# Flexible block-iterative analysis for the Frank-Wolfe algorithm

Gábor Braun, Sebastian Pokutta, and Zev Woodstock*
Zuse Institute Berlin and
Institute of Mathematics, Technische Universität Berlin, Germany

July 24, 2024

**Abstract**

We prove that the block-coordinate Frank-Wolfe (BCFW) algorithm converges with state-of-the-art rates in both convex and nonconvex settings under a very mild "block-iterative" assumption, newly allowing for (I) progress without activating the most-expensive linear minimization oracle(s), LMO(s), at every iteration, (II) parallelized updates that do not require all LMOs, and therefore (III) deterministic parallel update strategies that take into account the numerical cost of the problem's LMOs. Our results apply for short-step BCFW as well as an adaptive method for convex functions. New relationships between updated coordinates and primal progress are proven, and a favorable speedup is demonstrated using `FrankWolfe.jl`.

**Keywords**. conditional gradient, block-iterative algorithm, Frank-Wolfe, projection-free first-order method

**MSC Classification**. 49M27, 49M37, 65K05, 90C26, 90C30,

## 1 Introduction

Given a smooth function $f$ that maps from a finite Cartesian product of $m$ real Hilbert spaces $\boldsymbol{\mathcal{H}} \coloneqq \bigoplus_{i=1}^{m} \mathcal{H}_i$ to $\mathbb{R}$ and a product of nonempty compact convex subsets $\bigtimes_{i=1}^{m} C_i \subset \boldsymbol{\mathcal{H}}$ with $C_i \subseteq \mathcal{H}_i$, we seek to solve the following problem

$$\underset{\boldsymbol{x} \in C_1 \times \ldots \times C_m}{\text{minimize}} \; f(\boldsymbol{x}), \tag{1}$$

which has applications in matrix factorization, support vector machine training, sequence labeling, intersection verification, and more [5, 8, 15, 18, 19, 26, 29, 30]. Frank-Wolfe (F-W), also known as

*Corresponding author: woodstock@zib.de

*conditional gradient*, methods have become an increasingly popular choice for solving (1) on large-scale problems, because their method of enforcing set constraints, namely the *linear minimization oracle*, is oftentimes computationally faster than other techniques such as projection algorithms [12]. A *linear minimization oracle* $\mathrm{LMO}_C$ for a compact convex set $C \subseteq \boldsymbol{\mathcal{H}}$, computes for any linear objective $\boldsymbol{c} \in \boldsymbol{\mathcal{H}}$, a point in $\operatorname{Argmin}_{\boldsymbol{x} \in C} \langle \boldsymbol{c} \mid \boldsymbol{x} \rangle$. The oracle approach is advantageous for problems such as the maximum matching problem [7, 24], where efficient linear minimization is possible despite large linear program formulations; this also makes reduction between problems easier [9].

Although (1) can be solved via the classical Frank-Wolfe algorithm, it would necessitate that, at every iteration, the linear minimization oracle for $C_1 \times \ldots \times C_m$ is evaluated. This step can cause a computational bottleneck, since

$$\mathrm{LMO}_{C_1 \times \ldots \times C_m}(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m) = (\mathrm{LMO}_{C_1} \boldsymbol{x}^1, \ldots, \mathrm{LMO}_{C_m} \boldsymbol{x}^m), \tag{2}$$

i.e., evaluating the Cartesian LMO amounts to computing $m$ separate LMOs. To avoid this slowdown, there has been an increasing effort over the last decade to reduce the per-iteration complexity required by classical Frank-Wolfe algorithms while maintaining theoretical guarantees of convergence [2, 5, 8, 18, 19, 29]. These benefits make performing a single iteration on larger-scale problems more tractable, and oftentimes allow for the more efficient use of kilowatts in practice.

Here we are interested in improvements making use of the product structure of the feasible region, which can later be combined with other improvement techniques to better use linear minimization, such as delaying updates via local acceleration [14], generalized self-concordant objective functions [10], and *boosting*, i.e., using multiple linear minimizations to choose a direction for progress [11].

Perhaps the earliest work using the product structure of the feasible region was [21], which proved that, for Armijo and exact line searches, asymptotic convergence to a solution of (1) could be achieved by, at each iteration, only updating one component (also called coordinate) of the iterate and thereby requiring one LMO evaluation. In particular, [21] showed that convergence is guaranteed as long as an *essentially cyclic* selection scheme is used, that is, as long as there exists some $K$ such that all $m$ components are updated at least once over each consecutive $K$ iterations. In other words, the index $\mathrm{i}(t)$ of the component updated at iteration $t \in \mathbb{N}$, satisfies

$$\{\mathrm{i}(t), \ldots, \mathrm{i}(t + K - 1)\} = \{1, \ldots, m\}. \tag{3}$$

About 17 years later, [2] significantly improved upon these results for the *cyclic* setting ($K = m$), by (A) widening to a scope of many more Frank-Wolfe variants (e.g., adaptive steps, open-loop predefined steps, and backtracking) and (B) deriving modern convergence rates. This cyclic scheme has shown to be particularly useful with randomly shuffling the order of updating the components for each cycle. In contrast to the deterministic methods, [18] showed that by selecting uniformly at random the component to update in each iteration, one can also solve (1). Since then, two methods have been proposed that select one component to update based on a suboptimality criterion: the one in [19] is stochastic and selects the component via a non-uniform distribution, while the Gauss-Southwell, or "greedy", update scheme of [5] is deterministic. Such techniques can provide improved per-iteration progress, although they are agnostic to the numerical costs of the selected LMO.

In contrast to singleton-update schemes, the vanilla Frank-Wolfe algorithm and several of its modern variants [5, 29] are particularly suitable for updating several components of an iterate in parallel, which can yield better per-iteration progress. In these *block-iterative* settings, at iteration $t \in \mathbb{N}$, a *block* $I_t \subset \{1, \dots, m\}$ of components are updated (possibly in parallel) while leaving the remaining components in $\{1, \dots, m\} \setminus I_t$ unchanged. Updated components $i \in I_t$ are modified via a Frank-Wolfe subroutine which relies on evaluating $\mathrm{LMO}_{C_i}$, and therefore the selection of $I_t$ has a great influence on the per-iteration cost of the algorithm. Some of the earliest results for parallel block updated Frank-Wolfe algorithms again arise from [21], which proved convergence (without rates) for parallel synchronous updates with the full updating scheme $I_t = \{1, \dots, m\}$ and a uniform step size across all components[1]. As pointed out by [2], using a single step size in all components can impede progress, since the relative scale between componentwise constraints can be significant. The recent work [5] allowed for full updates with variable componentwise stepsizes, also significantly improving convergence rates in certain settings.

However, outside of full-updates, it appears that only [29] allows for block-sizes larger than 1. In particular, for a fixed block-size $p$, the results in [29] permit selecting the updated coordinates uniformly at random. This application is ideal when all LMOs are expected to require the same amount of time, and $p$ processor cores are available. However, unless all the operators $(\mathrm{LMO}_{C_i})_{i \in \{1,\dots,m\}}$ require similar levels of computational effort, there appear to be no other good options for leveraging parallelism. In particular, regardless of the stepsizes considered, it appears that (prior to this work) no block selection technique for a F-W algorithm allows block sizes to change between iterations, and there are no *deterministic* rules which even allow for blocks $I_t$ with sizes between 1 and $m$. This poses a significant drawback from a computational perspective, because the current "state-of-the-art" leaves very little customizability or adaptability in how the block-updates are selected. In particular, a central goal of this work is to allow for the design of cost-aware update techniques which take into account the relative numerical cost of the LMOs of a given problem, and utilize all available processors at a given iteration.

Even though the Frank-Wolfe algorithm predates many methods which rely on proximity operators, advances in block-coordinate proximal algorithms seem to have outpaced those in the Frank-Wolfe literature. So, in this article we consider parallel and partial componentwise updates for BCFW under the following assumption, which comes from the proximal-based literature [20].

**Assumption 1.1.** *There exists a positive integer $K$ such that, for every iteration $t$,*

$$(\forall i \in \{1, \dots, m\}) \quad i \in \bigcup_{k=t}^{t+K-1} I_k. \tag{4}$$

We emphasize the flexibility of Assumption 1.1: in addition to allowing for the computation of expensive LMOs at any (bounded) rate, Assumption 1.1 allows deterministic parallelized block-updates of variable sizes, up to the user. Assumption 1.1 also unifies several existing selection schemes. Below are some example use-cases.

[1] Although [21, Section 4] also contains results for more general selection schemes of $I_t$, they do not apply to the Frank-Wolfe setting (see [21, Table 1]).

(i) With the $I_t$ singletons, this becomes the *essentially cyclic* selection scheme (3) of [21, 28]; if additionally $K = m$, Assumption 1.1 becomes the *cyclic* scheme of [2].

(ii) With $I_t = \{1, \dots, m\}$, this becomes the *full* selection method, also called *parallel* [5, 21].

(iii) If $p$ processor cores are available, one can queue $p$ many LMO operations to be performed in parallel, hence satisfying Assumption 1.1 with $K = \lceil m/p \rceil$. This strategy is well-suited for reducing processor wait times if the LMOs for the selected blocks require roughly the same amount of computational time (which occurs, e.g., in [2, 19]).

(iv) If the operators $(\mathrm{LMO}_{C_i})_{i\in\{1,\dots,m\}}$ require drastically different levels of computational time (e.g., where some LMOs are fast, while others require comparatively slower computations such as eigendecomposing a large matrix or solving a large linear program), one can postpone evaluating the most expensive LMOs, provided they are evaluated once every $K$ iterations. The experiments in Section 4 demonstrate that, by repeatedly iterating on the "cheaper" components, one can nonetheless provide good per-iteration progress on the overall problem.[2]

(v) Assumption 1.1 also allows for a quasi-stochastic strategy: For all iterations from $t$ to $t+K-2$, use any stochastic selection technique; then, at iteration $t + K - 1$, additionally activate the (potentially empty) set of components which were not selected by the stochastic method.

### Contributions

Our main contributions are threefold. To the best of our knowledge, this article contains the first result concerning converge of BCFW in the nonconvex case where the objective function has a Lipschitz-continuous gradient and no extra assumptions. We are only aware of one work which addresses BCFW with nonconvex objectives, namely [5] establishes linear convergence under several assumptions including a Kurdyka-Łojasiewicz-type inequality. As is standard in Frank-Wolfe methods, Theorem 3.3 proves that after $t$ iterations, the algorithm is guaranteed to produce a point with *Frank-Wolfe gap* (a quantity closely related to Clarke stationarity [6]) being at most $\mathcal{O}(1/\sqrt{t})$. Second, for the case of convex objective functions, an $\mathcal{O}(1/t)$ primal gap convergence rate is proven for an adaptive step size version of BCFW which does not require a-priori smoothness estimation (Theorem 2.5); in consequence, Corollary 2.8 establishes convergence for short-step BCFW with a rate and constant which matches short-step FW [6, Theorem 2.2]. Third, throughout the entire article we only assume the flexible block-activation scheme, Assumption 1.1, which unifies many previous activation schemes for BCFW into one simple framework and allows for new block-selection strategies, e.g., those available for some prox-based algorithms. On toy problems for which there is a significantly disparate cost of linear minimization oracles, these new selection strategies are shown to perform comparably, or even *better* than existing methods in iterations, gradient evaluations, LMO evaluations, and time.

The remainder of the article is organized as follows. Section 1.1 details background and preliminary results; Section 1.2 presents a general formulation of BCFW, a discussion on step size

[2] This strategy is reminiscent of the Shamanskii-type Newton/Chord algorithms that only perform numerically expensive Hessian updates once over a finite sequence of iterations [17, 25].

variants, and the common progress estimation for convex objective functions. Section 2 considers convex objective functions and proves convergence under both adaptive step sizes and short-step sizes. Section 3 proves a convergence guarantee for nonconvex objective functions with Lipschitz-continuous gradients. Finally, Section 4 shows computational experiments.

