involved estimation based on a new progress bound from smoothness. For completeness we also include a significantly simplified estimation based on the regular smoothness bound in Appendix A, however there we only guarantee approximation of the smoothness constant within a factor of 2. We start with introducing another variant of the smoothness inequality. Note that all these inequalities are equivalent when considering all x, y, however we want

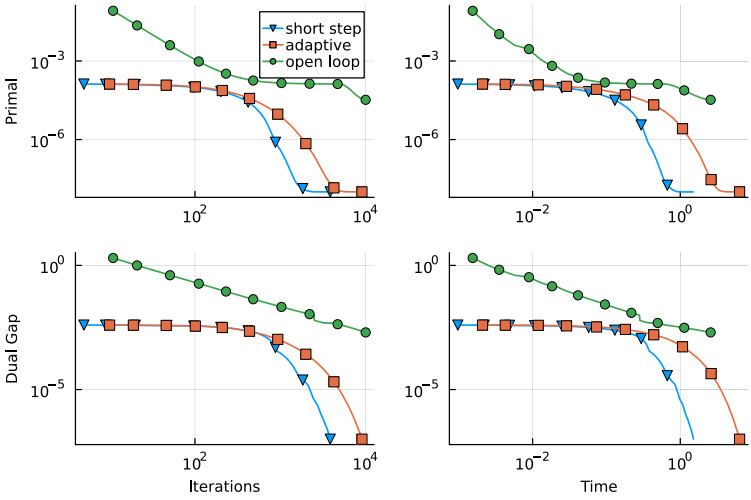

Fig. 9 Convergence speed for a simple quadratic over a K-sparse polytope with three different step-size strategies. The basic open loop step-size $\gamma_t = \frac{2}{2+t}$, the short step rule, which requires a smoothness estimate L (here we used the exact smoothness), and the adaptive step-size rule that dynamically approximates L. Plot is log-log so that the order of the convergence corresponds to different slopes of the trajectories

to apply them for a *specific* pair of points x, y and then their transformations from one into another might not be sharp as demonstrated in the following remark:

Remark 4.8 (Point-wise smoothness estimations) Suppose that f is L-smooth and convex and consider two points x, y. Suppose we want to derive (2.3) from the gradient-based variant in (2.4) using only the two points x, y. Then the naive way of doing so it:

$$\begin{aligned} f(y) - f(x) &\leq \langle \nabla f(y), y - x \rangle && \text{(convexity)} \\ &\leq \langle \nabla f(x), y - x \rangle + L\|y - x\|^2. && \text{(using (2.4))} \end{aligned}$$

Observe that this is *almost* the desired inequality (2.3), except for the smoothness constant $2L$ and not L.

The following lemma provides a different smoothness inequality that allows for tighter estimations. It requires f to be L-smooth and convex on a potentially slightly larger domain containing P.

Lemma 4.9 (Smoothness revisited) *Let f be an L-smooth and convex function on the D-neighborhood of a compact convex set P, where D is the diameter of P. Then for all $x, y \in P$ it holds*:

$$\frac{\langle \nabla f(y) - \nabla f(x), y - x \rangle^2}{2L\|y - x\|^2} \leq f(y) - f(x) - \langle \nabla f(x), y - x \rangle. \tag{4.10}$$

Proof As shown in Braun et al. [15, Lemma 1.8], if f is an L-smooth convex function on the D-neighborhood of a convex set P, then for any points $x, y \in P$ it holds

$$\|\nabla f(y) - \nabla f(x)\|^2 \leq 2L(f(y) - f(x) - \langle \nabla f(x), y - x\rangle). \tag{4.11}$$

Next we lower bound the left-hand side as

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

Chaining these two inequalities together and rearranging gives the desired claim. □

The proof above explicitly relies on the convexity of f via Braun et al. [15, Lemma 1.8]. With Lemma 4.9 we can provide the following guarantee on the primal progress.

Lemma 4.10 (Primal progress from (altAdaptive)) *Let f be an L-smooth and convex function on the D-neighborhood of a compact convex set P, where D is the diameter of P. Further, let $x_{t+1} = (1-\gamma_t)x_t + \gamma_t v_t$ with $\gamma_t = \min\left\{\frac{\langle \nabla f(x_t), x_t - v_t\rangle}{M\|x_t - v_t\|^2}, 1\right\}$ for some M. If $\langle \nabla f(x_{t+1}), x_t - v_t\rangle \geq 0$, then it holds:*

$$f(x_t) - f(x_{t+1}) \geq \begin{cases} \frac{\langle \nabla f(x_t), x_t - v_t\rangle^2 + \langle \nabla f(x_{t+1}), x_t - v_t\rangle^2}{2\max\{L, M\}\|x_t - v_t\|^2} & \text{if} \quad \gamma_t \in [0,1] \\ \frac{\langle \nabla f(x_t), x_t - v_t\rangle}{2} + \frac{\langle \nabla f(x_{t+1}), x_t - v_t\rangle^2}{2\langle \nabla f(x_t), x_t - v_t\rangle} \geq \frac{\langle \nabla f(x_t), x_t - v_t\rangle}{2} & \text{if} \quad \gamma_t = 1 \text{ and } M \geq L \end{cases}.$$

Proof If $\gamma_t = 1$, then without loss of generality we can assume that $M = \frac{\langle \nabla f(x_t), x_t - v_t\rangle}{\|x_t - v_t\|^2}$, as M only occurs in the definition of γ_t and x_{t+1}. Our starting point is Equation (4.10) with $x \leftarrow x_{t+1}$ and $y \leftarrow x_t$:

$$\begin{aligned} f(x_t) - f(x_{t+1}) &\geq \frac{\langle \nabla f(x_t) - \nabla f(x_{t+1}), x_t - x_{t+1}\rangle^2}{2L\|x_t - x_{t+1}\|^2} \\ &\quad + \langle \nabla f(x_{t+1}), x_t - x_{t+1}\rangle \\ &= \frac{\langle \nabla f(x_t) - \nabla f(x_{t+1}), x_t - v_t\rangle^2}{2L\|x_t - v_t\|^2} \\ &\quad + \frac{\langle \nabla f(x_t), x_t - v_t\rangle \cdot \langle \nabla f(x_{t+1}), x_t - v_t\rangle}{M\|x_t - v_t\|^2} && \text{(definition of } x_{t+1} \text{ and } \gamma_t) \\ &\geq \frac{\langle \nabla f(x_t) - \nabla f(x_{t+1}), x_t - v_t\rangle^2}{2\max\{L, M\}\|x_t - v_t\|^2} \\ &\quad + \frac{\langle \nabla f(x_t), x_t - v_t\rangle \cdot \langle \nabla f(x_{t+1}), x_t - v_t\rangle}{\max\{L, M\}\|x_t - v_t\|^2} && (\langle \nabla f(x_{t+1}), x_t - v_t\rangle \geq 0) \\ &= \frac{\langle \nabla f(x_t), x_t - v_t\rangle^2 + \langle \nabla f(x_{t+1}), x_t - v_t\rangle^2}{2\max\{L, M\}\|x_t - v_t\|^2}. \end{aligned}$$

