

**manuscript No.**  
(will be inserted by the editor)

---

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

Gábor Braun · Jannis Halbey · Sebastian  
Pokutta · Zev Woodstock

the date of receipt and acceptance should be inserted later

**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. This appears to be the first result on BCFW addressing the setting of nonconvex objective functions with Lipschitz-continuous gradients and no additional assumptions. This analysis newly allows 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, parallel updates

**Mathematics Subject Classification (2000)** 49M27, 49M37, 65K05, 90C26, 90C30

## 1 Introduction

Given a smooth function  $f$  that maps from a finite Cartesian product of  $m$  real Hilbert spaces  $\mathcal{H} := \bigoplus_{i=1}^m \mathcal{H}_i$  to  $\mathbb{R}$  and a product of nonempty compact convex

---

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

Zev Woodstock (`woodstzc@jmu.edu`), Corresponding author  
James Madison University, Harrisonburg, VA, USA;  
The work of ZW was supported by the National Science Foundation under grant DMS-2532423.

subsets  $\times_{i=1}^m C_i \subset \mathcal{H}$  with  $C_i \subseteq \mathcal{H}_i$ , we seek to solve the following problem

$$\underset{\mathbf{x} \in C_1 \times \dots \times C_m}{\text{minimize}} \quad f(\mathbf{x}), \quad (1)$$

which has applications in matrix factorization, support vector machine training, sequence labeling, intersection verification, and more [5, 8, 16, 19, 20, 27, 31, 32]. Frank-Wolfe (F-W), also known as *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* LMO <sub>$C$</sub>  for a compact convex set  $C \subseteq \mathcal{H}$ , computes for any linear objective  $c \in \mathcal{H}$ , a point in  $\text{Argmin}_{\mathbf{x} \in C} \langle c | \mathbf{x} \rangle$ . The oracle approach is advantageous for problems such as the maximum matching problem [7, 25], 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  $C_1 \times \dots \times C_m$  is evaluated. This step can cause a computational bottleneck, since

$$\text{LMO}_{C_1 \times \dots \times C_m}(\mathbf{x}^1, \dots, \mathbf{x}^m) = (\text{LMO}_{C_1} \mathbf{x}^1, \dots, \text{LMO}_{C_m} \mathbf{x}^m), \quad (2)$$

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, 19, 20, 31]. 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 [15], 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 [22], 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, [22] 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  $i(t)$  of the component updated at iteration  $t \in \mathbb{N}$ , satisfies

$$\{i(t), \dots, i(t + K - 1)\} = \{1, \dots, m\}. \quad (3)$$

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, [19] 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 [20] 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, 31] 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  $\text{LMO}_{C_i}$ , 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 [22], 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<sup>1</sup>. 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 step sizes, also significantly improving convergence rates in certain settings.

However, outside of full-updates, it appears that only [31] allows for block-sizes larger than 1. In particular, for a fixed block-size  $p$ , the results in [31] 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  $(\text{LMO}_{C_i})_{i \in \{1, \dots, m\}}$  require similar levels of computational effort, there appear to be no other good options for leveraging parallelism. In particular, regardless of the step sizes 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 par-

---

<sup>1</sup> Although [22, Section 4] also contains results for more general selection schemes of  $I_t$ , they do not apply to the Frank-Wolfe setting (see [22, Table 1]).

ticular, 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. Newly-allowed update methods for BCFW will also be compared to existing update techniques, including the essentially-cyclic singleton-update scheme (3) that also allows for the design of cost-aware update strategies without parallel block updates.

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 [21].

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

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

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.

- (i) With the  $I_t$  singletons, this becomes the *essentially cyclic* selection scheme (3) of [22, 30]; 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, 22].
- (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, 20]).
- (iv) If the operators  $(\text{LMO}_{C_i})_{i \in \{1, \dots, m\}}$  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.<sup>2</sup>
- (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

<sup>2</sup> 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 [18, 26].

$t + K - 1$ , 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 3.1 proves that after  $t$  iterations, the algorithm is guaranteed to produce a point with *Frank-Wolfe gap* (a quantity closely related to 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 2.2); in consequence, Corollary 2.2 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 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  $I := \{1, 2, \dots, m\}$ , and we consider the direct sum  $\mathcal{H} := \bigoplus_{i \in I} \mathcal{H}_i$  of real Hilbert spaces  $\mathcal{H}_i$ . We denote points of  $\mathcal{H}$  by bold letters, and components in the direct sum by upper indices, i.e.,  $\mathbf{x} = (x^1, x^2, \dots, x^m) \in \mathcal{H}$  with  $x^i \in \mathcal{H}_i$ . The inner product on  $\mathcal{H}$  is  $\langle \mathbf{x} | \mathbf{y} \rangle_{\mathcal{H}} = \sum_{i \in I} \langle x^i | y^i \rangle_{\mathcal{H}_i}$ , yielding the norm identity  $\|\mathbf{x} - \mathbf{y}\|_{\mathcal{H}}^2 = \sum_{i \in I} \|x^i - y^i\|_{\mathcal{H}_i}^2$ . For notational convenience, we treat the  $\mathcal{H}_i$  as orthogonal subspaces of  $\mathcal{H}$ , in particular,  $\mathbf{x} = \sum_{i \in I} x^i$ . We will omit the subscripts  $\mathcal{H}, \mathcal{H}_i$  from norms and scalar products; this will not cause

ambiguity as all are restrictions of the ones on  $\mathcal{H}$ . For  $J \subseteq I$ , let  $\mathbf{x}^J := \sum_{i \in J} \mathbf{x}^j$  be the part of  $\mathbf{x}$  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  $\times_{i \in J} C_i$  be the set of points  $\mathbf{x} \in \mathcal{H}$  with  $\mathbf{x}^i \in C_i$  for all  $i \in J$  and  $\mathbf{x}^i = 0$  for  $i \notin J$ . Let  $D_J$  be the diameter of  $\times_{i \in J} C_i$  (treated as a subset of  $\bigoplus_{i \in J} \mathcal{H}_i \subset \mathcal{H}$ ). We shall use the simplified notation  $D_i := D_{\{i\}}$  and  $D := D_I$ .

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

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

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

$$(\forall \mathbf{x}, \mathbf{y} \in C) \quad \langle \nabla f(\mathbf{x}) | \mathbf{y} - \mathbf{x} \rangle \leq f(\mathbf{y}) - f(\mathbf{x}). \quad (6)$$

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

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

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

$$G_I(\mathbf{x}) := \sup_{\mathbf{v} \in \times_{i \in I} C_i} \langle \nabla f(\mathbf{x}) | \mathbf{x} - \mathbf{v} \rangle = \sum_{i \in I} G_i(\mathbf{x}). \quad (8)$$

Note that, for every  $\mathbf{x} \in \times_{i \in I} C_i$  and every  $J \subset I$ , we have  $G_J(\mathbf{x}) \geq 0$ . The FW gap vanishes at a solution of (1) in the following sense [6]

$$\mathbf{x} \text{ is a stationary point of } \underset{\mathbf{x} \in \times_{i \in I} C_i}{\text{minimize}} f(\mathbf{x}) \iff \begin{cases} \mathbf{x} \in \times_{i \in I} C_i \\ G_I(\mathbf{x}) \leq 0. \end{cases} \quad (9)$$

Before proceeding further, we gather several useful results.