### 1.1 Notation, standing assumptions, and auxiliary results

Let $I \coloneqq \{1, 2, \dots, m\}$, and we consider the direct sum $\boldsymbol{\mathcal{H}} \coloneqq \bigoplus_{i\in I} \mathcal{H}_i$ of real Hilbert spaces $\mathcal{H}_i$. We denote points of $\boldsymbol{\mathcal{H}}$ by bold letters, and components in the direct sum by upper indices, i.e., $\boldsymbol{x} = (\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^m) \in \boldsymbol{\mathcal{H}}$ with $\boldsymbol{x}^i \in \mathcal{H}_i$. The inner product on $\boldsymbol{\mathcal{H}}$ is $\langle \boldsymbol{x} \mid \boldsymbol{y}\rangle_{\boldsymbol{\mathcal{H}}} = \sum_{i\in I} \langle \boldsymbol{x}^i \mid \boldsymbol{y}^i\rangle_{\mathcal{H}_i}$, yielding the norm identity $\|\boldsymbol{x} - \boldsymbol{y}\|^2_{\boldsymbol{\mathcal{H}}} = \sum_{i\in I} \|\boldsymbol{x}^i - \boldsymbol{y}^i\|^2_{\mathcal{H}_i}$. For notational convenience, we treat the $\mathcal{H}_i$ as orthogonal subspaces of $\boldsymbol{\mathcal{H}}$, in particular, $\boldsymbol{x} = \sum_{i\in I} \boldsymbol{x}^i$. We will omit the subscripts $\boldsymbol{\mathcal{H}}$, $\mathcal{H}_i$ from norms and scalar products; this will not cause ambiguity as all are restrictions of the ones on $\boldsymbol{\mathcal{H}}$. For $J \subseteq I$, let $\boldsymbol{x}^J \coloneqq \sum_{i\in J} \boldsymbol{x}^j$ be the part of $\boldsymbol{x}$ in the components $\mathcal{H}_i$ for $i \in J$. For $i \in I$, let $C_i$ be a nonempty compact convex subset of $\mathcal{H}_i$. For $J \subset I$, let $\bigtimes_{i\in J} C_i$ be the set of points $\boldsymbol{x} \in \boldsymbol{\mathcal{H}}$ with $\boldsymbol{x}^i \in C_i$ for all $i \in J$ and $\boldsymbol{x}^i = 0$ for $i \notin J$. Let $D_J$ be the diameter of $\bigtimes_{i\in J} C_i$ (treated as a subset of $\bigoplus_{i\in J} \mathcal{H}_i \subset \boldsymbol{\mathcal{H}}$). We shall use the simplified notation $D_i \coloneqq D_{\{i\}}$ and $D \coloneqq D_I$.

Let $f$ be a Fréchet differentiable function mapping from $\bigtimes_{i\in J} C_i$ to $\mathbb{R}$. We denote partial gradients by $\nabla^J f(\boldsymbol{x}) \coloneqq (\nabla f(\boldsymbol{x}))^J$. For $L_f > 0$, a function $f$ is *$L_f$-smooth* on a convex set $C$ if

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in C) \quad f(\boldsymbol{y}) - f(\boldsymbol{x}) \le \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{L_f}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2; \tag{5}$$

which holds, e.g., if $\nabla f$ is $L_f$-Lipschitz continuous [6]. Recall that $f$ is *convex* on a convex set $C$ if

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in C) \quad \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{y} - \boldsymbol{x}\rangle \le f(\boldsymbol{y}) - f(\boldsymbol{x}). \tag{6}$$

For nonempty $J \subset I$ and $\boldsymbol{x}^J \in \bigoplus_{i\in J} \mathcal{H}_i$, the *linear minimization oracle* $\mathrm{LMO}_J(\boldsymbol{x}^J)$ returns a point in $\operatorname{Argmin}_{\boldsymbol{v}\in\bigtimes_{i\in J} C_i} \langle \boldsymbol{x}^J \mid \boldsymbol{v}\rangle$; also set $\mathrm{LMO}_i \coloneqq \mathrm{LMO}_{\{i\}}$. A *partial Frank-Wolfe gap* is given by

$$(\forall J \subset I)\left(\forall \boldsymbol{x} \in \bigtimes_{i\in I} C_i\right) \quad G_J(\boldsymbol{x}) = \langle \nabla^J f(\boldsymbol{x}) \mid \boldsymbol{x}^J - \mathrm{LMO}_J(\nabla^J f(\boldsymbol{x}))\rangle, \tag{7}$$

with $G_i = G_{\{i\}}$. The *Frank-Wolfe gap* (F-W gap) of $f$ over $\bigtimes_{i\in I} C_i$ at $\boldsymbol{x} \in \boldsymbol{\mathcal{H}}$ is given by

$$G_I(\boldsymbol{x}) \coloneqq \sup_{\boldsymbol{v}\in\bigtimes_{i\in I} C_i} \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{x} - \boldsymbol{v}\rangle = \sum_{i\in I} G_i(\boldsymbol{x}). \tag{8}$$

Note that, for every $\boldsymbol{x} \in \bigtimes_{i\in I} C_i$ and every $J \subset I$, we have $G_J(\boldsymbol{x}) \ge 0$. The F-W gap vanishes at a solution of (1) in the following sense [6]

$$\boldsymbol{x} \text{ is a stationary point of } \operatorname*{minimize}_{\boldsymbol{x}\in\bigtimes_{i\in I} C_i} f(\boldsymbol{x}) \quad \Leftrightarrow \quad \begin{cases} \boldsymbol{x} \in \bigtimes_{i\in I} C_i \\ G_I(\boldsymbol{x}) \le 0. \end{cases} \tag{9}$$

Before proceeding further, we gather several useful results.

**Lemma 1.2.** *Let $(C_i)_{i\in I}$ be nonempty compact convex subsets of real Hilbert spaces $(\mathcal{H}_i)_{i\in I}$, let $f\colon \bigtimes_{i\in I} C_i \to \mathbb{R}$ be convex, let $J \subset I$ be nonempty, and let $G_J$ be given by (7). Then,*

$$\left(\forall \boldsymbol{z} \in \bigtimes_{i\in I} C_i\right) \quad G_J(\boldsymbol{z}) \geq f(\boldsymbol{z}) - \min_{\substack{\boldsymbol{x}\in\bigtimes_{i\in I} C_i \\ \boldsymbol{x}^{I\setminus J}=\boldsymbol{z}^{I\setminus J}}} f(\boldsymbol{x}) \geq 0. \tag{10}$$

*Proof.* Let $\boldsymbol{x}^*_{\boldsymbol{z}} \in \operatorname{Argmin}_{\boldsymbol{x}^J\in\bigtimes_{i\in J} C_i} f(\boldsymbol{x}^J + \boldsymbol{z}^{I\setminus J})$. By (7) and optimality of the LMO, $G_J(\boldsymbol{z}) = \langle \nabla^J f(\boldsymbol{z}) \mid \boldsymbol{z}^J - \mathrm{LMO}_J(\nabla^J f(\boldsymbol{z}))\rangle \geq \langle \nabla^J f(\boldsymbol{z}) \mid \boldsymbol{z}^J - \boldsymbol{x}^*_{\boldsymbol{z}}\rangle = \langle \nabla f(\boldsymbol{z}) \mid \boldsymbol{z} - (\boldsymbol{x}^*_{\boldsymbol{z}} + \boldsymbol{z}^{I\setminus J})\rangle$, so using convexity yields (10). □

We will use the perspective function $\rho$ of a Huber loss, to simplify handling the minimum in the short step formula (short) below.

$$\rho\colon \mathbb{R}\times\mathbb{R}_{>0} \to \mathbb{R}\colon (x,b) \mapsto \begin{cases} |x| - \frac{b}{2} & \text{if } |x| \geq b \\ \frac{|x|^2}{2b} & \text{if } |x| \leq b. \end{cases} \tag{11}$$

**Fact 1.3.** The function $\rho$ is proper and jointly convex [1, Proposition 8.25, Ex. 8.44]. Also, for $b > 0$ and $x \geq 0$, note $\rho(x,\cdot)$ is clearly decreasing on $\mathbb{R}_{>0}$; $\rho(\cdot,b)$ is even and hence increasing on $\mathbb{R}_{\geq 0}$ [1, Proposition 11.7]; and $x - \frac{b}{2} \leq \rho(x,b)$ as a tangent line of the convex function $\rho(\cdot,b)$. Furthermore, $\rho$ is subadditive [1, Example 10.5]:

$$\sum_{i=1}^n \rho(x_i,b_i) \geq \rho\left(\sum_{i=1}^n x_i, \sum_{i=1}^n b_i\right). \tag{12}$$

**Lemma 1.4.** *Let $x$, $c$ be nonnegative numbers and let $b$ be a positive number. Then,*

$$\frac{(x-c)^2}{2b} + c \geq \rho(x,b) \tag{13}$$

*Proof.* Fixing $x$ and $b$, the left-hand side of (13) is a quadratic function of $c$ with minimum attained at $c = x - b$ for $x \geq b$, and $c = 0$ for $x \leq b$. Thus,

$$\text{if } x \geq b, \text{ then } \quad \frac{(x-c)^2}{2b} + c \geq x - \frac{b}{2}; \text{ if } x \leq b, \text{ then } \quad \frac{(x-c)^2}{2b} + c \geq \frac{x^2}{2b}. \tag{14}$$

□

The following takes inspiration from [3] and includes a nonmonotone sequence $(a_t)_{t\in\mathbb{N}}$ representing extra progress.

**Lemma 1.5.** *Let $h_t$ and $a_t$ be nonnegative numbers for $t \in \mathbb{N}$, let $b > 0$, let $\rho$ be given by (11), and suppose that $h_t - h_{t+1} \geq \rho(h_t + a_t, b)$ for every $t \in \mathbb{N}$. Then $(h_t)_{t\in\mathbb{N}}$ decreases monotonically and*

$$(\forall t \in \mathbb{N}) \quad h_t \leq \begin{cases} \frac{b}{2} - a_0 & \text{if } t = 1 \\ \dfrac{2b}{t - 1 + \frac{2b}{h_1} + \sum_{k=1}^{t-1} \frac{2a_k}{h_1} + \left(\frac{a_k}{h_1}\right)^2} & \text{if } t \geq 2. \end{cases} \tag{15}$$

*Proof.* Since $h_t - h_{t+1} \geq \rho(h_t + a_t, b) \geq 0$, the sequence $(h_t)_{t\in\mathbb{N}}$ is decreasing. Since $x - \frac{b}{2} \leq \rho(x, b)$, our recursion yields $h_0 + a_0 - b/2 \leq h_0 - h_1$, and rearranging proves $h_1 \leq b/2 - a_0$. Next, we observe that since $(h_t)_{t\in\mathbb{N}}$ is monotonic and $\rho$ is strictly monotonically increasing in its first argument, for every $k \geq 1$, we have $\rho(b, b) = \frac{b}{2} \geq h_1 \geq h_k \geq h_k - h_{k-1} \geq \rho(h_k + a_k, b)$, so $h_k + a_k \leq b$ and hence $\rho(h_k + a_k, b) = (h_k + a_k)^2/(2b)$. Now, fix $t \in \mathbb{N} \setminus \{0\}$. If $h_{t+1} = 0$, we are done; otherwise, by monotonicity we have $0 < h_{t+1} \leq \cdots \leq h_1$. So,

$$(\forall k \in \{1, \ldots, t\}) \quad \begin{aligned} \frac{1}{h_{k+1}} - \frac{1}{h_k} = \frac{h_k - h_{k+1}}{h_k h_{k+1}} &\geq \frac{(h_k + a_k)^2}{2b h_k h_{k+1}} = \frac{1}{2b}\left(\frac{h_k}{h_{k+1}} + \frac{2a_k}{h_{k+1}} + \frac{a_k^2}{h_k h_{k+1}}\right) \\ &\geq \frac{1}{2b}\left(1 + \frac{2a_k}{h_1} + \left(\frac{a_k}{h_1}\right)^2\right). \end{aligned} \tag{16}$$

We sum (16) over $k \in \{1, \ldots, t\}$ to find

$$\frac{1}{h_{t+1}} - \frac{1}{h_1} \geq \frac{1}{2b}\left(t + \sum_{k=1}^{t} \frac{2a_k}{h_1} + \left(\frac{a_k}{h_1}\right)^2\right), \tag{17}$$

and rearranging (17) completes the result. □

## 1.2 Generic form of BCFW

Consider the generic form of the block-coordinate Frank-Wolfe algorithm shown in Algorithm 1. The selection strategies of the blocks $(I_t)_{t\in\mathbb{N}}$ in [2, 18, 21] arise as special cases.