Now if $\gamma_t = 1$ and $M \geq L$, then the above simplifies to:

$$f(x_t) - f(x_{t+1}) \geq \frac{\langle \nabla f(x_t), x_t - v_t \rangle}{2} + \frac{\langle \nabla f(x_{t+1}), x_t - v_t \rangle^2}{2\langle \nabla f(x_t), x_t - v_t \rangle} \geq \frac{\langle \nabla f(x_t), x_t - v_t \rangle}{2}.$$

This finishes the proof. □

Before we continue a few remarks are in order. First of all, observe that

$$f(x_t) - f(x_{t+1}) \geq \frac{\langle \nabla f(x_t), x_t - v_t \rangle^2 + \langle \nabla f(x_{t+1}), x_t - v_t \rangle^2}{2 \max\{L, M\} \|x_t - v_t\|^2},$$

has an additional term compared to the standard smoothness estimation (4.9) and $\langle \nabla f(x_{t+1}), x_t - v_t \rangle = 0$ if and only if x_{t+1} is identical to the line search solution. This is in particular the case if f is a standard quadratic as then the line search solution is identical to the short step solution. Nonetheless, in the typical case this extra term provides additional primal progress. Taking the maximum in the denominator ensures that if $M < L$, then we recover the primal progress that one would have obtained with the estimation $M = L$. This seems counter-intuitive as usually using estimations $M < L$ would lead to overshooting and negative primal progress, however here we still require that (altAdaptive) holds for M, which prevents exactly this as can be seen from the proof. In particular, disregarding adaptivity for a second, in the case where L is known, then with the choice $M = L$, Lemma 4.11 provides a stronger primal progress bound compared to (4.9) assuming that (altAdaptive) holds for L (which holds always as f is L-smooth; see Lemma 4.11):

$$f(x_t) - f(x_{t+1}) \geq \frac{\langle \nabla f(x_t), x_t - v_t \rangle^2 + \langle \nabla f(x_{t+1}), x_t - v_t \rangle^2}{2L \|x_t - v_t\|^2}.$$

This improved primal progress bound might give rise to improved convergence rates in some regimes; see also Teboulle and Vaisbourd [76] for a similar analysis for the unconstrained case providing optimal constants for the convergence rates of gradient descent. Moreover, the discussion above also implies that if (altAdaptive) holds it might provide more primal progress than the original test via (adaptive) used in Pedregosa et al. [69].

To conclude, we will now show that (altAdaptive) holds for L, whenever the function is L-smooth and γ_t is the corresponding short step for L. This implies that both (altAdaptive) and (adaptive) hold for L whenever f is L-smooth with the added benefit of numerical stability and additional primal progress via (altAdaptive).

Lemma 4.11 (Smoothness implies (altAdaptive)) *Let f be L-smooth. Further, let $x_{t+1} = (1 - \gamma_t)x_t + \gamma_t v_t$ with $\gamma_t = \min\left\{\frac{\langle \nabla f(x_t), x_t - v_t \rangle}{L\|x_t - v_t\|^2}, 1\right\}$. Then* (altAdaptive) *holds, i.e.,*

$$\langle \nabla f(x_{t+1}), x_t - v_t \rangle \geq 0.$$

Proof We use the alternative definition of smoothness using the gradients (2.4), i.e., we have

$$\langle \nabla f(y) - \nabla f(x), y - x \rangle \leq L \|y - x\|^2 \quad \text{for all } x, y \in P.$$

Now plug in $x \leftarrow x_t$ and $y \leftarrow x_{t+1}$, so that we obtain

$$\langle \nabla f(x_{t+1}) - \nabla f(x_t), x_{t+1} - x_t \rangle \leq L \|x_{t+1} - x_t\|^2$$

and using the definition of x_{t+1} it follows

$$\langle \nabla f(x_{t+1}) - \nabla f(x_t), \gamma_t(v_t - x_t) \rangle \leq L\gamma_t^2 \|v_t - x_t\|^2.$$

Now, if $\gamma_t = 1$, then $\langle \nabla f(x_t), x_t - v_t \rangle \geq L\|x_t - v_t\|^2$, so that

$$\langle \nabla f(x_{t+1}) - \nabla f(x_t), v_t - x_t \rangle \leq \langle \nabla f(x_t), x_t - v_t \rangle,$$

holds. Otherwise, if $0 < \gamma_t < 1$, then dividing by γ_t and plugging in the definition of γ_t yields

$$\langle \nabla f(x_{t+1}) - \nabla f(x_t), v_t - x_t \rangle \leq \langle \nabla f(x_t), x_t - v_t \rangle.$$

In both cases, rearranging gives the desired inequality

$$\langle \nabla f(x_{t+1}), x_t - v_t \rangle \geq 0.$$

Finally, in case $\gamma_t = 0$ we have $x_t = x_{t+1}$ and the assertion holds trivially. □

Remark 4.12 (Faster open loop convergence) The adaptive step-size strategy from above uses feedback from the function and as such is not of the open loop type. In many applications such adaptive strategies are the strategies of choice as the function feedback is relatively minimal but convergence speed is superior (in most but not all cases as mentioned in Sect. 4.2). For many important cases we can also obtain improved convergence with rates of higher order for open loop step-sizes by using the modified step-size rule $\gamma_t = \frac{\ell}{t+\ell}$ with $\ell \in \mathbb{N}_{\geq 1}$; see Wirth et al. [82, 84] for details. This is quite surprising as we only change the shift ℓ and not the order of t in the denominator of γ_t. In fact, note that the order of t in the denominator cannot be changed significantly as we need that $\sum_t \gamma_t = \infty$ and that $\sum_t \gamma_t^2$ converges for the step-size strategy to work; see Braun et al. [15]. If the corresponding ℓ cannot be set in practice, and if it has to be an open loop strategy, $\gamma_t = \frac{2+\log(t+1)}{t+2+\log(t+1)}$ works very well; we can use t or $t+1$ in the log depending on whether the first iteration is $t = 0$ or $t = 1$. This corresponds essentially to a strategy where ℓ is gradually increased and it provides accelerated convergence rates when those exist while maintaining the same worst-case convergence rates as the basic strategy $\gamma_t = \frac{2}{t+2}$ (Wirth et al. [83]). For a sample computation, see Fig. 10.