**Lemma 1.1** *Let  $(C_i)_{i \in I}$  be nonempty compact convex subsets of real Hilbert spaces  $(\mathcal{H}_i)_{i \in I}$ , let  $f: \times_{i \in I} C_i \rightarrow \mathbb{R}$  be convex, let  $J \subset I$  be nonempty, and let  $G_J$  be given by (7). Then,*

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

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

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

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

**Fact 1.2** *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]:*

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

**Lemma 1.2** *Let  $x, c$  be nonnegative numbers and let  $b$  be a positive number. Then,*

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

*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,

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

$\square$

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

**Lemma 1.3** *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*

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

*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  $0 < h_{t+1} \leq \dots \leq h_1$ . So,

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

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

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

and rearranging (17) completes the result.  $\square$

## 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, 19, 22] arise as special cases.

---

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

---

**Require:** Function  $f: \times_{i \in I} C_i \rightarrow \mathbb{R}$ , gradient  $\nabla f$ , point  $\mathbf{x}_0 \in \times_{i \in I} C_i$ , linear minimization oracles  $(\text{LMO}_i)_{i \in I}$

- 1: **for**  $t = 0, 1$  **to**  $\dots$  **do**
- 2:   Choose a nonempty block  $I_t \subset I$
- 3:    $\mathbf{g}_t \leftarrow \nabla f(\mathbf{x}_t)$
- 4:   **for**  $i = 1$  **to**  $m$  **do**
- 5:     **if**  $i \in I_t$  **then**
- 6:        $\mathbf{v}_t^i \leftarrow \text{LMO}_i(\mathbf{g}_t^i)$
- 7:        $\gamma_t^i \leftarrow$  Step size parameter (see also Sections 2, 3)
- 8:        $\mathbf{x}_{t+1}^i \leftarrow \mathbf{x}_t^i + \gamma_t^i(\mathbf{v}_t^i - \mathbf{x}_t^i)$
- 9:     **else**
- 10:       $\mathbf{x}_{t+1}^i \leftarrow \mathbf{x}_t^i$
- 11:     **end if**
- 12:   **end for**
- 13: **end for**

---

*Remark 1.1* For  $L_f$ -smooth objective functions  $f$ , the smoothness inequality (5) and Line 8 of Algorithm 1 yield

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

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, 14]:

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

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.

This work focuses on using short-step step sizes that rely on  $L_f$  (or its estimator in Section 2.1). In contrast, the following examples demonstrate that Algorithm 1 need not converge using componentwise analogues of classical FW step sizes.

*Example 1.1 (Non-convergent componentwise line search)* Frank-Wolfe methods at a point  $\mathbf{x}_t$  with vertex  $\mathbf{v}_t$  are commonly known to converge where step sizes are selected by a line search, i.e.,  $\gamma_t = \operatorname{Argmin}_{\gamma \in [0,1]} f(\mathbf{x}_t + \gamma(\mathbf{v}_t - \mathbf{x}_t))$  [6]. However, when line search step sizes are chosen componentwise, namely via

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

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

*Example 1.2 (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  $\gamma_t^i = \min\{1, G_i(\mathbf{x}_t)/\beta_i \|\mathbf{v}_t^i - \mathbf{x}_t^i\|^2\}$  allow for larger step sizes [2], since  $\beta_i \leq L_f$ . However, in Example 1.1,  $\beta_1 = \beta_2 = 2 \neq 4 = L_f$ , i.e.,  $\gamma_t^i$  is the same as in Example 1.1. Therefore, using Algorithm 1 with  $I_t = \{1, 2\}$ , this short-step variant may produce the same iterates as Example 1.1, which do not get close to the optimal solution.

The following technical lemma is for combining with inequalities of the form  $f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}) \geq \tau P_t$  ( $\tau > 0$ ) 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 1.4** Let  $\times_{i \in I} C_i \subset \mathcal{H}$  be a finite product of nonempty compact convex sets  $C_i$ , let  $D$  be the diameter of  $\times_{i \in I} C_i$ , let  $f: \times_{i \in I} C_i \rightarrow \mathbb{R}$  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) and set  $\mathbf{v}_t^J = \text{LMO}_J(\mathbf{g}_t)$ . In the setting of Algorithm 1, for every  $t \in \mathbb{N}$  and  $J \subset I$ , set  $\mathbf{v}_t^J = \text{LMO}_J(\mathbf{g}_t)$ , set

$$P_t = \langle \mathbf{g}_t | \mathbf{x}_t - \mathbf{x}_{t+1} \rangle - \frac{M_{t+1} \|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2}{2} + \frac{\|\mathbf{g}_t^{I \setminus I_t} - \mathbf{g}_{t+1}^{I \setminus I_t}\|^2}{2M_{t+1}}, \quad (20)$$

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

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

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

- (i)  $(\forall J \subseteq I \setminus I_t) P_t \geq \rho(|G_{I_t \cup J}(\mathbf{x}_t) - \langle \mathbf{g}_{t+1}^J | \mathbf{x}_t^J - \mathbf{v}_t^J \rangle|, M_{t+1} \|\mathbf{x}_t^{I_t \cup J} - \mathbf{v}_t^{I_t \cup J}\|^2)$ .
- (ii)  $\sum_{k=0}^{K-1} P_{t+k} \geq \rho(G_{I_t \cup \dots \cup I_{t+K-1}}(\mathbf{x}_t) + A_t, \sum_{k=1}^K M_{t+k} D^2)$ .

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

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

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

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

If  $\langle \mathbf{g}_t^i | \mathbf{x}_t^i - \mathbf{v}_t^i \rangle \leq M_{t+1} \|\mathbf{x}_t^i - \mathbf{v}_t^i\|^2$ , then  $\gamma_t^i = \langle \mathbf{g}_t^i | \mathbf{x}_t^i - \mathbf{v}_t^i \rangle / (M_{t+1} \|\mathbf{x}_t^i - \mathbf{v}_t^i\|^2)$ , so

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

Summing up (22) for  $i \in I_t$  and using subadditivity of  $\rho$  in Fact 1.2, we obtain (i) for  $J = \emptyset$ . To show (i) for arbitrary  $J \subseteq I \setminus I_t$ , we use an additional norm inequality, then (i) for  $J = \emptyset$ , and Lemma 1.2 (with  $c = 0$ ), followed by

subadditivity and monotonicity of  $\rho$  (12):

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

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

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

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  $\rho$ :

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

where, using Line 10 of Algorithm 1 and the convention that  $G_\emptyset(\mathbf{x}_{t+K-1}) = 0$ ,

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

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

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

which completes the proof.  $\square$

*Remark 1.2 (Interpretation of the gaps  $A_t$  in Lemma 1.4)* For every  $t \in \mathbb{N}$ , each of the following summands in the lower bound of Lemma 1.4

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

is a partial Frank-Wolfe gap for components that are updated more than once between iterations  $t$  and  $t + K - 1$ . Via Lemma 1.1, for each collection of reactivated components  $J \subset I$ , if  $f$  is convex then each summand can be bounded by

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

As will be seen in Sections 2 and 3, the gaps  $A_t$  contribute to faster convergence, and they may explain the favorable behavior observed in Section 4. However, in general,  $A_t$  may not always be strictly positive. Hence, we do not know how to utilize these gaps 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 [30]), these gaps 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  $\mathcal{O}(K/t)$  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 2.2 and Corollary 2.2) 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  $x_t$  and the next iterate  $x_{t+1}$ , at the expense of extra gradient and/or function evaluations [2, 23, 24]. 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 [17] and was later shown to characterize convex smooth interpolability [29].