**Algorithm 1** Block-Coordinate Frank-Wolfe (BCFW), Generic form

**Require:** Function $f\colon \bigtimes_{i\in I} C_i \to \mathbb{R}$, gradient $\nabla f$, point $\boldsymbol{x}_0 \in \bigtimes_{i\in I} C_i$, linear minimization oracles $(\mathrm{LMO}_i)_{i\in I}$

**for** $t = 0, 1$ **to** $\ldots$ **do**
  Choose a nonempty block $I_t \subset I$
  $\boldsymbol{g}_t \leftarrow \nabla f(\boldsymbol{x}_t)$
  **for** $i = 1$ **to** $m$ **do**
    **if** $i \in I_t$ **then**
      $\boldsymbol{v}_t^i \leftarrow \mathrm{LMO}_i(\boldsymbol{g}_t^i)$
      $\gamma_t^i \leftarrow$ Step size parameter (see also Sections 2, 3)
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i + \gamma_t^i(\boldsymbol{v}_t^i - \boldsymbol{x}_t^i)$
    **else**
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i$
    **end if**
  **end for**
**end for**

**Remark 1.6.** For $L_f$-smooth objective functions $f$, the smoothness inequality (5) and Line 8 of Algorithm 1 yield

$$f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t) \leq \sum_{i \in I_t} \gamma_t^i \langle \nabla^i f(\boldsymbol{x}_t) \mid \boldsymbol{v}_t^i - \boldsymbol{x}_t^i \rangle + \frac{L_f}{2} (\gamma_t^i)^2 \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2. \tag{18}$$

To tighten the bound (18), a common step size choice is to minimize the summands via a componentwise analogue of the so-called *short step* [2]:

$$\gamma_t^i = \operatorname*{Argmin}_{\gamma \in [0,1]} \left( -\gamma G_i(\boldsymbol{x}_t) + \gamma^2 \frac{L_f}{2} \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2 \right) = \min \left\{ \frac{G_i(\boldsymbol{x}_t)}{L_f \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2}, 1 \right\}. \tag{short}$$

This is analyzed in Sections 2.2 and 3 for convex and nonconvex objectives respectively. Section 2.1 addresses a situation where a similar update to (short) is performed using an estimation.

The following examples demonstrate that Algorithm 1 need not converge using componentwise analogues of classical F-W stepsizes.

**Example 1.7** (Non-convergent componentwise line search)**.** Frank-Wolfe methods at a point $\boldsymbol{x}_t$ with vertex $\boldsymbol{v}_t$ are commonly known to converge where step sizes are selected by a linesearch, i.e., $\gamma_t = \operatorname{Argmin}_{\gamma \in [0,1]} f(\boldsymbol{x}_t + \gamma(\boldsymbol{v}_t - \boldsymbol{x}_t))$ [6]. However, when linesearch stepsizes are chosen componentwise, namely via

$$\gamma_t^i \in \operatorname*{Argmin}_{\gamma \in [0,1]} f\big(\boldsymbol{x}_t + \gamma(\boldsymbol{v}_t^i - \boldsymbol{x}_t^i)\big), \tag{19}$$

Algorithm 1 need not converge. Let $I \coloneqq \{1,2\}$, $\mathcal{H}_1 = \mathcal{H}_2 = \mathbb{R}$, $C_1 = C_2 = [-1,1]$, and $f(\boldsymbol{x}) \coloneqq (\boldsymbol{x}^1 + \boldsymbol{x}^2)^2$; in particular, $L_f = 4$. The minimal function value of $0$ is attained at the points $\boldsymbol{x}$ for which $\boldsymbol{x}^1 = -\boldsymbol{x}^2$. With full block activation $I_t = \{1,2\}$ and componentwise linesearch (19), the iterates of Algorithm 1 satisfy $\boldsymbol{x}_{t+1}^1 = -\boldsymbol{x}_t^2$ and $\boldsymbol{x}_{t+1}^2 = -\boldsymbol{x}_t^1$. Hence, a possible sequence of iterates is $((-1)^t, (-1)^t)$, which does not converge to optimality in function value.

**Example 1.8** (Non-convergent componentwise short-step)**.** In singleton-update cyclic schemes, it is possible to use a variant of (short) where, for every $i \in I$, $L_f$ is replaced by the Lipschitz constant $\beta_i$ of $\nabla f$ over the component $C_i$. More precisely, componentwise short-steps $\gamma_t^i = \min\{1, G_i(\boldsymbol{x}_t)/\beta_i \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2\}$ allow for larger step sizes [2], since $\beta_i \leq L_f$. However, in Example 1.7, $\beta_1 = \beta_2 = 2 \neq 4 = L_f$, i.e., $\gamma_t^i$ is the same as in Example 1.7. Therefore, using Algorithm 1 with $I_t = \{1,2\}$, this short step variant may produce the same iterates as Example 1.7, which do not get close to the optimal solution.

The following technical lemma is for combining with inequalities of the form $f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1}) \geq \tau P_t$ ($\tau > 0$) to construct convergence results in Sections 2 and 3. For $\tau = 1$, the above inequality naturally arises as a consequence of convexity and smoothness (seen in Fact 2.1). The $(M_t)_{t \in \mathbb{N}}$ play the role of approximate smoothness constants.

**Lemma 1.9.** *Let $\bigtimes_{i \in I} C_i \subset \boldsymbol{\mathcal{H}}$ be a finite product of nonempty compact convex sets $C_i$, let $D$ be the diameter of $\bigtimes_{i \in I} C_i$, let $f \colon \bigtimes_{i \in I} C_i \to \mathbb{R}$ be Fréchet differentiable, let $K$ be a positive integer, let*

*$(M_t)_{t\in\mathbb{N}}$ be a sequence of positive numbers, and for every $J \subset I$, let $G_J$ be given by (7) In the setting of Algorithm 1, for every $t \in \mathbb{N}$ and $J \subset I$, set $\boldsymbol{v}_t^J = \mathrm{LMO}_J(\boldsymbol{g}_t)$, set*

$$P_t = \langle \boldsymbol{g}_t \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - \frac{M_{t+1}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2}{2} + \frac{\|\boldsymbol{g}_t^{I\setminus I_t} - \boldsymbol{g}_{t+1}^{I\setminus I_t}\|^2}{2M_{t+1}}, \tag{20}$$

*set $A_t = \sum_{k=1}^{K-1} G_{I_{t+k-1}\cap(I_{t+k}\cup\cdots\cup I_{t+K-1})}(\boldsymbol{x}_{t+k}) \geq 0$, and for every $i \in I_t$ let Line 7 be specified by*

$$\gamma_t^i = \min\left\{1, \frac{G_i(\boldsymbol{x}_t)}{M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{v}_t^i\|^2}\right\}. \tag{21}$$

*Then, for every $t \in \mathbb{N}$, the following hold:*

(i) $(\forall J \subseteq I \setminus I_t) \quad P_t \geq \rho\big(|G_{I_t\cup J}(\boldsymbol{x}_t) - \langle \boldsymbol{g}_{t+1}^J \mid \boldsymbol{x}_t^J - \boldsymbol{v}_t^J\rangle|, M_{t+1}\|\boldsymbol{x}_t^{I_t\cup J} - \boldsymbol{v}_t^{I_t\cup J}\|^2\big)$.

(ii) $\sum_{k=0}^{K-1} P_{t+k} \geq \rho\left(G_{I_t\cup\cdots\cup I_{t+K-1}}(\boldsymbol{x}_t) + A_t, \sum_{k=1}^{K} M_{t+k}D^2\right)$.

*Proof.* Let $i \in I_t$. We claim

$$\langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{x}_{t+1}^i\rangle - \frac{M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{x}_{t+1}^i\|^2}{2} = \rho(\langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{v}_t^i\rangle, M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{v}_t^i\|^2). \tag{22}$$

We distinguish two cases depending on $\gamma_t^i$. If $\langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{v}_t^i\rangle \geq M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{v}_t^i\|^2$ then $\gamma_t^i = 1$ and $\boldsymbol{x}_{t+1}^i = \boldsymbol{v}_t^i$, therefore we find

$$\begin{aligned}\langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{x}_{t+1}^i\rangle - \frac{M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{x}_{t+1}^i\|^2}{2} &= \langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{v}_t^i\rangle - \frac{M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{v}_t^i\|^2}{2} \\ &= \rho(\langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{v}_t^i\rangle, M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{v}_t^i\|^2).\end{aligned} \tag{23}$$

On the other hand, if $\langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{v}_t^i\rangle \leq M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{v}_t^i\|^2$, then $\gamma_t^i = \langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{v}_t^i\rangle/(M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{v}_t^i\|^2)$, so

$$\begin{aligned}\langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{x}_{t+1}^i\rangle - \frac{M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{x}_{t+1}^i\|^2}{2} &= \frac{\langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{v}_t^i\rangle^2}{2M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{v}_t^i\|^2} \\ &= \rho(\langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{v}_t^i\rangle, M_{t+1}\|\boldsymbol{x}_t^i - \boldsymbol{v}_t^i\|^2).\end{aligned} \tag{24}$$

Summing up (22) for $i \in I_t$ and using subadditivity of $\rho$ (12), we obtain (i) for $J = \varnothing$. To show (i) for arbitrary $J \subseteq I \setminus I_t$, we use an additional norm inequality, then (i) for $J = \varnothing$, and Lemma 1.4 (with $c = 0$), followed by subadditivity and monotonicity of $\rho$ (12):

$$\begin{aligned}P_t &= \langle \boldsymbol{g}_t \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - \frac{M_{t+1}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2}{2} + \frac{\|\boldsymbol{g}_t^{I\setminus I_t} - \boldsymbol{g}_{t+1}^{I\setminus I_t}\|^2}{2M_{t+1}} \\ &\geq \langle \boldsymbol{g}_t \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - \frac{M_{t+1}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2}{2} + \frac{\langle \boldsymbol{g}_t^J - \boldsymbol{g}_{t+1}^J \mid \boldsymbol{x}_t^J - \boldsymbol{v}_t^J\rangle^2}{2M_{t+1}\|\boldsymbol{x}_t^J - \boldsymbol{v}_t^J\|^2} \\ &\geq \rho(\langle \boldsymbol{g}_t^{I_t} \mid \boldsymbol{x}_t^{I_t} - \boldsymbol{v}_t^{I_t}\rangle, M_{t+1}\|\boldsymbol{x}_t^{I_t} - \boldsymbol{v}_t^{I_t}\|^2) + \rho(|\langle \boldsymbol{g}_t^J - \boldsymbol{g}_{t+1}^J \mid \boldsymbol{x}_t^J - \boldsymbol{v}_t^J\rangle|, M_{t+1}\|\boldsymbol{x}_t^J - \boldsymbol{v}_t^J\|^2) \\ &\geq \rho(|G_{I_t\cup J}(\boldsymbol{x}_t) - \langle \boldsymbol{g}_{t+1}^J \mid \boldsymbol{x}_t^J - \boldsymbol{v}_t^J\rangle|, M_{t+1}\|\boldsymbol{x}_t^{I_t\cup J} - \boldsymbol{v}_t^{I_t\cup J}\|^2).\end{aligned}$$

Next, to show (ii), we begin by using (i) with monotonicity of $\rho$:

$$(\forall J \subset I) \quad P_t \geq \rho(|G_J(\boldsymbol{x}_t) - \langle \boldsymbol{g}_{t+1}^{J \setminus I_t} \mid \boldsymbol{x}_t^{J \setminus I_t} - \boldsymbol{v}_t^{J \setminus I_t} \rangle|, M_{t+1} D^2). \tag{25}$$