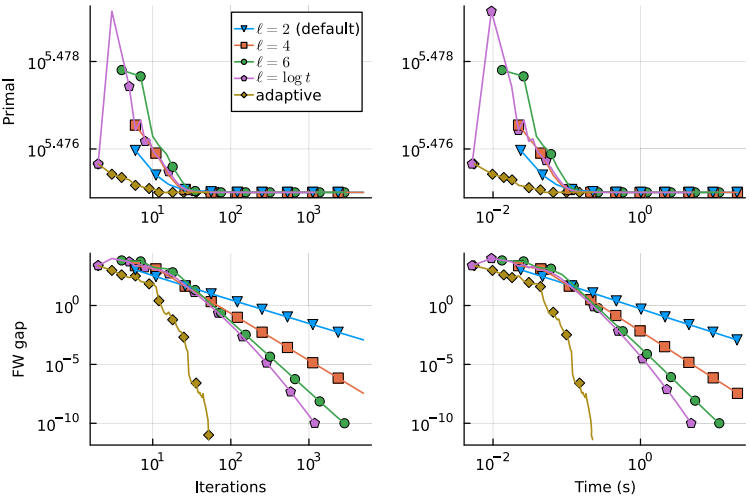

Fig. 10 Convergence speed for a simple quadratic over a K-sparse polytope for open loop strategies of the form $\gamma_t = \frac{\ell}{\ell+t}$. We can see that (depending on the specifics of the problem) larger ℓ achieve convergence rates of a higher order. For comparison also the adaptive step-size strategy has been included. Plot is log-log so that the order of the convergence corresponds to different slopes of the trajectories

5 Two Applications

In the following we will present two applications of the Frank-Wolfe algorithm. Both examples use very simple quadratic objectives of the form $f(x) = \|x - p\|^2$ for some p for the sake of exposition; for more complex examples see also Braun et al. [15].

5.1 The Approximate Carathéodory Problem

Our first example, is the *Approximate Carathéodory Problem*. For this example, the Frank-Wolfe algorithm does not only present a practical means to solve the problem but in fact, its convergence guarantees itself provide a proof of the theorem and optimal bounds for a wide variety of regimes. Here we will confine ourselves to the 2-norm case not assuming any additional properties, however many more involved cases are possible as studied in Combettes and Pokutta [25].

Given a compact convex set $P \subseteq \mathbb{R}^n$, recall that Carathéodory's theorem states that any $x^* \in P$ can be written as a convex combination of no more than $n+1$ extreme points of P, i.e., $x^* = \sum_{1 \le i \le n+1} \lambda_i v_i$ with $\lambda \ge 0$, $\sum_i \lambda_i = 1$, and v_i extreme points of P with $1 \le i \le n+1$. In the context of Carathéodory's theorem, the *cardinality* of a point $x^* \in P$, refers to the minimum number of required extreme points to express x^* as a convex combination of those. If x^* is of low cardinality it is often also referred to as *sparse*. Every specific convex combination that expresses x^* provides an upper bound on the cardinality of x^*. The approximate variant of Carathéodory's problem asks: given $x^* \in P$, what is required cardinality of an $x \in P$ to approximate x^* within an error of no more than $\varepsilon > 0$ (in a given norm)? Put differently, we are looking for $x \in P$ with $\|x - x^*\| \le \varepsilon$ of low cardinality. The approximate Carathéodory theorem states:

Theorem 5.1 (Approximate Carathéodory Theorem) *Let $p \geq 2$ and P be a compact convex set. For every $x^* \in P$, there exists $x \in P$ with cardinality of no more than $\mathcal{O}(pD_p^2/\epsilon^2)$ satisfying $\|x - x^*\|_p \leq \epsilon$, where $D_p = \max_{v,w \in P} \|w - v\|_p$ is the p-norm diameter of P.*

Note that the bounds of Theorem 5.1 are essentially tight in many cases (Mirrokni et al. [59]). In the following, we briefly discuss the case $p = 2$ without any additional assumptions. Suppose we have given a point $x^* \in P$ we can consider the objective

$$f(x) \doteq \|x - x^*\|^2.$$

Further, let $\varepsilon > 0$ be the approximation guarantee. Assuming, we have access to an LMO for P, we can now minimize the function $f(x)$ over P via the Frank-Wolfe algorithm (Algorithm 4). In order to achieve $\|x - x^*\| \leq \varepsilon$ we have to run the Frank-Wolfe algorithm until $f(x_t) = f(x_t) - f(x^*) \leq \varepsilon^2$, which by Theorem 4.4 takes no more than $\mathcal{O}(2D^2/\epsilon^2)$ iterations, where D is the ℓ_2-diameter of P. Moreover, in each iteration the algorithm is picking up at most one extreme point as discussed in Sect. 4.1. This establishes the guarantee for case $p = 2$ in Theorem 5.1. Here we applied the basic convergence guarantee from Theorem 4.4. However, for the Frank-Wolfe algorithm many more convergence guarantees are known, depending on properties of the feasible domain and position of the point x^* that we want to approximate with a sparse convex combination. These improved convergence rates immediately translate into improved approximation guarantees for the approximate Carathéodory problem and we state some of these guarantees in Table 1.

Apart from establishing theoretical results by means of the Frank-Wolfe algorithm's convergence guarantees it can be easily used in actual computations, see e.g., Figs. 11 and 12 for an example. Observe that in the particular case of $f(x)$ here, we can also directly observe the primal gap and hence use it as a stopping criterion. The Frank-Wolfe approach to the approximate Carathéodory Problem has been also recently used in Quantum Mechanics to establish new Bell inequalities and local models, as well as improve the Grothendieck constant $K_G(3)$ of order 3 (see Designolle et al. [30, 31] and the references contained therein). This approach is also very useful in the context of the *coreset problem*, which asks for a subset of data points of a large data set that maintains approximately the same statistical properties (see Braun et al. [15, Sect. 5.2.5]).

5.2 Separating Hyperplanes

In Sect. 5.1 we have used the Frank-Wolfe algorithm for obtaining a convex decomposition of a point $x^* \in P$, i.e., we certified membership in P. We can also use the Frank-Wolfe algorithm with the same objective $f(x) = \|x - \tilde{x}\|^2$ to obtain separating hyperplanes for points $\tilde{x} \notin P$, i.e., we certify non-membership. This has been successfully applied in Designolle et al. [30, 31] to certify that the correlations of certain quantum states exhibit non-locality, i.e., are truly quantum by separating them from the local polytope, the polytope of all classical correlations. Moreover, it has also been used in Thuerck et al. [77] to derive separating hyperplanes from enumeration oracles.