**Fact 2.1 ([17])** *Let  $f: \mathcal{H} \rightarrow \mathbb{R}$  be convex and  $L_f$ -smooth on  $\mathcal{H}$ . Then,*

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

The interpolability result [29] implies that a function  $f$  satisfying (29) for all  $\mathbf{x}, \mathbf{y}$  in a convex set has an extension to a convex  $L_f$ -smooth function on  $\mathcal{H}$ , and therefore, for simplicity of presentation, our results in this section assume that  $f$  is already extended, i.e., convex and  $L_f$ -smooth on  $\mathcal{H}$ . For objective functions that cannot be extended to  $\mathcal{H}$ , see Remark 3.1.

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 2.1). 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 [23] or another consequence as in [24], Algorithm 2 checks (29). Note that Algorithm 2 can be viewed as a version of Algorithm 1 where  $\gamma_t^i$  is computed with an adaptive subroutine (see also Remark 1.1).

*Remark 2.1* 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  $M_{t+1} \geq L_f$ , potentially overshooting by a factor of  $\tau$ . Hence  $M_{t+1}$  can only be at least  $\tau L_f$  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,

$$M_t = \eta^t M_0 > \tau L_f \quad 1 \leq t < t_0 \quad (30)$$

$$M_t \leq \tau L_f \quad t \geq t_0. \quad (31)$$

This is consistent with the behavior of other adaptive constants in similar works [23, 24]: Unless  $M_0$  is initialized above  $\tau L_f$ ,  $M_t$  underestimates  $\tau L_f$ .

*Remark 2.2* Even though the adaptive step size strategy in Algorithm 2 requires extra function and gradient evaluations (Lines 11–16), the LMOs are

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

---

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

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

---

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 2.1 (Progress bound via smoothness and convexity (29))** *Let  $\times_{i \in I} C_i \subset \mathcal{H}$  be a product of  $m$  nonempty compact convex sets, let  $D$  be the diameter of  $\times_{i \in I} C_i$ , let  $f: \mathcal{H} \rightarrow \mathbb{R}$  be convex and  $L_f$ -smooth, let  $\rho$  be given by (11), let  $\mathbf{x}^*$  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  $A_t = \sum_{k=1}^{K-1} G_{I_{t+k-1} \cap (I_{t+k} \cup \dots \cup I_{t+K-1})}(\mathbf{x}_{t+k}) \geq 0$ . Then  $(f(\mathbf{x}_t))_{t \in \mathbb{N}}$  is monotonically decreasing and*

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

*Proof.* Recall from Remark 2.1 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  $\mathbf{x}_t^{I \setminus I_t} = \mathbf{x}_{t+1}^{I \setminus I_t}$ .

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

where  $P_t$  is the same as in Lemma 1.4. Monotonicity of  $(f(\mathbf{x}_t))_{t \in \mathbb{N}}$  follows from Lemma 1.4(i). Telescoping the lefthand sum of (33) and invoking Lemma 1.4(ii) with Assumption 1.1, we find  $f(\mathbf{x}_t) - f(\mathbf{x}_{t+K}) \geq \rho(G_I(\mathbf{x}_t) + A_t, \sum_{k=1}^K M_{t+k} D^2)$ . Since  $f$  is convex, by optimality of the LMO and (6), we have

$$G_I(\mathbf{x}_t) \geq \langle \nabla f(\mathbf{x}_t) | \mathbf{x}_t - \mathbf{x}^* \rangle \geq f(\mathbf{x}_t) - f(\mathbf{x}^*), \tag{34}$$

so (32) follows from monotonicity of  $\rho$  (Fact 1.2).  $\square$

**Theorem 2.2** Let  $\times_{i \in I} C_i \subset \mathcal{H}$  be a product of  $m$  nonempty compact convex sets, let  $D$  be the diameter of  $\times_{i \in I} C_i$ , let  $f: \mathcal{H} \rightarrow \mathbb{R}$  be convex and  $L_f$ -smooth, let  $\tau > 1 \geq \eta > 0$  and  $M_0 > 0$  be approximation parameters, let  $\mathbf{x}^*$  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  $n_0 = 0$ <sup>3</sup>; otherwise,  $n_0 := \max\{\lceil \log(\tau L_f / (\eta M_0)) / (K \log \eta) \rceil, 0\}$ . In the setting of Algorithm 2, suppose that  $K$  satisfies Assumption 1.1, and set

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

Then, in the first  $t$  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  $n \in \mathbb{N}$ ,  $f(\mathbf{x}_{nK}) -$

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

$f(\mathbf{x}^*)$  is bounded above by

$$\begin{cases} \min_{0 \leq p \leq n-1} \left\{ \frac{K\eta^{pK}M_0D^2}{2} - A_{pK} \right\} & \text{if } 1 \leq n \leq n_0 + 1 \\ \frac{2K\tau L_f D^2}{n - n_0 + \sum_{p=n_0}^n \frac{2A_{pK}}{f(\mathbf{x}_{n_0}) - f(\mathbf{x}^*)} + \left( \frac{A_{pK}}{f(\mathbf{x}_{n_0}) - f(\mathbf{x}^*)} \right)^2} & \text{if } n > n_0 + 1. \end{cases} \quad (35)$$

*Proof.* We start by estimating the number of function and gradient computations of Algorithm 2. Except for  $t = 0$ , where  $f(\mathbf{x}_0)$  and  $\nabla f(\mathbf{x}_0)$  are computed, for all  $t \geq 1$ , in the preceding iteration  $f(\mathbf{x}_t)$  and  $\nabla f(\mathbf{x}_t)$  have already been computed. So, in the first  $t$  iterations, there have been  $t + 1$  function and gradient evaluations for the *initial* check of Line 11 in each iteration. Now, let  $k$  denote the total number of function and gradient evaluations in the first  $t$  iterations, 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. By Remark 2.1, unless  $k = 0$ , we have  $M_t = \eta^t \tau^{k-t-1} M_0 < \tau L_f$ , therefore at most  $k \leq t + 1 + \max\{0, \lceil \log_\tau(\eta^{-t} L_f / M_0) \rceil\}$  function and gradient evaluations are performed.