Summing up (25) for $k \in \{t, \dots, t+K-1\}$ with the sets $J_k := \bigcup_{j=t+k}^{t+K-1} I_j$, we bound the righthand side using monotonicity and subadditivity of $\rho$:

$$\begin{aligned} \sum_{k=0}^{K-1} P_{t+k} &\geq \sum_{k=0}^{K-1} \rho(|G_{J_k}(\boldsymbol{x}_{t+k}) - \langle \boldsymbol{g}_{t+k+1}^{J_k \setminus I_{t+k}} \mid \boldsymbol{x}_{t+k}^{J_k \setminus I_{t+k}} - \boldsymbol{v}_{t+k}^{J_k \setminus I_{t+k}} \rangle|, M_{t+k+1} D^2) \\ &\geq \rho\left( \tilde{G}, \sum_{k=1}^{K} M_{t+k} D^2 \right), \end{aligned} \tag{26}$$

where, using Line 10 and the notational convention that $G_\varnothing(\boldsymbol{x}_{t+K-1}) = 0$,

$$\begin{aligned} \tilde{G} &:= \sum_{k=0}^{K-1} G_{J_k}(\boldsymbol{x}_{t+k}) - \langle \boldsymbol{g}_{t+k+1}^{J_k \setminus I_{t+k}} \mid \boldsymbol{x}_{t+k}^{J_k \setminus I_{t+k}} - \boldsymbol{v}_{t+k}^{J_k \setminus I_{t+k}} \rangle \\ &= \sum_{k=0}^{K-1} G_{J_k}(\boldsymbol{x}_{t+k}) - G_{J_k \setminus I_{t+k}}(\boldsymbol{x}_{t+k+1}) + \langle \boldsymbol{g}_{t+k+1}^{J_k \setminus I_{t+k}} \mid \boldsymbol{v}_{t+k}^{J_k \setminus I_{t+k}} - \boldsymbol{v}_{t+k+1}^{J_k \setminus I_{t+k}} \rangle \\ &= G_{J_0}(\boldsymbol{x}_t) + \sum_{k=1}^{K-1} \left( G_{J_k}(\boldsymbol{x}_{t+k}) - G_{J_k \setminus I_{t+k-1}}(\boldsymbol{x}_{t+k}) \right) + \sum_{k=0}^{K-1} \langle \boldsymbol{g}_{t+k+1}^{J_k \setminus I_{t+k}} \mid \boldsymbol{v}_{t+k}^{J_k \setminus I_{t+k}} - \boldsymbol{v}_{t+k+1}^{J_k \setminus I_{t+k}} \rangle \\ &= G_{I_t \cup \dots \cup I_{t+K-1}}(\boldsymbol{x}_t) + \sum_{k=1}^{K-1} G_{I_{t+k-1} \cap J_k}(\boldsymbol{x}_{t+k}) + \sum_{k=0}^{K-1} \langle \boldsymbol{g}_{t+k+1}^{J_k \setminus I_{t+k}} \mid \boldsymbol{v}_{t+k}^{J_k \setminus I_{t+k}} - \boldsymbol{v}_{t+k+1}^{J_k \setminus I_{t+k}} \rangle \\ &\geq G_{I_t \cup \dots \cup I_{t+K-1}}(\boldsymbol{x}_t) + \sum_{k=1}^{K-1} G_{I_{t+k-1} \cap J_k}(\boldsymbol{x}_{t+k}) \geq 0. \end{aligned}$$

The last two inequalities use nonnegativity of all the summands involved, relying on minimality of the $\boldsymbol{v}_{t+k+1}$ terms. Finally, using monotonicity of $\rho$ again:

$$\sum_{k=0}^{K-1} P_{t+k} \geq \rho\left( \tilde{G}, \sum_{k=1}^{K} M_{t+k} D^2 \right) \geq \rho\left( G_{I_t \cup \dots \cup I_{t+K-1}}(\boldsymbol{x}_t) + A_t, \sum_{k=1}^{K} M_{t+k} D^2 \right).$$

□

**Remark 1.10** (Interpretation of the terms $A_t$ in Lemma 1.9)**.** For every $t \in \mathbb{N}$, each of the following summands in the lower bound of Lemma 1.9

$$A_t = \sum_{k=1}^{K-1} G_{I_{t+k-1} \cap (I_{t+k} \cup \dots \cup I_{t+K-1})}(\boldsymbol{x}_{t+k}) \geq 0 \tag{27}$$

is a partial Frank-Wolfe gap for components that are updated more than once between iterations $t$ and $t+K-1$. Via Lemma 1.2, for each collection of reactivated components $J \subset I$, if $f$ is convex then each summand can be bounded by

$$G_J(\boldsymbol{x}_{t+k}) \geq f(\boldsymbol{x}_{t+k}) - \min_{\substack{\boldsymbol{x} \in \bigtimes_{i \in I} C_i \\ \boldsymbol{x}^{I \setminus J} = \boldsymbol{x}_{t+k}^{I \setminus J}}} f(\boldsymbol{x}) \geq 0. \tag{28}$$

As will be seen in Sections 2 and 3, the terms (27) contribute to faster convergence, and they may explain the favorable behavior observed in Section 4. However, in general, (27) may not always be strictly positive. Hence, we do know how to utilize these terms to construct a worst-case rate which is better than the cyclic-type rates of $\mathcal{O}(K/t)$ for convex objective functions (Section 2) and $\mathcal{O}(\sqrt{K/t})$ for nonconvex objectives (Section 3). We conjecture that under additional hypotheses (potentially hemivariance, which has been successfully used in other block-coordinate problems [28]), these terms may lead to an improved convergence result.

# 2 Convex objective functions

In this section we show that under two step size regimes, using Algorithm 1 with Assumption 1.1, the primal gap of a convex objective function is guaranteed to converge at a rate of $\mathcal{O}(1/\lfloor t/K \rfloor)$ after $t$ iterations. Section 2.1 is devoted to an adaptive step size scheme whereby the constant $L_f$ may be unknown a-priori. As a consequence, in Section 2.2 we also achieve convergence for the block-wise "short-step" variant of Frank-Wolfe (also sometimes called "adaptive" [2, Section 4.2]), where an overestimation of $L_f$ is available. Our convergence rates (Theorem 2.5 and Corollary 2.8) match for the special case of cyclic activation [2].

## 2.1 Analysis for adaptive step sizes

In recent years, Frank-Wolfe methods have been developed to address the situation where the smoothness constant of the objective $L_f$ is not known. These *backtracking*, or *adaptive* variants dynamically maintain an estimated smoothness constant across iterations, typically ensuring that the smoothness inequality (5) holds empirically between the current iterate $\boldsymbol{x}_t$ and the next iterate $\boldsymbol{x}_{t+1}$, at the expense of extra gradient and/or function evaluations [2, 22, 23]. In this section, we present a similar method for BCFW under Assumption 1.1.

Our analysis relies on the following which, to the best of our knowledge, first appeared in [16] and was later shown to characterize convex smooth interpolability [27].

**Fact 2.1** ([16])**.** Let $f \colon \boldsymbol{\mathcal{H}} \to \mathbb{R}$ be convex and $L_f$-smooth on $\boldsymbol{\mathcal{H}}$. Then,

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{\mathcal{H}}) \quad f(\boldsymbol{x}) - f(\boldsymbol{y}) - \langle \nabla f(\boldsymbol{y}) \mid \boldsymbol{x} - \boldsymbol{y} \rangle \geq \frac{\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|^2}{2L_f}. \tag{29}$$

**Algorithm 2** Adaptive Block-Coordinate Frank-Wolfe

**Require:** Function $f\colon \bigtimes_{i\in I} C_i \to \mathbb{R}$, gradient $\nabla f$, point $\boldsymbol{x}_0 \in \bigtimes_{i\in I} C_i$, linear minimization oracles $(\mathrm{LMO}_i)_{i\in I}$, smoothness estimation $M_0 > 0$, and approximation parameters $0 < \eta \leq 1 < \tau$.

1: **for** $t = 0, 1$ **to** $\dots$ **do**
2: Choose a nonempty block $I_t \subset I$ (See Assumption 1.1)
3: $\boldsymbol{g}_t \leftarrow \nabla f(\boldsymbol{x}_t)$
4: $(\widetilde{\boldsymbol{x}}^i_{t+1})_{i\in I\setminus I_t} \leftarrow (\boldsymbol{x}^i_t)_{i\in I\setminus I_t}$ # Indices outside of $I_t$ unchanged
5: $\widetilde{M}_{t+1} \leftarrow \eta M_t$ # Candidate smoothness constant for iteration $t+1$
6: **for** $i \in I_t$ **do**
7: $\boldsymbol{v}^i_t \leftarrow \mathrm{LMO}_i(\boldsymbol{g}^i_t)$
8: $\gamma^i_t \leftarrow \min\left\{1, \dfrac{\langle \boldsymbol{g}^i_t \mid \boldsymbol{x}^i_t - \boldsymbol{v}^i_t\rangle}{\widetilde{M}_{t+1}\|\boldsymbol{x}^i_t - \boldsymbol{v}^i_t\|^2}\right\}$
9: $\widetilde{\boldsymbol{x}}^i_{t+1} \leftarrow \boldsymbol{x}^i_t + \gamma^i_t(\boldsymbol{v}^i_t - \boldsymbol{x}^i_t)$
10: **end for**
11: **while** $f(\boldsymbol{x}_t) - f(\widetilde{\boldsymbol{x}}_{t+1}) - \langle \nabla f(\widetilde{\boldsymbol{x}}_{t+1}) \mid \boldsymbol{x}_t - \widetilde{\boldsymbol{x}}_{t+1}\rangle < \|\boldsymbol{g}_t - \nabla f(\widetilde{\boldsymbol{x}}_{t+1})\|^2/2\widetilde{M}_{t+1}$ **do**
12: $\widetilde{M}_{t+1} \leftarrow \tau\widetilde{M}_{t+1}$ # If (29) does not hold, increase the smoothness estimate.
13: **for** $i \in I_t$ **do**
14: Update $\gamma^i_t$ and $\widetilde{\boldsymbol{x}}^i_{t+1}$ as in lines 8 and 9.
15: **end for**
16: **end while**
17: $\boldsymbol{x}_{t+1} \leftarrow \widetilde{\boldsymbol{x}}_{t+1}$ # Guarantees that (29) holds for relevant points
18: $M_{t+1} \leftarrow \widetilde{M}_{t+1}$
19: **end for**

The interpolability result [27] implies that a function $f$ satisfying (29) for all $\boldsymbol{x}, \boldsymbol{y}$ in a convex set has an extension to a convex $L_f$-smooth function on $\boldsymbol{\mathcal{H}}$, and therefore, for simplicity of presentation, our results in Section 2 assume that $f$ is already extended, i.e., convex and $L_f$-smooth on $\boldsymbol{\mathcal{H}}$. For objective functions which cannot be extended to $\boldsymbol{\mathcal{H}}$, see Remark 3.4.

Fact 2.1 is particularly attractive for block-iterative algorithms, where differences of gradients often arise as error terms, while in (29) the difference appears as a lower bound on primal progress (further demonstrated in Lemma 2.4). This feature is the key to obtaining the same constant factors in the convergence guarantee as for traditional Frank-Wolfe algorithms, e.g., in [6, Theorem 2.2]. Hence, instead of checking the smoothness inequality as in [22] or another consequence as in [23], Algorithm 2 checks (29). Note that Algorithm 2 can be viewed as a version of Algorithm 1 where $\gamma^i_t$ is computed with an adaptive subroutine (see also Remark 1.6).