Table 1 Cardinality bounds to achieve ϵ-approximation for the approximate Carathéodory problem with respect to the ℓ_p-norm; see Combettes and Pokutta [25] for full table. Recall that P is (α, q)-*uniformly convex* if for any $x, y \in P$, $\gamma \in [0,1]$, and $z \in \mathbb{R}^n$ with $\|z\| \leq 1$ we have $y + \gamma(x-y) + \gamma(1-\gamma) \cdot \alpha \|x-y\|^q z \in P$, where α and q are positive. An $(\alpha/2, 2)$-uniformly convex set is called α-*strongly convex*

ℓ_p-norm	Assumption	Cardinality bound
$p \in [2, +\infty[$	–	$\mathcal{O}\left(\frac{pD_p^2}{\epsilon^2}\right)$ or $\mathcal{O}\left(\frac{p(D_*^2+D_0^2)}{\epsilon^2}\right)$
		(D_p, D_0, D_* diameters; see Combettes and Pokutta [25])
	$x^* \in \mathrm{ri}(P)$	$\mathcal{O}\left(p\left(\frac{D_p}{r_p}\right)^2 \ln\left(\frac{1}{\epsilon}\right)\right)$
		($\mathrm{ri}(P)$ relative interior, r_p radius so that $B_{r_p}^p(x^*) \cap \mathrm{aff}(P) \subseteq P$)
	α_p-strongly convex P	$\mathcal{O}\left(\frac{\sqrt{p}D_p + p/\alpha_p}{\epsilon}\right)$
	(α_p, q_p)-uniformly convex P, $q_p \in [2, +\infty[$	$\mathcal{O}\left(\frac{(pD_p^2)^{(q_p-1)/q_p} + p/\alpha_p^{2/q_p}}{\epsilon^{2(q_p-1)/q_p}}\right)$
$p \in]1, 2[$	–	$\mathcal{O}\left(\frac{n^{(2-p)/p} D_2^2}{\epsilon^2}\right)$
$p = 1$	–	$\mathcal{O}\left(\frac{nD_2^2}{\epsilon^2}\right)$
		(n ambient dimension of P, D_2 is ℓ_2-diameter)
$p = +\infty$	–	$\mathcal{O}\left(\frac{D_2^2}{\epsilon^2}\right)$
		(D_2 is ℓ_2-diameter)

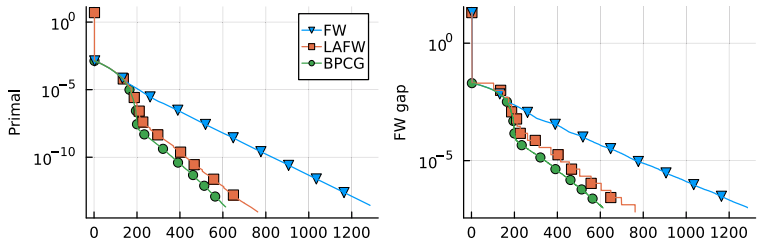

Fig. 11 Cardinality vs. approximation error in ℓ_2-norm over a polytope of dimension $n = 1000$ for the Frank-Wolfe algorithm and two more advanced variants *Lazy Away-step Frank-Wolfe* [14] and *Blended Pairwise Conditional Gradients* [78]. All algorithms use the short step step-size

In the following we provide the most naive way of computing separating hyperplanes. An improved strategy has been presented in Thuerck et al. [77], which derives a new algorithmic characterization of non-membership, that requires fewer iterations of the Frank-Wolfe algorithm compared to our naive strategy here. It is also interesting to note that from a complexity-theoretic perspective, what the Frank-Wolfe algorithm does is to turn an LMO for P into a separation oracle for P via optimizing the objective $\|x - \tilde{x}\|^2$.

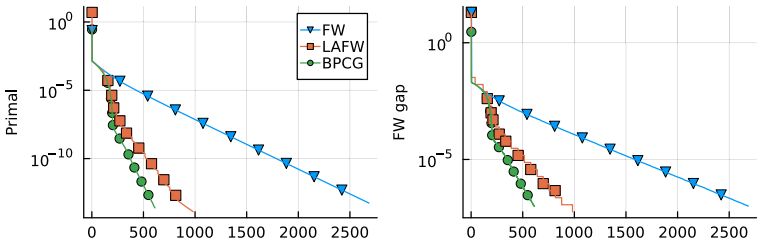

Fig. 12 Same setup as in Fig. 11, however the step-size strategy is the adaptive strategy from Sect. 4.5. As we can see the Frank-Wolfe algorithm is quite sensitive to the strategy while more advanced variants, due to their design of only adding new vertices when not enough progress can be made otherwise, are not

Given $\tilde{x} \notin P$, we consider the optimization problem

$$\min_{x \in P} \|x - \tilde{x}\|^2, \tag{Sep}$$

with $f(x) \doteq \|x - \tilde{x}\|^2$. Using Lemma 4.2 we can immediately obtain a separating hyperplane from an optimal solution $x^* \in P$ to (Sep):

$$\langle \nabla f(x^*), x^* \rangle \leq \langle \nabla f(x^*), v \rangle, \tag{sepHyperplane}$$

which holds for all $v \in P$. Moreover, as $\tilde{x} \notin P$, we have by convexity $\langle \nabla f(x^*), x^* - \tilde{x} \rangle \geq f(x^*) - f(\tilde{x}) = f(x^*) > 0$. and hence (sepHyperplane) is violated by $\tilde{x}$, i.e., $\langle \nabla f(x^*), x^* \rangle > \langle \nabla f(x^*), \tilde{x} \rangle$. This argument provides the desired separating hyperplane mathematically, but numerically it is problematic as we usually solve Problem (Sep) only up to some accuracy $\varepsilon > 0$, typically using the Frank-Wolfe gap $\max_{v \in P} \langle \nabla f(x_t), x_t - v \rangle \leq \varepsilon$ as stopping criterion. When the algorithm stops we similarly obtain

$$\langle \nabla f(x_t), x_t \rangle - \varepsilon \leq \langle \nabla f(x_t), x \rangle \quad \text{which simplifies to}$$

$$\min_{v \in P} \langle \nabla f(x_t), v \rangle \leq \langle \nabla f(x_t), x \rangle, \tag{validIneq}$$

which is valid for all $x \in P$. However this inequality does not necessarily separate $\tilde{x}$ from P. A sufficient condition for separation is that $\tilde{x}$ is $\sqrt{\varepsilon}$-far from P so that we have $\|x^* - \tilde{x}\| > \sqrt{\varepsilon}$. We then can use the same convexity argument as before:

$$\begin{aligned} &\langle \nabla f(x_t), x_t - \tilde{x} \rangle - \varepsilon && \text{(stopping criterion)} \\ &\geq f(x_t) - f(\tilde{x}) - \varepsilon && \text{(convexity)} \\ &\geq f(x^*) - \varepsilon > 0. && (\|x^* - \tilde{x}\| > \sqrt{\varepsilon}) \end{aligned}$$