We turn now to the convergence rate. As in Remark 2.1, 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 2.1,  $M_{(n-1)K+1} = \eta^{(n-1)K+1} M_0 > \tau L_f$  and  $M_{(n-1)K+1} \geq M_t$  for all  $t > (n-1)K$ . By Lemma 2.1 and Fact 1.2,

$$\begin{aligned} f(\mathbf{x}_{(n-1)K}) - f(\mathbf{x}_{nK}) &\geq \rho \left( f(\mathbf{x}_{(n-1)K}) - f(\mathbf{x}^*) + A_{(n-1)K}, \sum_{k=1}^K M_{nK+k} D^2 \right) \\ &\geq \rho(f(\mathbf{x}_{(n-1)K}) - f(\mathbf{x}^*) + A_{(n-1)K}, K\eta^{(n-1)K+1} M_0 D^2) \\ &\geq f(\mathbf{x}_{(n-1)K}) - f(\mathbf{x}^*) + A_{(n-1)K} - \frac{K\eta^{(n-1)K+1} M_0 D^2}{2}. \end{aligned} \quad (36)$$

Rearranging (36) shows  $f(\mathbf{x}_{nK}) - f(\mathbf{x}^*) \leq K\eta^{(n-1)K+1} M_0 D^2 / 2 - A_{(n-1)K}$ . Therefore, since  $(f(\mathbf{x}_t))_{t \in \mathbb{N}}$  is monotonically decreasing (Lemma 2.1), the first case of (35) 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 2.1 yields

$$\begin{aligned} f(\mathbf{x}_{(n-1)K}) - f(\mathbf{x}_{nK}) &\geq \rho \left( f(\mathbf{x}_{(n-1)K}) - f(\mathbf{x}^*), \sum_{k=1}^K M_{(n-1)K+k} D^2 \right) \\ &\geq \rho(f(\mathbf{x}_{(n-1)K}) - f(\mathbf{x}^*), K\tau L_f D^2). \end{aligned} \quad (37)$$

and the second case of (36) follows from Lemma 1.3.  $\square$

To interpret the extra gaps  $(A_t)_{t \in \mathbb{N}}$  in Theorem 2.2, see Remark 1.2.

**Corollary 2.1** In the context of Theorem 2.2, let Algorithm 2 use a block selection strategy without coordinate reactivation, i.e.,  $I_{nK+i} \cap I_{nK+j} = \emptyset$  for all  $n$  and  $1 \leq i < j \leq K$ . Then, for any  $0 < \varepsilon \leq K\tau L_f D^2/2$ , the primal gap  $f(\mathbf{x}_{nK}) - f(\mathbf{x}^*) \leq \varepsilon$  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 values and gradients.

*Remark 2.3* 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 2.1.

## 2.2 Short-steps with convex objectives

In this section, we consider the step size rule (short) of Remark 1.1.

---

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

---

**Require:** Function  $f: \times_{i \in I} C_i \rightarrow \mathbb{R}$ , gradient  $\nabla f$ , point  $\mathbf{x}_0 \in \times_{i \in I} C_i$ , linear minimization oracles  $(\text{LMO}_i)_{i \in I}$

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

---

Short-Step BCFW (Algorithm 3) requires an upper bound on  $L_f$ . For this price, the algorithm becomes easier to parallelize in lines 5–8, foregoes any function evaluations, and requires only one gradient evaluation per iteration. Also, both the convergence rate and the prefactor of obtained in this section match the non-block version ( $K = 1$ ) [6, Theorem 2.2].

**Corollary 2.2** Let  $\times_{i \in I} C_i \subset \mathcal{H}$  be a product of  $m$  nonempty compact convex sets, let  $D$  be the diameter of  $\times_{i \in I} C_i$ , let  $f: \mathcal{H} \rightarrow \mathbb{R}$  be convex and  $L_f$ -smooth, let  $\mathbf{x}^*$  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

$A_t = \sum_{k=1}^{K-1} G_{I_{t+k-1} \cap (I_{t+k} \cup \dots \cup I_{t+K-1})}(\mathbf{x}_{t+k}) \geq 0$ . Then, for every  $n \in \mathbb{N}$ ,

$$f(\mathbf{x}_{nK}) - f(\mathbf{x}^*) \leq \begin{cases} \frac{KL_f D^2}{2} - A_0 & \text{if } n = 1 \\ \frac{2KL_f D^2}{n - 1 + \sum_{p=1}^n \frac{2A_{pK}}{f(\mathbf{x}_1) - f(\mathbf{x}^*)} + \left( \frac{A_{pK}}{f(\mathbf{x}_1) - f(\mathbf{x}^*)} \right)^2} & \text{if } n \geq 2. \end{cases} \quad (38)$$

Furthermore Algorithm 3 requires one gradient evaluation per iteration.

*Proof.* This follows from the fact that Algorithm 3 produces the same sequence of iterates as Algorithm 2: by initializing Algorithm 2 with  $M_0 = L_f$  and  $\eta = 1$ , as by Fact 2.1, the condition in Line 11 of Algorithm 2 is always true. Hence, this case of Algorithm 2 coincides with Algorithm 3 and we achieve convergence from Theorem 2.2 for all  $\tau > 1$ ; taking the limit as  $\tau \searrow 1$  yields (38). Clearly, Algorithm 3 requires one gradient evaluation per iteration.  $\square$

### 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  $\times_{i \in I} C_i \subset \mathcal{H}$  be a product of  $m$  nonempty compact convex sets and let  $f: \mathcal{H} \rightarrow \mathbb{R}$  be  $L_f$ -smooth on  $\times_{i \in I} C_i$ . In the setting of Algorithm 3,  $(f(\mathbf{x}_t))_{t \in \mathbb{N}}$  is monotonically decreasing, and*

$$(\forall t \in \mathbb{N}) \quad f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}) \geq \frac{\langle \nabla f(\mathbf{x}_t) | \mathbf{x}_t - \mathbf{x}_{t+1} \rangle}{2} \geq \frac{L_f \|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2}{2}. \quad (39)$$

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

$$\langle \nabla f(\mathbf{x}_t) | \mathbf{x}_t^i - \mathbf{x}_{t+1}^i \rangle = \gamma_t^i G_i(\mathbf{x}_t) \geq (\gamma_t^i)^2 L_f \|\mathbf{v}_t^i - \mathbf{x}_t^i\|^2 = L_f \|\mathbf{x}_t^i - \mathbf{x}_{t+1}^i\|^2. \quad (40)$$

Summing (40) for all  $i \in I_t$  and using  $\mathbf{x}_t^i = \mathbf{x}_{t+1}^i$  for  $i \notin I_t$  (Line 10), we obtain

$$\langle \nabla f(\mathbf{x}_t) | \mathbf{x}_t - \mathbf{x}_{t+1} \rangle \geq L_f \|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2. \quad (41)$$

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

$$\begin{aligned} f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}) &\geq \langle \nabla f(\mathbf{x}_t) | \mathbf{x}_t - \mathbf{x}_{t+1} \rangle - \frac{L_f}{2} \|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2 \\ &\geq \frac{\langle \nabla f(\mathbf{x}_t) | \mathbf{x}_t - \mathbf{x}_{t+1} \rangle}{2} \\ &\geq \frac{L_f}{2} \|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2. \end{aligned} \quad (42)$$

□

**Lemma 3.2 (Progress bound via smoothness (5))** Let  $\times_{i \in I} C_i \subset \mathcal{H}$  be a product of  $m$  nonempty compact convex sets, let  $D$  be the diameter of  $\times_{i \in I} C_i$ , let  $f: \times_{i \in I} C_i \rightarrow \mathbb{R}$  be a function with  $L_f$ -Lipschitz continuous gradient  $\nabla f$  on  $\times_{i \in I} C_i$ , 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

$$A_t = \sum_{k=0}^{K-1} G_{(I_{t+k} \cup \dots \cup I_{t+K-1}) \cap I_{t+k-1}}(\mathbf{x}_{t+k}) \geq 0. \text{ Then}$$

$$f(\mathbf{x}_t) - f(\mathbf{x}_{t+K}) \geq \frac{\rho(G_I(\mathbf{x}_t) + A_t, K L_f D^2)}{2}. \quad (43)$$