**Remark 2.2.** Similarly to [22, 23], by Fact 2.1, for all convex $L_f$-smooth objective functions $f$, the loop starting at Line 11 of Algorithm 2 always terminates, at latest the first time when $\widetilde{M}_{t+1} \geq L_f$, potentially overshooting by a factor of $\tau$. Hence $M_{t+1}$ can only be larger than $\tau L_f$ if the loop terminates immediately, i.e., without any multiplication by $\tau$ in Line 12. Let $t_0$ be the smallest

nonnegative integer with $\eta^{t_0} M_0 \leq \tau L_f$, which exists unless $M_0 > \tau L_f$ and $\eta = 1$. Therefore,

$$M_t = \eta^t M_0 > \tau L_f \qquad 1 \leq t < t_0 \tag{30}$$

$$M_t \leq \tau L_f \qquad t \geq t_0. \tag{31}$$

**Remark 2.3.** Even though the adaptive step size strategy in Algorithm 2 requires extra function and gradient evaluations (Lines 11–16), the LMOs are only computed once per iteration, namely in Line 7. In tandem with Assumption 1.1, this allows for flexible management of LMO costs.

The following presents a lower bound on primal progress.

**Lemma 2.4** (Progress bound via smoothness and convexity (29))**.** *Let $\bigtimes_{i\in I} C_i \subset \boldsymbol{\mathcal{H}}$ be a product of $m$ nonempty compact convex sets, let $D$ be the diameter of $\bigtimes_{i\in I} C_i$, let $f\colon \boldsymbol{\mathcal{H}} \to \mathbb{R}$, be convex and $L_f$-smooth, let $\rho$ be given by (11), let $\boldsymbol{x}^*$ be a solution to (1), and for every nonempty $J \subset I$ let $G_J$ be given by (7). In the setting of Algorithm 2, suppose that $K$ satisfies Assumption 1.1 and set $A_t = \sum_{k=1}^{K-1} G_{I_{t+k-1}\cap(I_{t+k}\cup\cdots\cup I_{t+K-1})}(\boldsymbol{x}_{t+k}) \geq 0$. Then $(f(\boldsymbol{x}_t))_{t\in\mathbb{N}}$ is monotonically decreasing and*

$$(\forall t \in \mathbb{N}) \quad f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+K}) \geq \rho\left(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*) + A_t, \sum_{k=1}^{K} M_{t+k} D^2\right). \tag{32}$$

*Proof.* Recall from Remark 2.2 that in Algorithm 2 the loop starting at Line 11 terminates, and therefore the algorithm generates an infinite sequence of iterates satisfying the first inequality of the following chain. The second inequality is a simple norm estimation, and the third one is a quadratic inequality, not needing any assumption on the scalar products and norms. We also make use of the fact $\boldsymbol{x}_t^{I\setminus I_t} = \boldsymbol{x}_{t+1}^{I\setminus I_t}$.

$$\begin{aligned}
f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1}) &\geq \langle \boldsymbol{g}_{t+1} \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle + \frac{\|\boldsymbol{g}_t - \boldsymbol{g}_{t+1}\|^2}{2M_{t+1}} \\
&= \langle \boldsymbol{g}_{t+1}^{I_t} \mid \boldsymbol{x}_t^{I_t} - \boldsymbol{x}_{t+1}^{I_t}\rangle + \frac{\|\boldsymbol{g}_t^{I_t} - \boldsymbol{g}_{t+1}^{I_t}\|^2}{2M_{t+1}} + \frac{\|\boldsymbol{g}_t^{I\setminus I_t} - \boldsymbol{g}_{t+1}^{I\setminus I_t}\|^2}{2M_{t+1}} \\
&\geq \langle \boldsymbol{g}_{t+1}^{I_t} \mid \boldsymbol{x}_t^{I_t} - \boldsymbol{x}_{t+1}^{I_t}\rangle + \frac{\langle \boldsymbol{g}_t^{I_t} - \boldsymbol{g}_{t+1}^{I_t} \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle^2}{2M_{t+1}\|\boldsymbol{x}_t^{I_t} - \boldsymbol{x}_{t+1}^{I_t}\|^2} + \frac{\|\boldsymbol{g}_t^{I\setminus I_t} - \boldsymbol{g}_{t+1}^{I\setminus I_t}\|^2}{2M_{t+1}} \\
&\geq \langle \boldsymbol{g}_t^{I_t} \mid \boldsymbol{x}_t^{I_t} - \boldsymbol{x}_{t+1}^{I_t}\rangle - \frac{M_{t+1}\|\boldsymbol{x}_t^{I_t} - \boldsymbol{x}_{t+1}^{I_t}\|^2}{2} + \frac{\|\boldsymbol{g}_t^{I\setminus I_t} - \boldsymbol{g}_{t+1}^{I\setminus I_t}\|^2}{2M_{t+1}} \\
&= \langle \boldsymbol{g}_t \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - \frac{M_{t+1}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2}{2} + \frac{\|\boldsymbol{g}_t^{I\setminus I_t} - \boldsymbol{g}_{t+1}^{I\setminus I_t}\|^2}{2M_{t+1}}.
\end{aligned} \tag{33}$$

Monotonicity of $(f(\boldsymbol{x}_t))_{t\in\mathbb{N}}$ follows from Lemma 1.9(i). Since Algorithm 2 is a special case of Algorithm 1, we can telescope the lefthand sum of (33) and invoke Lemma 1.9(ii) with Assumption 1.1 to find $f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+K}) \geq \rho(G_I(\boldsymbol{x}_t) + A_t, \sum_{k=1}^{K} M_{t+k} D^2)$. Since $f$ is convex, by optimality of the LMO and (6), we have $G_I(\boldsymbol{x}_t) \geq \langle \nabla f(\boldsymbol{x}_t) \mid \boldsymbol{x}_t - \boldsymbol{x}^*\rangle \geq f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)$, so (32) follows from monotonicity of $\rho$ (Fact 1.3). □

**Theorem 2.5.** *Let $\bigtimes_{i\in I} C_i \subset \boldsymbol{\mathcal{H}}$ be a product of $m$ nonempty compact convex sets, let $D$ be the diameter of $\bigtimes_{i\in I} C_i$, let $f\colon \boldsymbol{\mathcal{H}} \to \mathbb{R}$ be convex and $L_f$-smooth, let $\tau > 1 \geq \eta > 0$ and $M_0 > 0$ be approximation parameters, let $\boldsymbol{x}^*$ be a solution to (1), and for every nonempty $J \subset I$ let $G_J$ be given by (7). If $\eta = 1$, we assume $M_0 \leq \tau L_f$ and set $n_0 = 0$;[3] otherwise, $n_0 \coloneqq \max\{\lceil \log(\tau L_f/(\eta M_0))/(K\log\eta)\rceil, 0\}$. In the setting of Algorithm 2, suppose that $K$ satisfies Assumption 1.1, and set $A_t = \sum_{k=1}^{K-1} G_{I_{t+k-1}\cap(I_{t+k}\cup\cdots\cup I_{t+K-1})}(\boldsymbol{x}_{t+k}) \geq 0$. Then, after $t$ iterations, Algorithm 2 evaluates $f$ and $\nabla f$ at most $t+1+\max\{0, \lceil \log_\tau(\eta^{-t}L_f/M_0)\rceil\}$ times. Furthermore, for every $n \in \mathbb{N}$,*

$$f(\boldsymbol{x}_{nK}) - f(\boldsymbol{x}^*) \leq \begin{cases} \min\limits_{0\leq p\leq n-1} \left\{ \dfrac{K\eta^{pK} M_0 D^2}{2} - A_{pK} \right\} & \text{if } 1 \leq n \leq n_0+1 \\ \dfrac{2K\tau L_f D^2}{n - n_0 + \sum_{p=n_0}^{n} \frac{2A_{pK}}{f(\boldsymbol{x}_{n_0}) - f(\boldsymbol{x}^*)} + \left(\frac{A_{pK}}{f(\boldsymbol{x}_{n_0}) - f(\boldsymbol{x}^*)}\right)^2} & \text{if } n > n_0+1. \end{cases} \tag{34}$$

*Proof.* We start by estimating the number of function and gradient computations of Algorithm 2. Except for $t=0$, where $f(\boldsymbol{x}_0)$ and $\nabla f(\boldsymbol{x}_0)$ are computed, for all $t \geq 1$, $f(\boldsymbol{x}_t)$ and $\boldsymbol{g}_t$ has already been computed. So, at iteration $t \geq 1$, there have been $t+1$ function and gradient evaluations devoted to an *initial* check of Line 11. Now, let $k$ denote the total number of function and gradient evaluations until iteration $t \in \mathbb{N}$, i.e., $k-t-1$ is the total number *subsequent* checks of Line 11 and also the number of times that line 12 has been executed. In the worst case, Fact 2.1 guarantees that the $t$th iteration completes after $L_f \leq M_t = \eta^t \tau^{k-t-1} M_0$ occurs; equivalently, after $t+1+\max\{0, \lceil \log_\tau(\eta^{-t}L_f/M_0)\rceil\}$ evaluations are performed.

We turn now to the convergence rate. As in Remark 2.2, let $t_0$ be the smallest nonnegative integer with $\eta^{t_0} M_0 \leq \tau L_f$. The number $n_0$ is chosen to be the smallest nonnegative integer with $t_0 \leq n_0 K + 1$. Let $1 \leq n \leq n_0$. By Remark 2.2, $M_{(n-1)K+1} = \eta^{(n-1)K+1} M_0 > \tau L_f$ and $M_{(n-1)K+1} \geq M_t$ for all $t > (n-1)K$. By Lemma 2.4 and Fact 1.3,

$$\begin{aligned} f(\boldsymbol{x}_{(n-1)K}) - f(\boldsymbol{x}_{nK}) &\geq \rho\left( f(\boldsymbol{x}_{(n-1)K}) - f(\boldsymbol{x}^*) + A_{(n-1)K}, \sum_{k=1}^{K} M_{nK+k} D^2 \right) \\ &\geq \rho(f(\boldsymbol{x}_{(n-1)K}) - f(\boldsymbol{x}^*) + A_{(n-1)K}, K\eta^{(n-1)K+1} M_0 D^2) \\ &\geq f(\boldsymbol{x}_{(n-1)K}) - f(\boldsymbol{x}^*) + A_{(n-1)K} - \frac{K\eta^{(n-1)K+1} M_0 D^2}{2}. \end{aligned} \tag{35}$$

Rearranging (35) shows $f(\boldsymbol{x}_{nK}) - f(\boldsymbol{x}^*) \leq K\eta^{(n-1)K+1} M_0 D^2/2 - A_{(n-1)K}$. Therefore, since $(f(\boldsymbol{x}_t))_{t\in\mathbb{N}}$ is monotonically decreasing (Lemma 2.4), the first case of (34) follows. By the choice of $n_0$, we have $\eta^{n_0 K+1} M_0 \leq \tau L_f$, thus $M_t \leq \tau L_f$ for $t \geq n_0 K$. Let $n \geq n_0 + 1$. Then Lemma 2.4 yields

$$\begin{aligned} f(\boldsymbol{x}_{(n-1)K}) - f(\boldsymbol{x}_{nK}) &\geq \rho\left( f(\boldsymbol{x}_{(n-1)K}) - f(\boldsymbol{x}^*), \sum_{k=1}^{K} M_{(n-1)K+k} D^2 \right) \\ &\geq \rho(f(\boldsymbol{x}_{(n-1)K}) - f(\boldsymbol{x}^*), K\tau L_f D^2). \end{aligned} \tag{36}$$

[3] If $M_0 > \tau L_f$, then $f$ is also $M_0$-smooth, so this assumption is WLOG for notational convenience in the case $\eta = 1$.

and the second case of (35) follows from Lemma 1.5.

□

To interpret the extra terms $(A_t)_{t\in\mathbb{N}}$ in Theorem 2.5, see Remark 1.10.

**Corollary 2.6.** *In the context of Theorem 2.5, suppose that there are no extra activations yielding $A_t \equiv 0$, and hence we deliberately count only one linear minimization on each coordinate per $K$ iterations (hence $K \leq m$). Then, for any $0 < \varepsilon \leq K\tau L_f D^2/2$, the primal gap $f(\boldsymbol{x}_{nK}) - f(\boldsymbol{x}^*) \leq \varepsilon$ is guaranteed after at most $m(n_0 + \frac{2K\tau L_f D^2}{\varepsilon})$ LMO calls and computation of at most $t + 1 + \max\{0, \lceil \log_\tau(\eta^{-t} L_f/M_0)\rceil\}$ function and gradient evaluations.*

**Remark 2.7.** Under stricter assumptions, one can achieve linear convergence by following the template [6, Section 2.2.1] from the penultimate inequality in the proof of Lemma 2.4.