Now turning this argument around, if $\tilde{x} \notin P$ is ε-far from P, we need to run the Frank-Wolfe algorithm until the Frank-Wolfe gap satisfies $\max_{v \in P} \langle \nabla f(x_t), x_t - v \rangle \leq \varepsilon^2$. Combining this with Theorem 4.5 we can estimate

$$\frac{6.75 L D^2}{t+2} \leq \varepsilon^2,$$

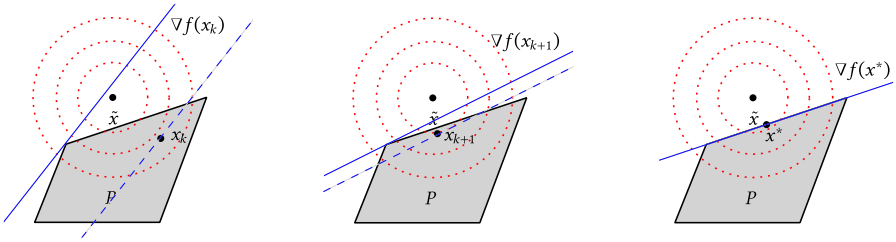

Fig. 13 Each iterate x_k induces a valid inequality of the form $\min_{v\in P}\langle \nabla f(x_t), v\rangle \leq \langle \nabla f(x_t), x\rangle$ in blue (dashed blue line is $\nabla f(x_t)$ at x_t), which may or may not separate $\tilde{x}$; see middle and left respectively. At x^* the induced inequality $\min_{v\in P}\langle \nabla f(x^*), v\rangle = \langle \nabla f(x^*), x^*\rangle \leq \langle \nabla f(x^*), x\rangle$ is guaranteed to separate $\tilde{x} \notin P$ and often (but not always) induces a facet of P

with $L = 2$. Thus we have found a separating hyperplane for $\tilde{x}$ whenever $t \geq 13.5D^2/\varepsilon^2$.

In practice however we usually do not know D and we also do not know how far $\tilde{x}$ is from P. Nonetheless, we can simply test in each iteration t whether $\tilde{x}$ violates (validIneq), i.e.,

$$\nabla f(x_t) \text{ separates } \tilde{x} \quad \Leftrightarrow \quad \min_{v\in P}\langle \nabla f(x_t), v\rangle > \langle \nabla f(x_t), \tilde{x}\rangle.$$

and simply stop then and are guaranteed this will take no more than $\mathcal{O}(D^2/\varepsilon^2)$ iterations. The process is illustrated in Fig. 13. A similar approach, basically combining Sects. 5.1 and 5.2 can also be used to compute the intersection of two compact convex sets or certify their disjointness by means of a separating hyperplane (assuming LMO access to each) as shown in Braun et al. [16].

6 Computational Codes

For actual computations we have developed the `FrankWolfe.jl` Julia package, which implements many Frank-Wolfe variants and is highly customizable. Moreover, we have also developed a mixed-integer extension `Boscia.jl` that allows for some of the variables taking discrete values.

Appendix A: Adaptive Step-Sizes: Simpler Estimation

In this section we will present a simplified estimation of Sect. 4.5 for adaptive step-sizes albeit at the cost being only able to approximate the smoothness of f within a factor of 2. The basic setup is identical to the one before, however we use a different test for accepting the estimation M of L:

$$\langle \nabla f(x_{t+1}), x_t - v_t\rangle \geq \frac{1}{2}\langle \nabla f(x_t), x_t - v_t\rangle, \qquad \text{(altAdaptive-simple)}$$

Algorithmus 6 : (modified) Adaptive step-size strategy – simple variant

Input: Objective function f, smoothness estimate $\widetilde{L}$, feasible points x, v with $\langle \nabla f(x), x - v \rangle \geq 0$, progress parameters $\eta \leq 1 < \tau$

Output: Updated estimate $\widetilde{L}^*$, step-size γ

```
M ← ηL̃
loop
    γ ← min{⟨∇f(x), x − v⟩ / (M‖x − v‖²), 1}        {compute short step for estimation M}
    if ⟨∇f(x + γ(v − x)), x − v⟩ ≥ ½⟨∇f(x), x − v⟩ then
        L̃* ← M
        return L̃*, γ
    end if
    M ← τM
end loop
```

where $x_{t+1} = (1 - \gamma_t)x_t + \gamma_t v_t$ as before with $\gamma_t = \min\left\{\frac{\langle \nabla f(x_t), x_t - v_t\rangle}{M\|x_t - v_t\|^2}, 1\right\}$ being the short step for the estimation M and the corresponding algorithm becomes Algorithm 6 in this case. We proceed similarly as before: we first show that condition (altAdaptive-simple) implies primal progress and then we will show that (altAdaptive-simple) holds for $2L$ if f is L-smooth; this is were (altAdaptive-simple) is weaker than (altAdaptive).

Lemma 7.1 (Primal progress from (altAdaptive-simple)) *Let* $x_{t+1} = (1-\gamma_t)x_t + \gamma_t v_t$ *with* $\gamma_t = \min\left\{\frac{\langle \nabla f(x_t), x_t - v_t\rangle}{M\|x_t - v_t\|^2}, 1\right\}$ *for some* M. *If* $\langle \nabla f(x_{t+1}), x_t - v_t\rangle \geq \frac{1}{2}\langle \nabla f(x_t), x_t - v_t\rangle$, *then it holds*:

$$f(x_t) - f(x_{t+1}) \geq \gamma_t \langle \nabla f(x_t), x_t - v_t\rangle/2 = \begin{cases} \frac{\langle \nabla f(x_t), x_t - v_t\rangle^2}{2M\|x_t - v_t\|^2} & \text{if} \quad \gamma_t \in [0,1) \\ \langle \nabla f(x_t), x_t - v_t\rangle/2 & \text{if} \quad \gamma_t = 1 \end{cases}.$$

Proof The proof follows directly via convexity and plugging in the definitions:

$$\begin{aligned} & f(x_t) - f(x_{t+1}) \\ & \geq \langle \nabla f(x_{t+1}), x_t - x_{t+1}\rangle && \text{(convexity)} \\ & \geq \gamma_t \langle \nabla f(x_{t+1}), x_t - v_t\rangle && \text{(definition of } x_{t+1}) \\ & \geq \gamma_t \langle \nabla f(x_t), x_t - v_t\rangle/2 && \text{(assumption of (altAdaptive-simple))} \\ & = \begin{cases} \frac{\langle \nabla f(x_t), x_t - v_t\rangle^2}{2M\|x_t - v_t\|^2} & \text{if } \gamma_t \in [0,1) \\ \langle \nabla f(x_t), x_t - v_t\rangle/2 & \text{if } \gamma_t = 1 \end{cases}. && \text{(definition of } \gamma_t) \end{aligned}$$

□

Note that the proof above (again) explicitly relies on the convexity of f. It remains to show that (altAdaptive-simple) holds for $2L$, whenever the function is L-smooth and γ_t is the corresponding short step for $M = 2L$. The proof is very similar to before, however the last step is different.