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

$$f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}) \geq \langle \mathbf{g}_t \mid \mathbf{x}_t - \mathbf{x}_{t+1} \rangle - \frac{L_f \|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2}{2}. \quad (44)$$

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

$$f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}) \geq \frac{L_f \|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2}{2} \geq \frac{\|\mathbf{g}_t - \mathbf{g}_{t+1}\|^2}{2 L_f} \geq \frac{\|\mathbf{g}_t^{I \setminus I_t} - \mathbf{g}_{t+1}^{I \setminus I_t}\|^2}{2 L_f}. \quad (45)$$

The sum of (44) and (45) is

$$2(f(\mathbf{x}_t) - f(\mathbf{x}_{t+1})) \geq \langle \mathbf{g}_t \mid \mathbf{x}_t - \mathbf{x}_{t+1} \rangle - \frac{L_f \|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2}{2} + \frac{\|\mathbf{g}_t^{I \setminus I_t} - \mathbf{g}_{t+1}^{I \setminus I_t}\|^2}{2 L_f}. \quad (46)$$

Summing (46) from  $t$  to  $t + K - 1$ , invoking Lemma 1.4(ii), then dividing by 2 yields (43). □

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, 23], this is closely related to stationarity.

**Theorem 3.1 (Nonconvex convergence)** Let  $\times_{i \in I} C_i \subset \mathcal{H}$  be a product of  $m$  nonempty compact convex sets with diameter  $D$ . Let  $f: \mathcal{H} \rightarrow \mathbb{R}$  be such that  $\nabla f$  is  $L_f$ -Lipschitz continuous on  $\times_{i \in I} C_i$ . Let  $G_I$  be given by (8). In the setting of Algorithm 3, suppose that  $K$  satisfies Assumption 1.1, set  $H_0 = f(\mathbf{x}_0) - \inf_{\mathbf{x} \in \times_{i \in I} C_i} f(\mathbf{x})$ , and for every  $n \in \mathbb{N}$  set

$$A_n = \sum_{k=1}^{K-1} G_{I_{n+k-1} \cap (I_{n+k} \cup \dots \cup I_{n+K-1})}(\mathbf{x}_{n+k}) \geq 0. \quad (47)$$

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

$$\begin{aligned} \min_{0 \leq p \leq n-1} G_I(\mathbf{x}_{pK}) &\leq \frac{1}{n} \sum_{p=0}^{n-1} G_I(\mathbf{x}_{pK}) \\ &\leq \begin{cases} \frac{2H_0 - \sum_{p=0}^{n-1} A_{pK}}{n} + \frac{KL_f D^2}{2} & \text{if } n \leq \frac{2H_0}{KL_f D^2} \\ 2D \sqrt{\frac{H_0 K L_f}{n}} - \frac{\sum_{p=0}^{n-1} A_{pK}}{n} & \text{otherwise.} \end{cases} \end{aligned} \quad (48)$$

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

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

$$\begin{aligned} 2H_0 &\geq 2(f(x_0) - f(x_{nK})) \geq \sum_{p=0}^{n-1} \rho(G_I(\mathbf{x}_{pK}) + A_{pK}, KL_f D^2) \\ &\geq \rho \left( \sum_{p=0}^{n-1} G_I(\mathbf{x}_{pK}) + A_{pK}, nKL_f D^2 \right). \end{aligned}$$

Observe that, for  $x, y \geq 0$  and  $b > 0$ , we have for  $y \geq \frac{b}{2}$  by strict monotonicity of  $\rho$  that  $y \geq \rho(x, b)$  if and only if  $x \leq y + \frac{b}{2}$ . For  $y \leq \frac{b}{2}$  we have  $y \geq \rho(x, b)$  if and only if  $x \leq \sqrt{2by}$ . Therefore,

$$\sum_{p=0}^{n-1} G_I(\mathbf{x}_{pK}) + A_{pK} \leq \begin{cases} 2H_0 + \frac{nKL_f D^2}{2} & \text{if } 2H_0 \geq nKL_f D^2 \\ 2D \sqrt{H_0 nKL_f} & \text{otherwise.} \end{cases} \quad (49)$$

Dividing (49) by  $n$  and rearranging yields (48).  $\square$

*Remark 3.1* It is possible to still achieve  $\mathcal{O}(K/t)$  convergence, by replacing Line 11 in Algorithm 2 with an analogue of (5) (as in [23]); the proof proceeds by a similar argument to that of Theorem 2.2, replacing Lemma 2.1 with Lemmas 3.1 and 3.2, and yielding the same rate with a worse constant. This may be useful in situations where checking (5) is preferable to (29) (e.g., Section 4.3), or if  $f$  is convex and  $\nabla f$  is Lipschitz-continuous on  $\times_{i \in I} C_i$ , yet  $f$  is not extendable to a smooth function on  $\mathcal{H}$  (see Fact 2.1).

## 4 Numerical Experiments

In this section we examine different block selection strategies covered by Assumption 1.1; Sections 4.1 and 4.2 contain simple experiments where the LMO for the last constraint is far more expensive than the other LMOs, while Section 4.3 uses BCFW to solve a structural SVM training problem [19]. In line with Theorems 2.2 and 3.1, our optimality criterion is the primal gap for the

convex problems in Sections 4.1 and 4.3, and minimal F-W gap for the non-convex problem in Section 4.2. Computations were performed on a single Slurm node with 3 GB RAM and no concurrencies, allocated on an Intel Xeon Gold 6338 machine with 2.0 GHz CPU speed, running Linux. For each experiment, we averaged the results over 20 trials.

In Sections 4.1 and 4.2, we ran 10,000 iterations of BCFW using the package `FrankWolfe.jl` (v0.3.3) [4] in Julia 1.11.5. To obtain feasible initial iterates  $x_0$ , we evaluate the LMO for each component in directions with  $\mathcal{N}(0, 1)$ -distributed entries. Within both experiments, we vary 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, 22]) 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  $\{1, \dots, m\}$ .
- (iv) *Lazy essentially cyclic activation, E-Cyclic*:  $I_t = \{i(t)\}$  satisfying (3), (which requires  $K \geq m$ ); each sequence of  $K$  components is created by randomly shuffling tuples of  $\{1, \dots, m - 1\}$ , with the final activation being the expensive  $m$ th component.

Among these selection methods, only E-cyclic allows for the user to leverage a-priori knowledge of LMO runtimes.

In addition to plotting optimality criterion against iterations and time, we also include plots against the number of calls to 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 both experiments, since it correlated with time used for all algorithms, and it was the dominant cost of even a full F-W iteration. 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.