## 2.2 Short-steps with convex objectives

In this section, we consider the short step rule of Remark 1.6

$$\gamma_t^i = \operatorname*{Argmin}_{\gamma\in[0,1]} \left( -\gamma G_i(\boldsymbol{x}_t) + \gamma^2 \frac{L_f}{2} \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2 \right) = \min\left\{ \frac{G_i(\boldsymbol{x}_t)}{L_f \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2} 1 \right\}, \tag{short}$$

which requires an upper bound on $L_f$. For this price, Short-Step BCFW (Algorithm 3) becomes easier to parallelize in lines 5–8, foregoes any function evaluations, and requires only one gradient evaluation per iteration.

**Algorithm 3** Block-Coordinate Frank-Wolfe (BCFW) with Short Steps

**Require:** Function $f\colon \bigtimes_{i\in I} C_i \to \mathbb{R}$, gradient $\nabla f$, point $\boldsymbol{x}_0 \in \bigtimes_{i\in I} C_i$, linear minimization oracles $(\mathrm{LMO}_i)_{i\in I}$

1: **for** $t = 0, 1$ **to** $\dots$ **do**
2: Choose a nonempty block $I_t \subset I$
3: $\boldsymbol{g}_t \leftarrow \nabla f(\boldsymbol{x}_t)$
4: **for** $i = 1$ **to** $m$ **do**
5: **if** $i \in I_t$ **then**
6: $\boldsymbol{v}_t^i \leftarrow \mathrm{LMO}_i(\boldsymbol{g}_t^i)$
7: $\gamma_t^i \leftarrow \min\left\{1, \frac{\langle \boldsymbol{g}_t^i \mid \boldsymbol{x}_t^i - \boldsymbol{v}_t^i\rangle}{L_f \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2}\right\}$
8: $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i + \gamma_t^i(\boldsymbol{v}_t^i - \boldsymbol{x}_t^i)$
9: **else**
10: $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i$
11: **end if**
12: **end for**
13: **end for**

**Corollary 2.8.** *Let $\bigtimes_{i\in I} C_i \subset \boldsymbol{\mathcal{H}}$ be a product of $m$ nonempty compact convex sets, let $D$ be the diameter of $\bigtimes_{i\in I} C_i$, let $f\colon \boldsymbol{\mathcal{H}} \to \mathbb{R}$ be convex and $L_f$-smooth, let $\boldsymbol{x}^*$ be a solution to (1), and for every nonempty $J \subset I$ let $G_J$ be given by (7). In the setting of Algorithm 3, suppose that $K$ satisfies Assumption 1.1, and set $A_t = \sum_{k=1}^{K-1} G_{I_{t+k-1}\cap(I_{t+k}\cup\cdots\cup I_{t+K-1})}(\boldsymbol{x}_{t+k}) \geq 0$. Then,*

$$(\forall n \in \mathbb{N}) \quad f(\boldsymbol{x}_{nK}) - f(\boldsymbol{x}^*) \leq \begin{cases} \dfrac{KL_fD^2}{2} - A_0 & \text{if } n = 1 \\ \dfrac{2KL_fD^2}{n - 1 + \sum_{p=1}^{n} \frac{2A_{pK}}{f(\boldsymbol{x}_1)-f(\boldsymbol{x}^*)} + \left(\frac{A_{pK}}{f(\boldsymbol{x}_1)-f(\boldsymbol{x}^*)}\right)^2} & \text{if } n \geq 2. \end{cases} \tag{37}$$

*Furthermore Algorithm 3 requires one gradient evaluation per iteration.*

*Proof.* This follows from the fact that Algorithm 3 can produce the same sequence of iterates as Algorithm 2: by initializing Algorithm 2 with $M_0 = L_f$ and $\eta = 1$, we have that Fact 2.1 guarantees the condition in Line 11 does not need to be checked. Hence, this case of Algorithm 2 coincides with Algorithm 3 and we achieve convergence from Theorem 2.5 for all $\tau > 1$; taking the limit as $\tau \searrow 1$ yields (37). Clearly, Algorithm 3 requires one gradient evaluation per iteration. □

We note that both the convergence rate and the prefactor of Corollary 2.8 match the non-block version ($K = 1$) [6, Theorem 2.2].

## 3 Nonconvex objective functions

In this section, we consider Algorithm 3 under Assumption 1.1 on nonconvex objective functions with $L_f$-Lipschitz continuous gradients. Since (29) only holds for smooth and convex functions, a different progress lemma which relies on the traditional smoothness inequality (5) is derived. We begin with a blockwise descent lemma.

**Lemma 3.1.** *Let $\bigtimes_{i\in I} C_i \subset \boldsymbol{\mathcal{H}}$ be a product of $m$ nonempty compact convex sets and let $f\colon \boldsymbol{\mathcal{H}} \to \mathbb{R}$ be $L_f$-smooth on $\bigtimes_{i\in I} C_i$. In the setting of Algorithm 3, $(f(\boldsymbol{x}_t))_{t\in\mathbb{N}}$ is monotonically decreasing, and*

$$(\forall t \in \mathbb{N}) \quad f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1}) \geq \frac{\langle \nabla f(\boldsymbol{x}_t) \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle}{2} \geq \frac{L_f\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2}{2}. \tag{38}$$

*Proof.* By (short), for every $i \in I_t$, $L_f\|\boldsymbol{x}_t^i - \boldsymbol{v}_t^i\|\gamma_t^i \leq G_i(\boldsymbol{x}_t)$, so

$$\langle \nabla f(\boldsymbol{x}_t) \mid \boldsymbol{x}_t^i - \boldsymbol{x}_{t+1}^i\rangle = \gamma_t^i G_i(\boldsymbol{x}_t) \geq (\gamma_t^i)^2 L_f\|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2 = L_f\|\boldsymbol{x}_t^i - \boldsymbol{x}_{t+1}^i\|^2. \tag{39}$$

Summing (39) for all $i \in I_t$ and using $\boldsymbol{x}_t^i = \boldsymbol{x}_{t+1}^i$ for $i \notin I_t$ (Line 10), we obtain

$$\langle \nabla f(\boldsymbol{x}_t) \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle \geq L_f\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2. \tag{40}$$

We combine this with the smoothness inequality (5) to derive the claim:

$$\begin{aligned} f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1}) &\geq \langle \nabla f(\boldsymbol{x}_t) \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{L_f}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \\ &\geq \frac{\langle \nabla f(\boldsymbol{x}_t) \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle}{2} \\ &\geq \frac{L_f}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2. \end{aligned} \tag{41}$$

□

**Lemma 3.2** (Progress bound via smoothness (5))**.** *Let $\bigtimes_{i \in I} C_i \subset \boldsymbol{\mathcal{H}}$ be a product of $m$ nonempty compact convex sets, let $D$ be the diameter of $\bigtimes_{i \in I} C_i$, let $f \colon \bigtimes_{i \in I} C_i \to \mathbb{R}$ be such that $\nabla f$ is $L_f$-Lipschitz continuous on $\bigtimes_{i \in I} C_i$, let $G_I$ be given by (8), and let $t \in \mathbb{N}$. In the setting of Algorithm 3, suppose that $K$ satisfies Assumption 1.1, and set $A_t = \sum_{k=0}^{K-1} G_{(I_{t+k} \cup \cdots \cup I_{t+K-1}) \cap I_{t+k-1}}(\boldsymbol{x}_{t+k}) \geq 0$. Then*

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+K}) \geq \frac{\rho(G_I(\boldsymbol{x}_t) + A_t, K L_f D^2)}{2}. \tag{42}$$

*Proof.* For any iteration $t$, we have by smoothness (5)

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1}) \geq \langle \boldsymbol{g}_t \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{L_f \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2}{2}. \tag{43}$$

By Lemma 3.1 and Lipschitz continuity of gradient, we also have

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1}) \geq \frac{L_f \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2}{2} \geq \frac{\|\boldsymbol{g}_t - \boldsymbol{g}_{t+1}\|^2}{2L_f} \geq \frac{\|\boldsymbol{g}_t^{I \setminus I_t} - \boldsymbol{g}_{t+1}^{I \setminus I_t}\|^2}{2L_f}. \tag{44}$$

The sum of (43) and (44) is

$$2(f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1})) \geq \langle \boldsymbol{g}_t \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{L_f \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2}{2} + \frac{\|\boldsymbol{g}_t^{I \setminus I_t} - \boldsymbol{g}_{t+1}^{I \setminus I_t}\|^2}{2L_f}. \tag{45}$$

Summing (45) from $t$ to $t + K - 1$, invoking Lemma 1.9(ii), then dividing by 2 yields (42). □

We are ready to provide convergence for nonconvex functions. Due to lack of optimality guarantees for nonconvex functions, a typical result for Frank-Wolfe algorithms states that the algorithm produces a point with arbitrarily small F-W gap [6, 22], this is closely related to stationarity.

**Theorem 3.3** (Nonconvex convergence)**.** *Let $\bigtimes_{i \in I} C_i \subset \boldsymbol{\mathcal{H}}$ be a product of $m$ nonempty compact convex sets with diameter $D$. Let $f \colon \boldsymbol{\mathcal{H}} \to \mathbb{R}$ be such that $\nabla f$ is $L_f$-Lipschitz continuous on $\bigtimes_{i \in I} C_i$. Let $G_I$ be given by (8). In the setting of Algorithm 3, suppose that $K$ satisfies Assumption 1.1, set $H_0 = f(\boldsymbol{x}_0) - \inf_{\boldsymbol{x} \in \bigtimes_{i \in I} C_i} f(\boldsymbol{x})$, and for every $n \in \mathbb{N}$ set*

$$A_n = \sum_{k=1}^{K-1} G_{I_{n+k-1} \cap (I_{n+k} \cup \cdots \cup I_{n+K-1})}(\boldsymbol{x}_{n+k}) \geq 0. \tag{46}$$

*Then, for every* $n \in \mathbb{N} \setminus \{0\}$,

$$\min_{0\le p\le n-1} G_I(\boldsymbol{x}_{pK}) \le \frac{1}{n}\sum_{p=0}^{n-1} G_I(\boldsymbol{x}_{pK}) \le \begin{cases} \frac{2H_0-\sum_{p=0}^{n-1} A_{pK}}{n} + \frac{KL_fD^2}{2} & \text{if } n \le \frac{2H_0}{KL_fD^2} \\ 2D\sqrt{\frac{H_0KL_f}{n}} - \frac{\sum_{p=0}^{n-1} A_{pK}}{n} & \text{otherwise.} \end{cases} \tag{47}$$

*In consequence, there exists a subsequence* $(n_k)_{k\in\mathbb{N}}$ *such that* $G_I(\boldsymbol{x}_{n_kK}) \to 0$*, and every accumulation point of* $(\boldsymbol{x}_{n_kK})_{k\in\mathbb{N}}$ *is a stationary point of* (1).

*Proof.* By telescoping the result of Lemma 3.2 over multiples of $K$, then using subadditivity (12),

$$2H_0 \ge 2(f(x_0)-f(x_{nK})) \ge \sum_{p=0}^{n-1} \rho(G_I(\boldsymbol{x}_{pK})+A_{pK}, KL_fD^2) \ge \rho\left(\sum_{p=0}^{n-1} G_I(\boldsymbol{x}_{pK}) + A_{pK}, nKL_fD^2\right).$$

Observe that, for $x, y \ge 0$ and $b > 0$, we have $y \ge \rho(x, b)$ if and only if $x \le \rho_+^{-1}(y, b) \coloneqq y + \frac{b}{2}$ if $y \ge \frac{b}{2}$; $\sqrt{2by}$ if $y \le \frac{b}{2}$. Therefore,

$$\sum_{p=0}^{n-1} G_I(\boldsymbol{x}_{pK}) + A_{pK} \le \rho_+^{-1}(2H_0, nKL_fD^2) = \begin{cases} 2H_0 + \frac{nKL_fD^2}{2} & \text{if } 2H_0 \ge nKL_fD^2 \\ 2D\sqrt{H_0nKL_f} & \text{otherwise.} \end{cases} \tag{48}$$

Dividing (48) by $n$ and rearranging yields (47). □

**Remark 3.4.** If $f$ is convex and $\nabla f$ is Lipschitz-continuous on $\bigtimes_{i\in I} C_i$, yet $f$ is not extendable to a smooth function on $\boldsymbol{\mathcal{H}}$ (see Fact 2.1), then one can nonetheless achieve $\mathcal{O}(K/t)$ convergence, by applying an argument similar to the proof of Theorem 2.5, replacing Lemma 2.4 with Lemmas 3.1 and 3.2, yielding the same rate with a worse constant.