Lemma 7.2 (Smoothness implies (altAdaptive-simple)) *Let f be L-smooth. Further, let $x_{t+1} = (1-\gamma_t)x_t + \gamma_t v_t$ with $\gamma_t = \min\left\{\frac{\langle \nabla f(x_t), x_t - v_t\rangle}{M\|x_t - v_t\|^2}, 1\right\}$ and $M = 2L$. Then* (altAdaptive-simple) *holds, i.e.,*

$$\langle \nabla f(x_{t+1}), x_t - v_t\rangle \geq \frac{1}{2}\langle \nabla f(x_t), x_t - v_t\rangle.$$

Proof We use the alternative definition of smoothness using the gradients, i.e., we have

$$\langle \nabla f(y) - \nabla f(x), y - x\rangle \leq L\|y - x\|^2 \quad \text{for all } x, y \in P,$$

by Remark 2.3. Now plug in $x \leftarrow x_t$ and $y \leftarrow x_{t+1}$, so that we obtain

$$\langle \nabla f(x_{t+1}) - \nabla f(x_t), x_{t+1} - x_t\rangle \leq L\|x_{t+1} - x_t\|^2$$

and with plugging in the definition of x_{t+1} we obtain

$$\langle \nabla f(x_{t+1}) - \nabla f(x_t), \gamma_t(v_t - x_t)\rangle \leq L\gamma_t^2\|v_t - x_t\|^2.$$

If $\gamma_t > 0$, dividing by γ_t and then plugging in the definition of γ_t yields

$$\langle \nabla f(x_{t+1}) - \nabla f(x_t), v_t - x_t\rangle \leq \frac{1}{2}\langle \nabla f(x_t), x_t - v_t\rangle,$$

and rearranging gives the desired inequality

$$\langle \nabla f(x_{t+1}), x_t - v_t\rangle \geq \frac{1}{2}\langle \nabla f(x_t), x_t - v_t\rangle.$$

In case $\gamma_t = 0$ we have $x_t = x_{t+1}$ and the assertion holds trivially. □

We will now show that (altAdaptive-simple) is indeed weaker than (altAdaptive) and that we cannot replace $M = L$ in Lemma 4.11. To this end consider the following 1-dimensional example: Pick $f(x) \doteq x^2$, so that $L = 2$ holds. Consider $f : [-1, 1] \mapsto \mathbb{R}$ and $x_t = 1$. Then we have $\nabla f(x_t) = 2$, $v_t = -1$, and

$$\frac{\langle \nabla f(x_t), x_t - v_t\rangle}{L\|x_t - v_t\|^2} = \frac{2(1-(-1))}{2(1-(-1))^2} = \frac{1}{2},$$

so that $\gamma_t = \min\left\{\frac{\langle \nabla f(x_t), x_t - v_t\rangle}{L\|x_t - v_t\|^2}, 1\right\} = \frac{1}{2}$, $x_{t+1} = 0$, and $\nabla f(x_{t+1}) = 0$. This contradicts $0 = \langle \nabla f(x_{t+1}), x_t - v_t\rangle \geq \frac{1}{2}\langle \nabla f(x_t), x_t - v_t\rangle > 0$.

Acknowledgements The author would like to thank Gábor Braun for pointing out the alternative smoothness inequality used in Sect. 4.5, which gave rise to tighter bounds.

Funding Open Access funding enabled and organized by Projekt DEAL. This research was partially supported by the DFG Cluster of Excellence MATH+ (EXC-2046/1, project id 390685689) funded by the Deutsche Forschungsgemeinschaft (DFG).

References

1. Anari, N., Haghtalab, N., Naor, S., Pokutta, S., Singh, M., Torrico, A.: Structured robust submodular maximization: offline and online algorithms. In: Proceedings of AISTATS (2019)
2. Anari, N., Haghtalab, N., Naor, S., Pokutta, S., Singh, M., Torrico, A.: Structured robust submodular maximization: offline and online algorithms. INFORMS J. Comput. **33**, 1259–1684 (2021)
3. Bach, F.: Duality between subgradient and conditional gradient methods. SIAM J. Optim. **25**(1), 115–129 (2015)
4. Bach, F.: Submodular functions: from discrete to continuous domains. Math. Program. **175**, 419–459 (2019)
5. Badanidiyuru, A., Vondrák, J.: Fast algorithms for maximizing submodular functions. In: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1497–1514 (2014)
6. Bolte, J., Daniilidis, A., Lewis, A.: The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. SIAM J. Optim. **17**(4), 1205–1223 (2007)
7. Bomze, I.M., Rinaldi, F., Zeffiro, D.: Frank–Wolfe and friends: a journey into projection-free first-order optimization methods. 4OR **19**, 313–345 (2021)
8. Boyd, S., Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
9. Braun, G., Pokutta, S.: The matching polytope does not admit fully-polynomial size relaxation schemes. In: Proceeedings of SODA (2015)
10. Braun, G., Pokutta, S.: The matching polytope does not admit fully-polynomial size relaxation schemes. IEEE Trans. Inf. Theory **61**(10), 1–11 (2015)
11. Braun, G., Pokutta, S.: Dual Prices for Frank-Wolfe Algorithms (2021). https://arxiv.org/abs/2101.02087. Preprint, available at
12. Braun, G., Pokutta, S., Zink, D.: Lazifying conditional gradient algorithms. In: Proceedings of the International Conference on Machine Learning (ICML) (2017)
13. Braun, G., Pokutta, S., Tu, D., Wright, S.: Blended conditional gradients: the unconditioning of conditional gradients. In: Proceedings of ICML (2019)
14. Braun, G., Pokutta, S., Zink, D.: Lazifying conditional gradient algorithms. J. Mach. Learn. Res. **20**(71), 1–42 (2019)
15. Braun, G., Carderera, A., Combettes, C.W., Hassani, H., Karbasi, A., Mokthari, A., Pokutta, S.: (2022). Conditional gradient methods. Preprint available at https://arxiv.org/abs/2211.14103
16. Braun, G., Pokutta, S., Weismantel, R.: Alternating Linear Minimization: Revisiting von Neumann's alternating projections (2022). Preprint
17. Carderera, A., Pokutta, S.: Second-order Conditional Gradient Sliding (2020). Preprint available at https://arxiv.org/abs/2002.08907
18. Carderera, A., Pokutta, S., Schütte, C., Weiser, M.: CINDy: Conditional gradient-based Identification of Non-linear Dynamics – Noise-robust recovery (2021). Preprint available at https://arxiv.org/abs/2101.02630
19. Chen, Z., Sun, Y.: Reducing discretization error in the Frank–Wolfe method (2023). Preprint available at https://arxiv.org/abs/2304.01432

