Projection free methods on product domains

Immanuel Bomze* Francesco Rinaldi† Damiano Zeffiro‡

February 9, 2023

Abstract

Projection-free block-coordinate methods avoid high computational cost per iteration and at the same time exploit the particular problem structure of product domains. Frank-Wolfe-like approaches rank among the most popular ones of this type. However, as observed in the literature, there was a gap between the classical Frank-Wolfe theory and the block-coordinate case. Moreover, most of previous research concentrated on convex objectives. This study now deals also with the non-convex case and reduces above-mentioned theory gap, in combining a new, fully developed convergence theory with novel active set identification results which ensure that inherent sparsity of solutions can be exploited in an efficient way. Preliminary numerical experiments seem to justify our approach and also show promising results for obtaining global solutions in the non-convex case.

Keywords: projection free optimization, first order optimization, block coordinate descent.

1 Introduction

We consider the problem

$$\min_{\mathbf{x}\in\mathcal{C}} f(\mathbf{x})\,, \tag{1}$$

with objective f having L-Lipschitz regular gradient, and feasible set $\mathcal{C}\subseteq\mathbb{R}^n$ closed and convex. Furthermore, we assume that $\mathcal{C}$ is block separable, that is

$$\mathcal{C}=\mathcal{C}_{(1)}\times\ldots\times\mathcal{C}_{(m)} \tag{2}$$

with $\mathcal{C}_{(i)}\subset\mathbb{R}^{n_i}$ closed and convex for $i\in[1\!:\!m]$, and of course $\sum_{i=1}^m n_i=n$.

Notice that problem (1) falls in the class of composite optimization problems

$$\min_{\mathbf{x}\in\mathcal{C}}[f(\mathbf{x})+g(\mathbf{x})] \tag{3}$$

with f smooth and $g(\mathbf{x})=\sum_{i=1}^m \chi_{\mathcal{C}_{(i)}}(\mathbf{x}^{(i)})$ convex and block separable (see, e.g., [32] for an overview of methods for this class of problems); here $\chi_D:\mathbb{R}^d\to[0,+\infty]$ denotes the indicator function of a convex set $D\subseteq\mathbb{R}^d$, and for a block vector $\mathbf{x}\in\mathbb{R}^n=\mathbb{R}^{n_1}\times\ldots\times\mathbb{R}^{n_m}$ we denote by $\mathbf{x}^{(i)}\in\mathbb{R}^{n_i}$ the component corresponding to the i-th block, so that $\mathbf{x}=(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(m)})$.

*ISOR/VCOR, Universität Wien (immanuel.bomze@univie.ac.at)

†Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Italy (rinaldi@math.unipd.it)

‡Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Italy (damiano.zeffiro@math.unipd.it)

Problems of this type arise in a wide number of real-world applications like, e.g., traffic assignment [27], structural SVMs [24], trace-norm based tensor completion [28], reduced rank nonparametric regression [19], semi-relaxed optimal transport [20], structured submodular minimization [22], group fused lasso [1], and dictionary learning [17].

Block-coordinate gradient descent (BCGD) strategies (see, e.g., [4]) represent a standard approach to solve problem (1) in the convex case. When dealing with non-convex objectives, those methods can anyway still be used as an efficient tool to perform local searches in probabilistic global optimization frameworks (see, e.g., [29] for further details). The way BCGD approaches work is very easy to understand: those methods build up, at each iteration, a suitable model of the original function for a block of variables and then perform a projection on the feasible set related to that block. Such a strategy might however be costly even when the projection is performed over some structured sets like, e.g., the flow polytope, the nuclear-norm ball, the Birkhoff polytope, the permutahedron (see, e.g., [18]). This is the reason why, in recent years, projection-free methods (see, e.g., [13, 21, 25]) have been massively used when dealing with those structured constraints.

These methods simply rely on a suitable oracle that minimizes, at each iteration, a linear approximation of the function over the original feasible set, returning a point in

$$\operatorname{argmin}_{\mathbf{x}\in\mathcal{C}}\langle \mathbf{g}, \mathbf{x}\rangle.$$

When $\mathcal{C}$ is defined as in (2), this decomposes in m independent problems thanks to the block separable structure of the feasible set. In turn, the resulting problems on the blocks can then be solved in parallel, a possibility that has widely been explored in the literature, especially in the context of traffic assignment (see, e.g., [27]). In a big data context, performing a full update of the variables might still represent a computational bottleneck that needs to be properly handled in practice. This is the reason why block-coordinate variants of the classic Frank-Wolfe (FW) method have been recently proposed (see, e.g., [24, 31, 35]). This method is proposed in [24] for structured support vector machine training, and randomly selects a block at each iteration to perform an FW update on the block. Several improvements on this algorithm, e.g., adaptive block sampling, use of pairwise and away-step directions, or oracle call caching, are described in [31], which obviously work in a sequential fashion.