## 4 Numerical Experiments

In this section we examine different block selection strategies covered by Assumption 1.1 on some simple experiments. For each experiment, we run $10,000$ iterations of Algorithm 3 using `FrankWolfe.jl` (v3.3) [4] in Julia 1.8.5. Computations were performed on one node allocated 3 GB RAM on an Intel Xeon Gold 6246 machine with 3.3 GHz CPU speed, running Linux managed by Slurm and no concurrent jobs on the node. To ensure feasibility of the initial iterate, for every $i \in I$, we generate an initial vector $\boldsymbol{c}^i \in \mathcal{H}^i$ with normally distributed entries of mean $0$ and standard deviation $1$, then set $\boldsymbol{x}_0^i = \mathrm{LMO}_i(\boldsymbol{c}^i)$. Within each experiment, the only thing that changes is how the blocks $(I_t)_{t\in\mathbb{N}}$ are selected. We compare block selection strategies newly allowed by Assumption 1.1 to the following techniques (e.g., in [2, 21]) covered by our results:

(i) *Full* activation: $I_t = I$.

(ii) *Cyclic* activation: $I_t = \{t\} \pmod m$.

(iii) *P-Cyclic* activation: $I_t = \sigma_{\lfloor t/m \rfloor}(t \pmod m)$, where for every cycle over $m$ iterations, $\sigma_{\lfloor t/m \rfloor}$ is a uniformly random permutation of $\{1, \dots, m\}$.

Section 4.1 studies a convex problem with 2 components, and Section 4.2 examines a nonconvex problem with many components. In line with Theorems 2.5 and 3.3, our optimality criterion is the primal gap for Section 4.1 and minimal F-W gap for Section 4.2.

In addition to plotting our optimality criterion against iterations and time, we also plot against the number of evaluations for the most computationally-intensive LMO (spectrahedron LMO for Section 4.1; nuclear norm ball LMO for Section 4.2). The expensive-LMO count is a more reproducible proxy for time in our experiments, since it correlated with time used for all algorithms, and it was the dominant cost of even a full F-W iteration. From some perspective, these plots are unfair to the full/cyclic selection schemes, since they are forced to activate the most expensive LMO at a fixed rate and our new methods have more flexibility to re-activate cheaper components; however, flexibility in activation is precisely the point here, and until this work it was unclear if such reactivations would provide progress for BCFW at all.

### 4.1 Experiment 1: Intersection problem

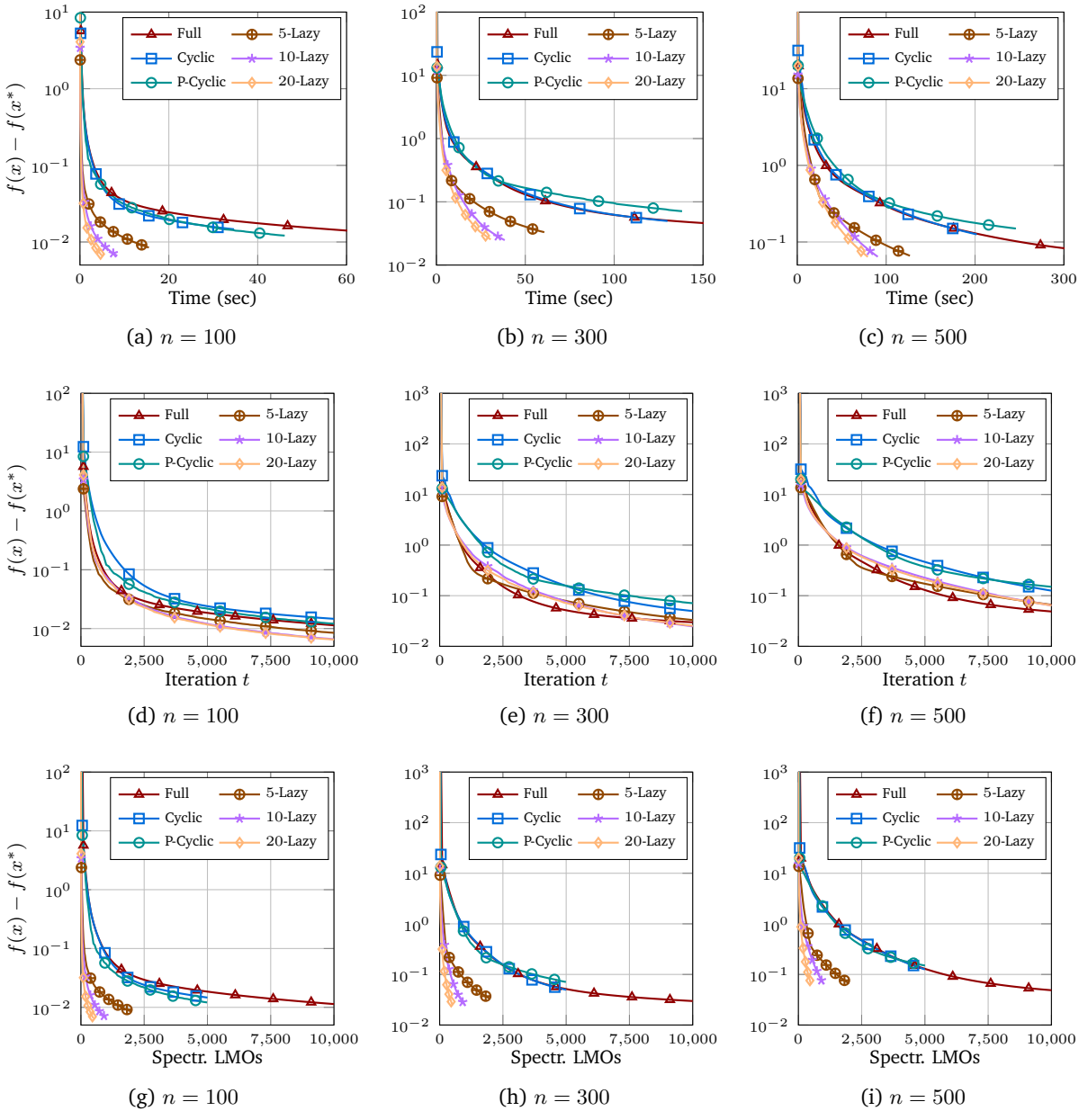
The goal is to find a matrix $x \in \mathbb{R}^{n \times n}$ in the intersection of the hypercube $C_1 = [-1, 1/n]^{n \times n}$ and the spectraplex $C_2 = \{x \in \mathbb{R}^{n \times n} \mid x \succeq 0, \operatorname{Trace}(x) = 1\}$ for various values of $n$. The convex sets are selected to have a thin intersection, and hence the minimal value of

$$\underset{\boldsymbol{x} \in C_1 \times C_2}{\text{minimize}} \frac{1}{2} \|\boldsymbol{x}^1 - \boldsymbol{x}^2\|^2 \tag{49}$$

is zero. Problem (49) is convex, with smoothness constant $L_f = 2$. In this problem, the spectrahedral linear minimization oracle, $\mathrm{LMO}_2$, is far more expensive than $\mathrm{LMO}_1$. So, for this experiment we compare the traditional BCFW activations (i)–(iii) with the following "$q$-lazy" scheme which is newly allowed for BCFW by Assumption 1.1 (with $K = q$) and has improved computational performance in proximal algorithms [13]:

$$(\forall t \in \mathbb{N}) \quad I_t = \begin{cases} \{1, 2\} & \text{if } t \equiv 0 \mod q; \\ \{1\} & \text{otherwise.} \end{cases} \tag{50}$$

We run 20 instances of this problem on random initializations, and the averaged results are shown in Figure 1. Even though using $q$-lazy activation is computationally cheaper on average, the per-iteration progress is still competitive with that of full activation and the existing methods. However, since they compute $\mathrm{LMO}_2$ at a much lower rate, these activation strategies also have a faster per-iteration computation time.




(a) $n = 100$ (b) $n = 300$ (c) $n = 500$

(d) $n = 100$ (e) $n = 300$ (f) $n = 500$

(g) $n = 100$ (h) $n = 300$ (i) $n = 500$

Fig. 1: Continued results of Experiment 1 (Section 4.1) displaying the primal gap $f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)$ versus time, iteration, and spectrahedral LMO count for problems with $n^2$ variables and block-activation strategies according to (i)–(iii) and (50).

### 4.2 Experiment 2: Difference of convex quadratics

The goal is to minimize a difference of convex quadratic functions of two collated matrices in $\mathbb{R}^{n\times n}$, where the submatrices are constrained to an $\ell_\infty$ ball and a nuclear-norm ball respectively. In order to examine the performance of Algorithm 3 when the number of components is large, we split the $\ell_\infty$ constraint into $n$ separate constraints. Hence, we set $C_1 = \cdots = C_n := \left\{x \in \mathbb{R}^n \;\middle|\; \|x\|_\infty \leq 1\right\}$, and $C_{n+1} = \left\{x \in \mathbb{R}^{n\times n} \;\middle|\; \|x\|_{\text{nuc}} \leq 1\right\}$. For $\boldsymbol{x} \in \bigtimes_{i\in I} C_i$ we use $[\boldsymbol{x}]$ to denote the collated $2n \times n$ matrix of its components. For each problem instance, the kernel $A, B$ of each quadratic is generated by projecting a matrix with random normal entries of mean $0$ and standard deviation $1$ onto the set of positive semidefinite matrices. Altogether, we seek to solve the following difference-of-convex problem involving the Frobenius inner product

$$\underset{\boldsymbol{x}\in C_1\times\cdots\times C_{n+1}}{\text{minimize}} \; \frac{1}{2}\Big(\Big\langle [\boldsymbol{x}] \;\Big|\; [\boldsymbol{x}]A \Big\rangle - \Big\langle [\boldsymbol{x}] \;\Big|\; [\boldsymbol{x}]B \Big\rangle\Big). \tag{51}$$

Note the objective function of (51) is smooth and nonseparable. In the experiments we used the Froebenous norm of $A - B$ as smoothness constant $L_f$. For each instance of (51), we verify that $A - B$ is indefinite, hence the objective is also neither convex nor concave. Since the $\ell_\infty$ LMO is far cheaper than the nuclear norm ball LMO [12], similarly to Section 4.1, we consider a family of customized activation strategies that delay evaluating the most expensive operator $\text{LMO}_{n+1}$. In addition, on the "lazy" iterations involving only the LMOs of the $\ell_\infty$ norm ball, we perform a parallel update involving a random subset of $I \setminus \{n+1\}$ of size $p$:

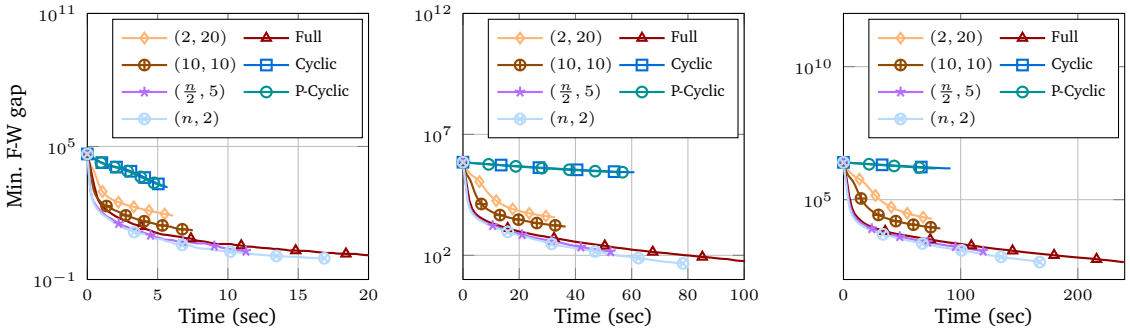
$$(\forall t \in \mathbb{N}) \quad I_t = \begin{cases} I & \text{if } \; t \equiv 0 \pmod q \\ \{i_1, \ldots, i_p\} \subset I \setminus \{n+1\} & \text{otherwise.} \end{cases} \tag{52}$$

Averaged results from $20$ instances of (51) are shown in Figures 2 and 3. Since the problem is nonconvex, we plot the minimal F-W gap observed (see Theorem 3.3). Since full F-W gaps are typically unavailable in BCFW (only partial gaps for the activated blocks $(G_i)_{i\in I_t}$ are computed), iterates were stored during the run of Algorithm 3 and full F-W gaps were computed post-hoc.