20. Chen, L., Harshaw, C., Hassani, H., Karbasi, A.: Projection-free online optimization with stochastic gradient: from convexity to submodularity. In: Proceedings of the 35th International Conference on Machine Learning (ICML), vol. 80, pp. 814–823. PMLR (2018)
21. Cheung, E., Li, Y.: Solving separable nonsmooth problems using Frank–Wolfe with uniform affine approximations. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), IJCAI'18, pp. 2035–2041. AAAI Press, Menlo Park (2018)
22. Clarkson, K.L.: Coresets, sparse greedy approximation, and the Frank–Wolfe algorithm. ACM Trans. Algorithms **6**(4), 1–30 (2010)
23. Combettes, C.W., Pokutta, S.: Boosting Frank-Wolfe by chasing gradients. In: Proceedings of ICML (2020)
24. Combettes, C.W., Pokutta, S.: Complexity of linear minimization and projection on some sets. Oper. Res. Lett. **49** (2021)
25. Combettes, C.W., Pokutta, S.: Revisiting the approximate Carathéodory problem via the Frank-Wolfe algorithm. Math. Program., Ser. A **197**, 191–214 (2023)
26. Dahik, C.: Robust discrete optimization under ellipsoidal uncertainty. PhD thesis, Bourgogne Franche-Comté (2021)
27. Dantzig, G.B.: Reminiscences about the origins of linear programming. Technical report, Stanford University, CA. Systems Optimization Lab (1981)
28. Dantzig, G.B.: Reminiscences About the Origins of Linear Programming, pp. 78–86. Springer, Berlin (1983)
29. de Oliveira, W.: Short paper – a note on the Frank–Wolfe algorithm for a class of nonconvex and nonsmooth optimization problems. Open J. Math. Optim. **4**, 1–10 (2023)
30. Designolle, S., Iommazzo, G., Besançon, M., Knebel, S., Gelß, P., Pokutta, S.: Improved local models and new Bell inequalities via Frank-Wolfe algorithms. Phys. Rev. Res. **5** (2023)
31. Designolle, S., Vértesi, T., Pokutta, S.: Symmetric multipartite Bell inequalities via Frank-Wolfe algorithms (2023). Preprint available at https://arxiv.org/abs/2310.20677
32. Dvurechensky, P., Ostroukhov, P., Safin, K., Shtern, S., Staudigl, M.: Self-concordant analysis of Frank–Wolfe algorithms. In: Daumé, H. III, Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 2814–2824. PMLR (2020)
33. Feldman, M., Naor, J.S., Schwartz, R.: A unified continuous greedy algorithm for submodular maximization. In: Proceedings of the 52nd Annual Symposium on Foundations of Computer Science (FOCS), pp. 570–579. IEEE, Los Alamitos (2011)
34. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Nav. Res. Logist. Q. **3**(1–2), 95–110 (1956)
35. Freund, R.M., Grigas, P., Mazumder, R.: An extended Frank–Wolfe method with "in-face" directions, and its application to low-rank matrix completion. SIAM J. Optim. **27**(1), 319–346 (2017)
36. Garber, D.: Revisiting Frank–Wolfe for polytopes: strict complementarity and sparsity. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 18883–18893. Curran Associates, Red Hook (2020)
37. Garber, D., Hazan, E.: Faster rates for the Frank–Wolfe method over strongly-convex sets. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning (ICML), vol. 37, pp. 541–549. PMLR (2015)
38. Garber, D., Hazan, E.: A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization. SIAM J. Optim. **26**(3), 1493–1528 (2016)
39. Garber, D., Kretzu, B.: Revisiting projection-free online learning: the strongly convex case. In: Banerjee, A., Fukumizu, K. (eds.) Proceedings of the 24th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 130, pp. 3592–3600. PMLR (2021)
40. Garber, D., Wolf, N.: Frank–Wolfe with a nearest extreme point oracle. In: Belkin, M., Kpotufe, S. (eds.) Proceedings of Thirty Fourth Conference on Learning Theory. Proceedings of Machine Learning Research, vol. 134, pp. 2103–2132. PMLR (2021)
41. Gilbert, E.G.: An iterative procedure for computing the minimum of a quadratic form on a convex set. SIAM J. Control **4**(1), 61–80 (1966)
42. GuéLat, J., Marcotte, P.: Some comments on Wolfe's 'away step'. Math. Program. **35**(1), 110–119 (1986)
43. Gupta, S., Goemans, M., Jaillet, P.: Solving combinatorial games using products, projections and lexicographically optimal bases (2016). Preprint available at https://arxiv.org/abs/1603.00522

44. Harchaoui, Z., Juditsky, A., Nemirovski, A.S.: Conditional gradient algorithms for norm-regularized smooth convex optimization. Math. Program. **152**, 75–112 (2015)
45. Hassani, H., Soltanolkotabi, M., Karbasi, A.: Gradient methods for submodular maximization. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems (NIPS), vol. 30, pp. 5841–5851. Curran Associates, Red Hook (2017)
46. Hazan, E., Kale, S.: Projection-free online learning. In: Proceedings of the 29th International Conference on Machine Learning (ICML), pp. 1843–1850. Omnipress, Madison (2012)
47. Hazan, E., Minasyan, E.: Faster projection-free online learning. In: Abernethy, J., Agarwal, S. (eds.) Proceedings of Thirty Third Conference on Learning Theory. Proceedings of Machine Learning Research, vol. 125, pp. 1877–1893. PMLR (2020)
48. Jaggi, M.: Revisiting Frank–Wolfe: projection-free sparse convex optimization. In: Dasgupta, S., McAllester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA. ICML'13, vol. 28, pp. 427–435. PMLR (2019)
49. Jing, N., Fang, E.X., Tang, C.Y.: Robust matrix estimations meet Frank–Wolfe algorithm. Mach. Learn., 1–38 (2023)
50. Joulin, A., Tang, K., Fei-Fei, L.: Efficient image and video co-localization with Frank–Wolfe algorithm. In: Proceedings of European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science, vol. 8694, pp. 253–268. Springer, Berlin (2014)
51. Kerdreux, T., Roux, C., d'Aspremont, A., Pokutta, S.: Linear bandits on uniformly convex sets. J. Mach. Learn. Res. **22**, 1–23 (2021)
52. Lacoste-Julien, S.: Convergence rate of Frank–Wolfe for non-convex objectives (2016). HAL hal-01415335
53. Lacoste-Julien, S., Jaggi, M.: On the global linear convergence of Frank–Wolfe optimization variants. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems (NIPS), vol. 28, pp. 496–504. Curran Associates, Red Hook (2015)
54. Lacoste-Julien, S., Jaggi, M., Schmidt, M., Pletscher, P.: Block-coordinate Frank–Wolfe optimization for structural SVMs. In: Dasgupta, S., McAllester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 28, pp. 53–61. PMLR (2013)
55. Lan, G.: The complexity of large-scale convex programming under a linear optimization oracle. Technical report, Department of Industrial and Systems Engineering, University of Florida (2013)
56. Lan, G., Pokutta, S., Zhou, Y., Zink, D.: Conditional accelerated lazy stochastic gradient descent. In: Proceedings of the International Conference on Machine Learning (ICML) (2017)
57. Levitin, E.S., Polyak, B.T.: Constrained minimization methods. USSR Comput. Math. Math. Phys. **6**(5), 1–50 (1966)
58. Macdonald, J., Besançon, M., Pokutta, S.: Interpretable neural networks with Frank-Wolfe: sparse relevance maps and relevance orderings. In: Proceedings of ICML (2022)
59. Mirrokni, V., Leme, R.P., Vladu, A., Wong, S.C.-W.: Tight bounds for approximate Carathéodory and beyond. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 70, pp. 2440–2448. PMLR, (2017)
60. Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A.: Fast constrained submodular maximization: personalized data summarization. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of the 33nd International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 48, pp. 1358–1367. PMLR (2016)
61. Mokhtari, A., Hassani, H., Karbasi, A.: Conditional gradient method for stochastic submodular maximization: closing the gap. In: Storkey, A., Perez-Cruz, F. (eds.) Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 84, pp. 1886–1895. PMLR (2018)
62. Mokhtari, A., Hassani, H., Karbasi, A.: Decentralized submodular maximization: bridging discrete and continuous settings. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on MachineLearning. Proceedings of Machine Learning Research, vol. 80, pp. 3616–3625. PMLR (2018)
63. Moondra, J., Mortagy, H., Gupta, S.: Reusing combinatorial structure: faster iterative projections over submodular base polytopes. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems, vol. 34, pp. 25386–25399. Curran Associates, Red Hook (2021)