In Section 4.3, we compare the original algorithm of [19] to Algorithm 2 in MATLAB R2023b using the same initial iterates. For block selection strategies, we consider a generalized P-Cyclic activation with blocks of size  $n$ :

$$I_t = \{\sigma_{\lfloor j/m \rfloor}(j \pmod{m}) \mid j \in \{(t-1)n+1, \dots, tn\}\},$$

where each  $\sigma$  is a random permutation of  $\{1, \dots, m\}$ . These activations satisfy Assumption 1.1 for  $K = 2\lceil n/m \rceil$ . The algorithm in [19] uses a P-Cyclic activation with singleton blocks as well as a uniform activation, i.e.,  $I_t = \{\sigma_t(t \pmod{m})\}$ , which does not satisfy Assumption 1.1. To ensure a fair comparison of the orginal algorithm and Algorithm 2, we run both algorithms for 150 epochs, i.e., 150 LMO calls for each component. This yields different iteration counts depending on the chosen block sizes. We plot against number of iterations and epochs as well as wall-clock time.

The code used to produce the result in Section 4.1 and 4.2 can be found at <https://github.com/zevwoodstock/BlockFW>. The code used to produce the results in Section 4.3 can be found at <https://github.com/JannisHal/BCFWstruct>. The new block updating schemes and Algorithm 2 are available in <https://github.com/ZIB-IOL/FrankWolfe.jl> as LazyUpdate and adaptive\_block\_coordinate\_frank\_wolfe.

#### 4.1 Experiment 1: Intersection problem

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

$$\underset{\mathbf{x} \in C_1 \times C_2}{\text{minimize}} \frac{1}{2} \|\mathbf{x}^1 - \mathbf{x}^2\|^2 \quad (50)$$

is zero. Problem (50) is convex, with smoothness constant  $L_f = 2$ . In this problem, the spectrahedral linear minimization oracle, LMO<sub>2</sub>, is far more expensive than LMO<sub>1</sub>. 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]:

$$(\forall t \in \mathbb{N}) \quad I_t = \begin{cases} \{1, 2\} & \text{if } t \equiv 0 \pmod{q}; \\ \{1\} & \text{otherwise.} \end{cases} \quad (51)$$

We run 20 instances of this problem on random initializations, and the averaged results are shown in Figure 1 for short-step (Algorithm 3). Even though using  $q$ -lazy activation is computationally cheaper on average, the per-iteration progress is still competitive with that of full, cyclic, P-cyclic, and E-cyclic activation. However, since they compute LMO<sub>2</sub> at a much lower rate, these activation strategies also have a faster per-iteration computation time. The activation scheme (51) performs similarly to an E-cyclic rule with a similar refresh rate; this may be due to the small number of components in this problem.

In Figure 2 we compare BCFW with short-step to the adaptive variant (Algorithm 2). The results indicate that, at least on problems like (50), i.e., with a small number of components and where  $L_f$  is known, short-step may be preferable. However, some updating strategies (e.g., full and cyclic) may exhibit performance improvements, while others exhibit relatively little improvement change. See Section 4.3 for a better use-case for Algorithm 2.

#### 4.2 Experiment 2: Difference of convex quadratics

The goal is to minimize a difference of convex quadratic functions of two collated matrices in  $\mathbb{R}^{s \times s}$ , where the submatrices are constrained to an  $\ell_\infty$

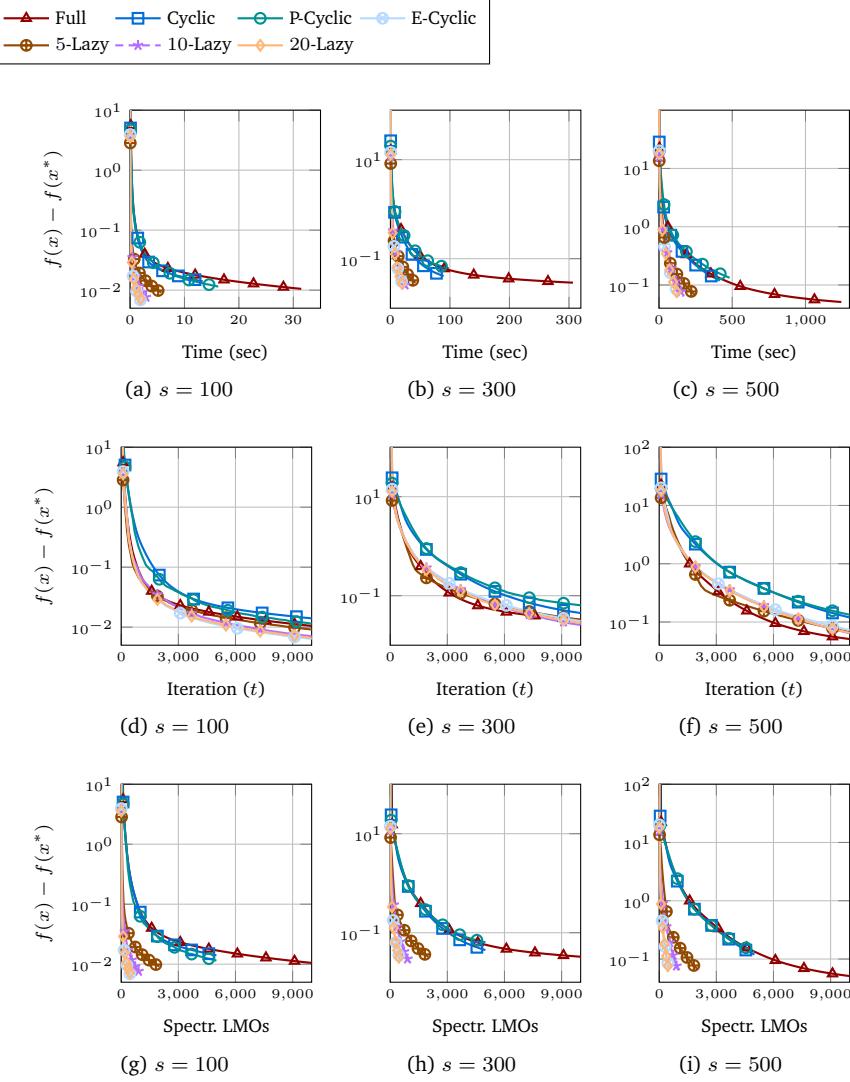


Fig. 1: Results of intersection for a cube and spectraplex (Section 4.1) displaying the primal gap  $f(\mathbf{x}_t) - f(\mathbf{x}^*)$  versus time, iteration, and spectrahedral LMO count for problems with  $s^2$  variables and block-activation strategies (i)–(iv) (E-cyclic uses  $K = 20$ ) and (51).