Similarly to Section 4.1, new selection strategies allowed by Assumption 1.1 can yield similar per-iteration performance to that of full-activation Frank-Wolfe; furthermore, since the iterations frequently involve the cheaper LMOs, this can yield faster convergence in wall-clock time. However, these results also demonstrate that, if the number of activated components in $I_t$ is too small and the $n+1$st component is activated too infrequently, results may worsen; this is reflected in the cyclic, $P$-cyclic, and $(p, q) = (2, 20)$ results.

## Acknowledgments

This research was partially supported by the DFG Cluster of Excellence MATH+ (EXC-2046/1, project id 390685689) funded by the Deutsche Forschungsgemeinschaft (DFG) and took place on the Research Campus MODAL funded by the German Federal Ministry of Education and Research




(a) $n = 100$ (b) $n = 300$ (c) $n = 500$

Fig. 2: Results of Experiment 2 (Section 4.2) displaying the minimal F-W gap versus time for problems with $n^2$ variables and block-activation strategies according to (i)–(iii) and (52) for values of $(p, q)$ in the left column of each legend.

(BMBF) (fund numbers 05M14ZAM, 05M20ZBM). Part of this work was conducted while SP was visiting Tokyo University via a JSPS International Research Fellowship.

We thank Jannis Halbey, Deborah Hendrych, Gabriele Iommazzo, Dominik Kuzinowicz, Mark Turner, and Elias Wirth for valuable feedback on an early draft of this article.

## References

[1] Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces, 2nd ed. Springer (2017)

[2] Beck, A., Pauwels, E., Sabach, S.: The cyclic block conditional gradient method for convex optimization problems. SIAM J. Optim. **25**(4), 2024–2049 (2015). DOI 10.1137/15M1008397

[3] Beck, A., Tetruashvili, L.: On the convergence of block coordinate descent type methods. SIAM J. Optim. **23**(4), 2037–2060 (2013)

[4] Besançon, M., Carderera, A., Pokutta, S.: FrankWolfe.jl: A high-performance and flexible toolbox for Frank–Wolfe algorithms and conditional gradients. INFORMS J. Comput. **34**(5), 2611–2620 (2022)

[5] Bomze, I., Rinaldi, F., Zeffiro, D.: Projection free methods on product domains. Comput. Optim. Appl. pp. 1–30 (2024)

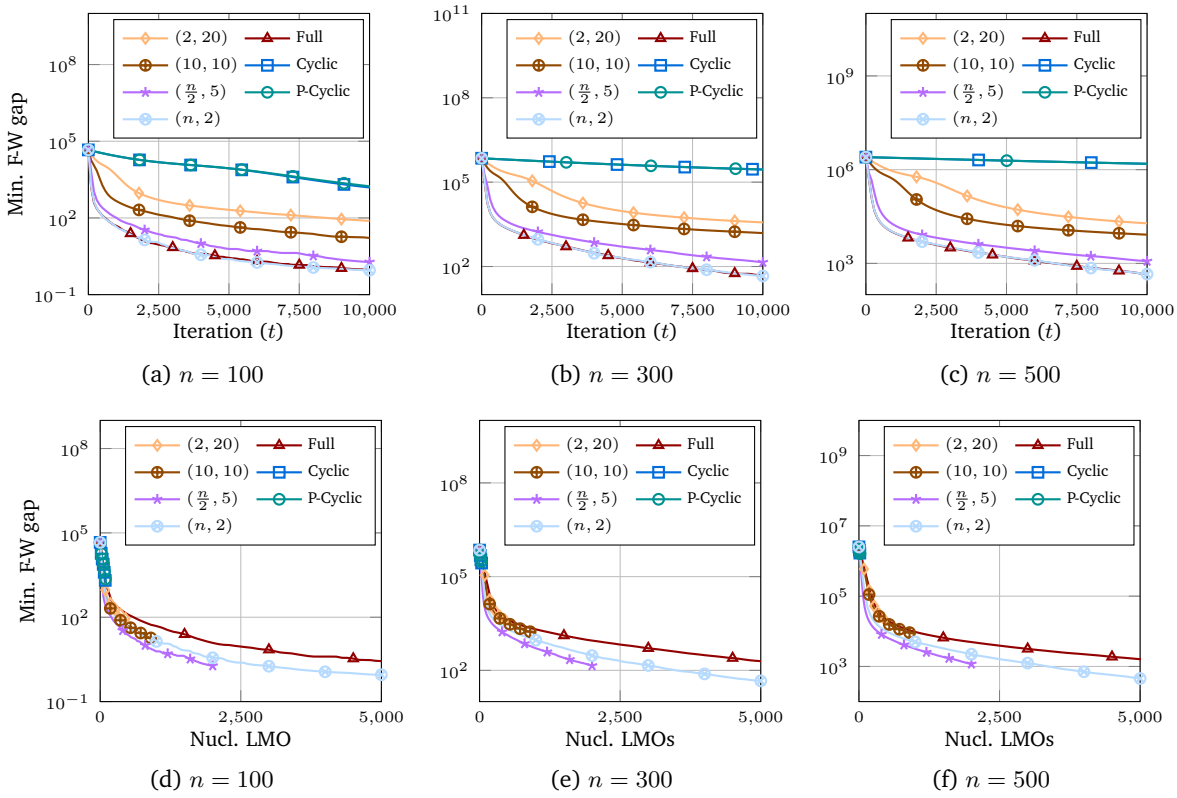
[6] Braun, G., Carderera, A., Combettes, C., Hassani, H., Karbasi, A., Mokhtari, A., Pokutta, S.: Conditional gradient methods. arXiv:2211.14103 (2022)

[7] Braun, G., Pokutta, S.: The matching polytope does not admit fully-polynomial size relaxation schemes. In: Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 837–846. SIAM (2015). DOI 10.1137/1.9781611973730.57

[8] Braun, G., Pokutta, S., Weismantel, R.: Alternating linear minimization: Revisiting von Neumann's alternating projections. arXiv:2212.02933 (2022)

[9] Braun, G., Pokutta, S., Zink, D.: Affine reductions for LPs and SDPs. Math. Program. **173**, 281–312 (2019). DOI 10.1007/s10107-017-1221-9

[10] Carderera, A., Besançon, M., Pokutta, S.: Scalable frank-wolfe on generalized self-concordant functions via simple steps. SIAM J. Optim. **34**(3), 2231–2258 (2024). DOI 10.1137/23M1616789




(a) $n = 100$ (b) $n = 300$ (c) $n = 500$

(d) $n = 100$ (e) $n = 300$ (f) $n = 500$

Fig. 3: Continued results of Experiment 2 (Section 4.2) displaying the minimal F-W gap versus iteration and nuclear norm LMO count for problems with $n^2$ variables and block-activation strategies according to (i)–(iii) and (52) for values of $(p, q)$ in the left column of each legend.


[11] Combettes, C.W., Pokutta, S.: Boosting Frank–Wolfe by chasing gradients. In: Proceedings of the 37th International Conference on Machine Learning (ICML), pp. 2111–2121. PLMR (2020)

[12] Combettes, C.W., Pokutta, S.: Complexity of linear minimization and projection on some sets. Oper. Res. Lett. **49**(4), 565–571 (2021)

[13] Combettes, P.L., Woodstock, Z.C.: A variational inequality model for the construction of signals from inconsistent nonlinear equations. SIAM J. Imaging Sci. **15**(1), 84–109 (2022). DOI 10.1137/21M1420368

[14] Diakonikolas, J., Carderera, A., Pokutta, S.: Locally accelerated conditional gradients. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), vol. 108, pp. 1737–1747. PMLR (2020)

[15] Ding, C.H., Li, T., Jordan, M.I.: Convex and semi-nonnegative matrix factorizations. IEEE Trans. Pattern Anal. Mach. Intell. **32**(1), 45–55 (2008)

[16] Hazan, E., Luo, H.: Variance-reduced and projection-free stochastic optimization. In: Proceedings of the 33rd International Conference on Machine Learning (ICML), vol. 48, pp. 1263–1271. PMLR (2016)

[17] Kelley, C.T.: Iterative Methods for Linear and Nonlinear Equations. No. 20 in Frontiers in Applied Mathematics. SIAM, Philadelphia (1995). DOI 10.1137/1.9781611970944

[18] Lacoste-Julien, S., Jaggi, M., Schmidt, M., Pletscher, P.: Block-coordinate Frank-Wolfe optimization for structural SVMs. In: S. Dasgupta, D. McAllester (eds.) Proceedings of the 30th International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 28, pp. 53–61. PMLR, Atlanta, Georgia, USA (2013)

[19] Osokin, A., Alayrac, J.B., Lukasewitz, I., Dokania, P., Lacoste-Julien, S.: Minding the gaps for block Frank-Wolfe optimization of structured SVMs. In: M.F. Balcan, K.Q. Weinberger (eds.) Proceedings of The 33rd International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 48, pp. 593–602. PMLR, New York, NY, USA (2016)

[20] Ottavy, N.: Strong convergence of projection-like methods in Hilbert spaces. J. Optim. Theory Appl. **56**, 433–461 (1988)

[21] Patriksson, M.: Decomposition methods for differentiable optimization problems over cartesian product sets. Comput. Optim. Appl. **9**, 5–42 (1998)

[22] Pedregosa, F., Negiar, G., Askari, A., Jaggi, M.: Linearly convergent Frank-Wolfe with backtracking line-search. In: International conference on artificial intelligence and statistics, pp. 1–10. PMLR (2020)

[23] Pokutta, S.: The Frank-Wolfe algorithm: a short introduction. Jahresber. Dtsch. Math.-Ver. **126**(1), 3–35 (2024). DOI 10.1365/s13291-023-00275-x

[24] Rothvoss, T.: The matching polytope has exponential extension complexity. J. ACM **64**(6), 41:1–19 (2017). DOI 10.1145/3127497

[25] Shamanskii, V.E.: A modification of Newton's method. Ukran. Mat. Zh. **19**(1), 133–138 (1967)

[26] Srebro, N., Rennie, J., Jaakkola, T.: Maximum-margin matrix factorization. In: L. Saul, Y. Weiss, L. Bottou (eds.) Advances in Neural Information Processing Systems, vol. 17, pp. 1329–1336. MIT Press (2004)

[27] Taylor, A.B., Hendrickx, J.M., Glineur, F.: Smooth strongly convex interpolation and exact worst-case performance of first-order methods. Math. Program. **161**, 307–345 (2017). DOI 10.1007/s10107-016-1009-3

[28] Tseng, P.: Convergence of a block coordinate descent method for nondifferentiable minimization. J. Optim. Theory Appl. **109**, 475–494 (2001)

[29] Wang, Y.X., Sadhanala, V., Dai, W., Neiswanger, W., Sra, S., Xing, E.: Parallel and distributed block-coordinate Frank-Wolfe algorithms. In: M.F. Balcan, K.Q. Weinberger (eds.) Proceedings of The 33rd International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 48, pp. 1548–1557. PMLR, New York, NY, USA (2016)

[30] Woodstock, Z., Pokutta, S.: Splitting the conditional gradient algorithm. arXiv:2311.05381 (2024)