64. Négiar, G., Dresdner, G., Tsai, A.Y.-T., El Ghaoui, L., Locatello, F., Freund, R.M., Pedregosa, F.: Stochastic Frank–Wolfe for constrained finite-sum minimization. In: Daumé, H. III, Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 119, pp. 7253–7262. PMLR (2020)
65. Nemirovski, A.S., Yudin, D.B.: Problem Complexity and Method Efficiency in Optimization. Wiley, New York (1983)
66. Nesterov, Y.E.: Introductory Lectures on Convex Optimization: A Basic Course, 1st edn. Applied Optimization, vol. 87. Springer, Berlin (2004)
67. Nesterov, Y.E.: Lectures on Convex Optimization. Optimization and Its Applications, vol. 137. Springer, Berlin (2018)
68. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer Series in Operations Research and Financial Engineering. Springer, Berlin (2006)
69. Pedregosa, F., Negiar, G., Askari, A., Jaggi, M.: Linearly convergent Frank–Wolfe with backtracking line-search. In: Chiappa, S., Calandra, R. (eds.) Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, (AISTATS). Proceedings of Machine Learning Research, vol. 108, pp. 1–10. PMLR (2020)
70. Pierucci, F., Harchaoui, Z., Malick, J.: A smoothing approach for composite conditional gradient with nonsmooth loss. In: Conférence d'Apprentissage Automatique (CAp) (2014)
71. Potra, F.A., Wright, S.J.: Interior-point methods. J. Comput. Appl. Math. **124**(1–2), 281–302 (2000)
72. Ravi, S.N., Collins, M.D., Singh, V.: A deterministic nonsmooth Frank Wolfe algorithm with coreset guarantees. INFORMS J. Optim. **1**(2), 120–142 (2019)
73. Rothvoss, T.: The matching polytope has exponential extension complexity. J. ACM **64**(6), 41 (2017)
74. Sinha, M.: Lower bounds for approximating the matching polytope. In: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1585–1604. SIAM, Philadelphia (2018)
75. Tang, K., Joulin, A., Li, L.-J., Fei-Fei, L.: Co-localization in real-world images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1464–1471 (2014)
76. Teboulle, M., Vaisbourd, Y.: An elementary approach to tight worst case complexity analysis of gradient based methods. Math. Program. **201**(1–2), 63–96 (2023)
77. Thuerck, D., Sofranac, B., Pfetsch, M., Pokutta, S.: Learning cuts via enumeration oracles. Proceedings of NeurIPS. (2023). To appear
78. Tsuji, K., Tanaka, K., Pokutta, S.: Pairwise conditional gradients without swap steps and sparser kernel herding. In: Proceedings of ICML (2022)
79. Vondrák, J.: Optimal approximation for the submodular welfare problem in the value oracle model. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC), pp. 67–74 (2008)
80. Wäldchen, S., Huber, F., Pokutta, S.: Training characteristic functions with reinforcement learning: XAI-methods play connect four. In: Proceedings of ICML (2022)
81. Wang, Y.-X., Sadhanala, V., Dai, W., Neiswanger, W., Sra, S., Xing, E.P.: Parallel and distributed block-coordinate Frank–Wolfe algorithms. In: Proceedings of the 33rd International Conference on Machine Learning (ICML), vol. 48, pp. 1548–1557. PMLR (2016)
82. Wirth, E., Kerdreux, T., Pokutta, S.: Acceleration of Frank-Wolfe algorithms with open loop step-sizes. In: Proceedings of AISTATS (2023)
83. Wirth, E., Peña, J., Pokutta, S.: A new open-loop strategy for Frank-Wolfe algorithms (2023). In preparation
84. Wirth, E., Peña, J., Pokutta, S.: Accelerated Affine-Invariant Convergence Rates of the Frank-Wolfe Algorithm with Open-Loop Step-Sizes (2023). Preprint available at https://arxiv.org/abs/2310.04096
85. Wolfe, P.: Convergence theory in nonlinear programming. In: Integer and Nonlinear Programming, pp. 1–36. North-Holland, Amsterdam (1970)
86. Zhang, W., Zhao, P., Zhu, W., Hoi, S.C.H., Zhang, T.: Projection-free distributed online learning in networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 70, pp. 4054–4062. PMLR (2017)
87. Zhang, W., Shi, Y., Zhang, B., Yuan, D.: Dynamic regret of distributed online Frank–Wolfe convex optimization (2023)

Sebastian Pokutta is the Vice President of Zuse Institute Berlin and a Professor at TU Berlin, specializing in AI and Optimization. He has an extensive academic and professional background, including positions at MIT, IBM ILOG, Krall Demmel Baumgarten, and the Georgia Institute of Technology. Pokutta has received numerous accolades, including the Gödel Prize in 2023, the STOC Test of Time award in 2022, and several early career and best paper awards.