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  $s$  separate constraints. Hence, we set  $C_1 = \dots = C_s := \{x \in \mathbb{R}^s \mid \|x\|_\infty \leq 1\}$ , and  $C_{s+1} = \{x \in \mathbb{R}^{s \times s} \mid \|x\|_{\text{nuc}} \leq 1\}$ . For  $\mathbf{x} \in \times_{i \in I} C_i$  we use  $[\mathbf{x}]$  to denote the collated  $2s \times s$  matrix of its components. For each problem instance, the kernel  $A, B$  of each quadratic is generated by project-

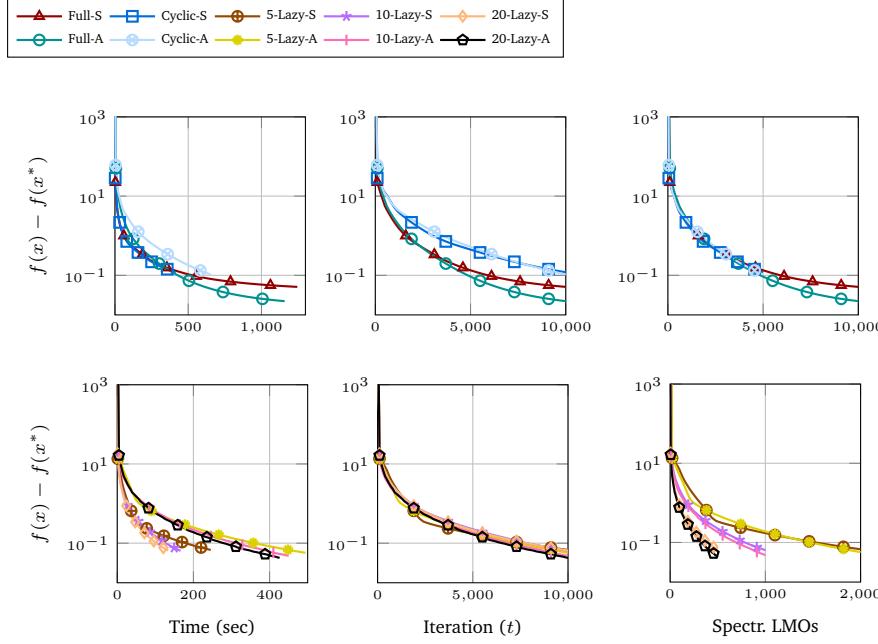


Fig. 2: Results of intersection for a cube and spectraplex for  $s = 500$  (Section 4.1) comparing adaptive step sizes (Algorithm 2) labelled with suffix “-A” to short-step sizes (Algorithm 3) labelled with suffix “-S”. The first row compares the performance of the the Full and Cyclic activation strategies, the second row compares the lazy activation strategies.

ing 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

$$\underset{\mathbf{x} \in C_1 \times \dots \times C_{s+1}}{\text{minimize}} \quad \frac{1}{2} \left( \langle [\mathbf{x}] \mid [\mathbf{x}] A \rangle - \langle [\mathbf{x}] \mid [\mathbf{x}] B \rangle \right). \quad (52)$$

Note the objective function of (52) is smooth and nonseparable. In the experiments we used the Frobenius norm of  $A - B$  as smoothness constant  $L_f$ . For each instance of (52), 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  $\text{LMO}_{s+1}$ . 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  $I \setminus \{s + 1\}$  of size  $p$ :

$$(\forall t \in \mathbb{N}) \quad I_t = \begin{cases} I & \text{if } t \equiv 0 \pmod{q} \\ \{i_1, \dots, i_p\} \subset I \setminus \{s + 1\} & \text{otherwise.} \end{cases} \quad (53)$$

Averaged results from 20 instances of (52) are shown in Figure 3. Since the problem is nonconvex, we plot the minimal F-W gap observed (see Theorem 3.1). 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  $(s+1)$ st component is activated too infrequently, results may worsen; this is reflected in the cyclic,  $P$ -cyclic,  $E$ -cyclic ( $K = 2s$ ), and  $(p, q) = (2, 20)$  results. The on-par performance with full-activation Frank-Wolfe suggests that, particularly if gradients are expensive relative to LMOs (as is the case here, but not in Experiment 1), selecting a larger number of (cheap) LMOs to update may yield a beneficial balance of progress-per-iteration and wallclock time.

#### 4.3 Experiment 3: Structural SVM training

In the final experiment, we consider the structural SVM training problem for sequence labeling from [19] and analyze the OCR dataset [28], which contains 6251 instances with 4028 features. This task is solved via a dual reformulation, detailed in [19], resulting in a problem of the form (1). The block-coordinate Frank-Wolfe (BCFW) method in [19] uses a permuted cyclic update strategy satisfying Assumption 1.1, but relies on componentwise optimal step sizes on the dual problem, which are not guaranteed to converge under parallel updates (see Example 1.1).

Computing the Lipschitz smoothness constant for this dual problem is infeasible, despite its quadratic objective, because the immense size of the Hessian, which corresponds to the number of potential outputs. For the OCR dataset, with maximum sequence length 14 and 36 possible labels per position, there are  $36^{14} \approx 6 \cdot 10^{21}$  potential outputs. Consequently, BCFW with short-step updates must approximate the smoothness constant, making this a natural use case for Algorithm 2.

As computing the dual gradient involves the same large matrix and is thus intractable, we rely on the smoothness inequality (5) rather than the interpolability condition in (29) (see Remark 3.1). While this still requires maintaining dual variables, they are highly sparse due to the structure of the LMO over the unit simplex, so the additional storage is manageable. In contrast, the primal-dual method in [19] with exact line search only stores primal variables, avoiding this overhead.

Figure 4 compares Algorithm 2 with P-Cyclic block updates of size 1, 5, and 10 to the original method in [19], which uses both P-Cyclic and uniformly random singleton updates. All methods achieve similar primal progress per epoch and time. The adaptive algorithm is slightly slower due to dual-variable

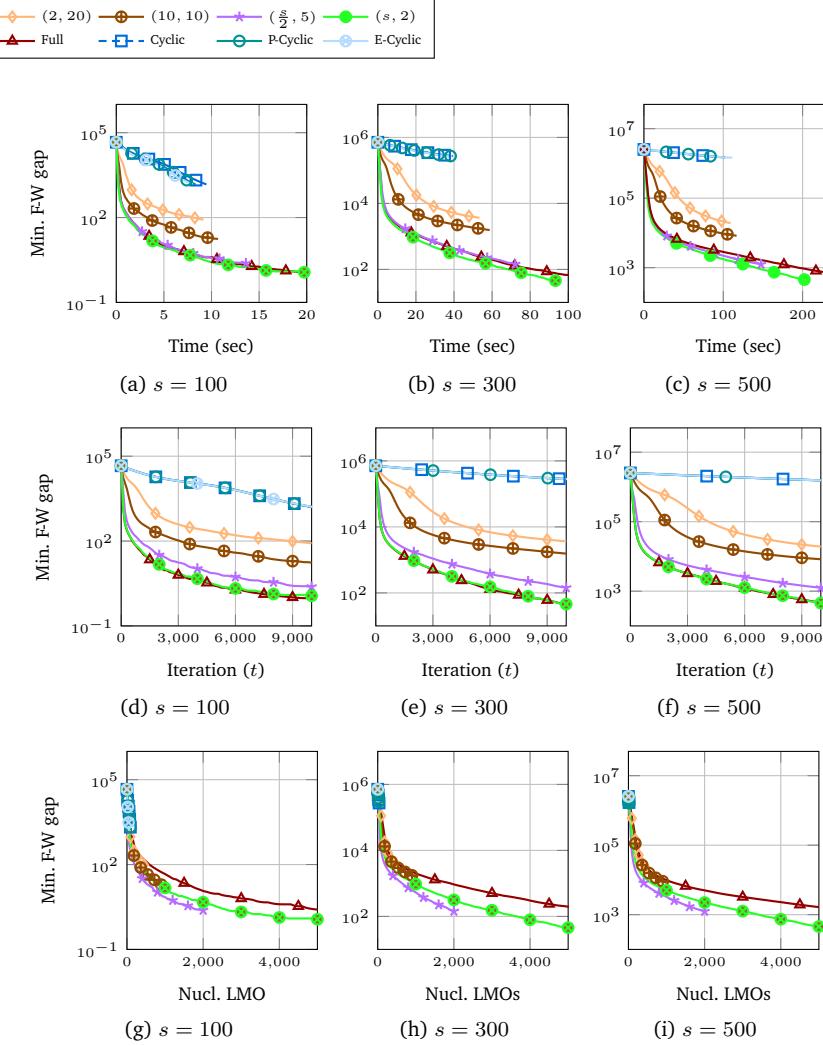


Fig. 3: Results for difference of quadratic functions (Section 4.2) displaying the minimal F-W gap versus time, iteration, and nuclear norm LMO count for problems with  $s^2$  variables and block-activation strategies (i)–(iii) and (53);  $K = 2s$  for E-cyclic activation; for  $(p, q)$  activation,  $p$  is the number of “cheaper” coordinates activated per iteration, and  $q$  is the frequency of full activation.

maintenance and extra computations for smoothness estimation. Updates with 5 or 10 blocks outperform singleton updates in iteration count.

In conclusion, Algorithm 2 is a competitive alternative to the original BCFW method for structural SVM training. It does not require knowledge of problem parameters like the smoothness constant, supports parallel updates, and can be applied to objectives where exact line search is infeasible or too costly.

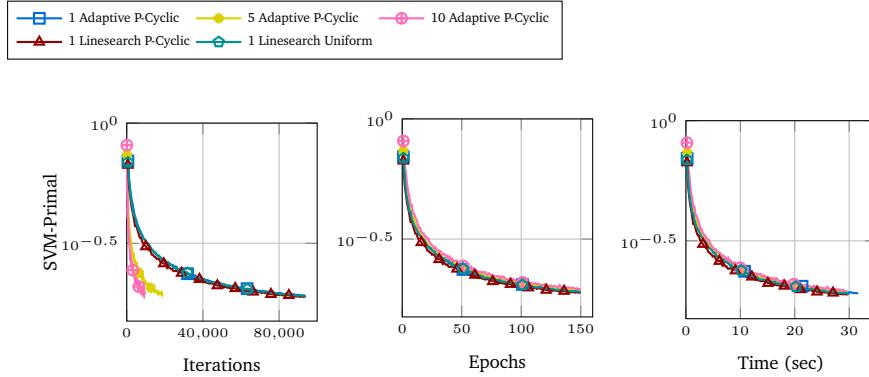


Fig. 4: Results of the structural SVM training experiment (Section 4.3) displaying the primal progress versus iterations, epochs and time. comparing Algorithm 2 using P-Cyclic updates of size 1, 5, and 10, and the original method in [19] using an exact line search with P-Cyclic and uniformly random singleton updates.

## 5 Future work

Here we conclude by outlining potential directions for future work. This article focuses on the short-step family of step sizes, and there is not a clear avenue for generalizing to other step size families, e.g., open-loop or line search [19, Algorithm 3]. Such a task may be more tractable on separable objective functions (note neither objective in Examples 1.1,1.2 are separable). The present analysis also does not generalize to the case of nonsmooth or partially nonsmooth functions, which can be common in applications.

**Acknowledgements** 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 (BMBF) (fund numbers 05M14ZAM, 05M20ZBM). The work of ZW is also supported by NSF DMS-2532423. Part of this work was conducted while SP was visiting Tokyo University via a JSPS International Research Fellowship.

We thank our referees, as well as D. Hendrych, G. Iommazzo, D. Kuzinowicz, M. Turner, and E. Wirth for valuable feedback on early drafts of this article.

**Data Availability Statement** The code used to produce these results can be found at <https://github.com/zewwoodstock/BlockFW> and <https://github.com/JannisHal/BCFWstruct>. New LazyUpdate and adaptive\_block\_coordinate\_frank\_wolfe methods are also in <https://github.com/ZIB-IOL/FrankWolfe.jl>.

## 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.W., Hassani, H., Karbasi, A., Mokhtari, A., Pokutta, S.: Conditional Gradient Methods. Society for Industrial and Applied Mathematics, Philadelphia, PA (2025). DOI 10.1137/1.9781611978568. URL <https://pubs.siam.org/doi/abs/10.1137/1.9781611978568>
7. Braun, G., Pokutta, S.: The matching polytope does not admit fully-polynomial size relaxation schemes. In: Proc. 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
11. Combettes, C.W., Pokutta, S.: Boosting Frank-Wolfe by chasing gradients. In: Proc. of the 37th International Conference on Machine Learning (ICML), pp. 2111–2121. PMLR (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. Demyanov, V.F., Rubinov, A.M.: Approximate Methods in Optimization Problems. American Elsevier Publishing Company, New York (1970)
15. Diakonikolas, J., Carderera, A., Pokutta, S.: Locally accelerated conditional gradients. In: Proc. of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), vol. 108, pp. 1737–1747. PMLR (2020)
16. 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)
17. Hazan, E., Luo, H.: Variance-reduced and projection-free stochastic optimization. In: Proc. of the 33rd International Conference on Machine Learning (ICML), vol. 48, pp. 1263–1271. PMLR (2016)
18. 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
19. Lacoste-Julien, S., Jaggi, M., Schmidt, M., Pletscher, P.: Block-coordinate Frank-Wolfe optimization for structural SVMs. In: S. Dasgupta, D. McAllester (eds.) Proc. of the 30th International Conference on Machine Learning, vol. 28, pp. 53–61. PMLR, Atlanta, GA, USA (2013)
20. 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.) Proc. of The 33rd International Conference on Machine Learning, vol. 48, pp. 593–602. PMLR, New York, NY, USA (2016)
21. Ottavy, N.: Strong convergence of projection-like methods in Hilbert spaces. *J. Optim. Theory Appl.* **56**, 433–461 (1988)

22. Patriksson, M.: Decomposition methods for differentiable optimization problems over cartesian product sets. *Comput. Optim. Appl.* **9**, 5–42 (1998)
23. 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)
24. 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
25. Rothvoss, T.: The matching polytope has exponential extension complexity. *J. ACM* **64**(6), 41:1–19 (2017). DOI 10.1145/3127497
26. Shamanskii, V.E.: A modification of Newton's method. *Ukrain. Mat. Zh.* **19**(1), 133–138 (1967)
27. 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)
28. Taskar, B., Guestrin, C., Koller, D.: Max-margin markov networks. In: S. Thrun, L. Saul, B. Schölkopf (eds.) *Advances in Neural Information Processing Systems*, vol. 16. MIT Press (2003). URL [https://proceedings.neurips.cc/paper\\_files/paper/2003/file/878d5691c824ee2aaf770f7d36c151d6-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2003/file/878d5691c824ee2aaf770f7d36c151d6-Paper.pdf)
29. 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
30. Tseng, P.: Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.* **109**, 475–494 (2001)
31. 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.) *Proc. of The 33rd International Conference on Machine Learning*, vol. 48, pp. 1548–1557. PMLR, New York, NY, USA (2016)
32. Woodstock, Z., Pokutta, S.: Splitting the conditional gradient algorithm. *SIAM J. Optim.* **35**(1), 347–368 (2025)