However, in case one wants to take advantage of modern multicore architectures or of distributed clusters, parallel and distributed versions of the block-coordinate FW algorithm are also available [35]. It is important to highlight that all the papers mentioned above only consider convex programming problems and use random sampling variants as the main block selection strategy.

Furthermore, as noticed in [31], the standard convergence analysis for FW variants (e.g., pairwise and away step FW) cannot be easily extended to the block-coordinate case. This is mainly due to the difficulties in handling the bad/short steps (i.e., those steps that do not give a good progress and are taken to guarantee feasibility of the iterate) within a block-coordinate framework. In [31], the authors hence claim that novel proof techniques are required to carry out the analysis and close the gap between FW and BCFW in this context.

Here we focus on the non-convex case and define a new general block-coordinate algorithmic framework that gives flexibility in the use of both block selection strategies and FW-like directions. Such a flexibility is mainly obtained thanks to the way we perform approximate minimizations in the blocks. At each iteration, after selecting one block at least, we indeed use the Short Step Chain (SSC) procedure described in [34], which skips gradient computations in consecutive short steps until proper conditions are satisfied, to get the approximate minimization done in the selected blocks.

Concerning the block selection strategies, we explore three different options. The first one we consider is a parallel or Jacobi-like strategy (see, e.g., [5]), where the SSC procedure is performed for all blocks. This obviously reduces the computational burden with respect to the use of the SSC in the whole variable space (see, e.g., [34]) and eventually enables to use multicore architectures to perform those tasks in parallel. The second one is the random sampling (see, e.g., [24]), where the SSC procedure is performed at each iteration on a randomly selected subset of blocks. Finally we have a variant of the Gauss-Southwell rule (see, e.g., [30]), where we perform SSC in all blocks and then select a block which violates optimality conditions at most. Such a greedy rule may make more progress in the objective function, since it uses first order information to choose the right block, but is, in principle, more expensive than the other options we mentioned before (notice that the SSC is performed, at each iteration, for all blocks).

Furthermore, we consider the following projection-free strategies: Away-step Frank-Wolfe (AFW), Pairwise Frank-Wolfe (PFW), and Frank-Wolfe method with in face directions (FDFW), see, e.g., [34] and references therein for further details. The AFW and PFW strategies depend on a set of "elementary atoms" A such that $\mathcal{C} = \text{conv}(A)$. Given A, for a base point $\mathbf{x} \in \mathcal{C}$ we can define

$$S_{\mathbf{x}} = \{S \subset A : \mathbf{x} \text{ is a proper convex combination of all the elements in } S\},$$

the family of possible active sets for a given point $\mathbf{x}$. For $\mathbf{x} \in \mathcal{C}$ and $S \in S_{\mathbf{x}}$, $\mathbf{d}^{\text{PFW}}$ is a PFW direction with respect to the active set S and gradient $-\mathbf{g}$ if and only if

$$\mathbf{d}^{\text{PFW}} = \mathbf{s} - \mathbf{q} \text{ with } \mathbf{s} \in \text{argmax}_{\mathbf{s}\in\mathcal{C}} \langle \mathbf{s}, \mathbf{g}\rangle \text{ and } \mathbf{q} \in \text{argmin}_{\mathbf{q}\in S} \langle \mathbf{q}, \mathbf{g}\rangle\,. \tag{4}$$

Similarly, given $\mathbf{x} \in \mathcal{C}$ and $S \in S_{\mathbf{x}}$, $\mathbf{d}^{\text{AFW}}$ is an AFW direction with respect to the active set S and gradient $-\mathbf{g}$ if and only if

$$\mathbf{d}^{\text{AFW}} \in \text{argmax}\{\langle \mathbf{g}, \mathbf{d}\rangle : \mathbf{d} \in \{\mathbf{d}^{\text{FW}}, \mathbf{d}^{\text{AS}}\}\}\,, \tag{5}$$

where $\mathbf{d}^{\text{FW}}$ is a classic Frank-Wolfe direction

$$\mathbf{d}^{\text{FW}} = \mathbf{s} - \mathbf{x} \text{ with } \mathbf{s} \in \text{argmax}_{\mathbf{s}\in\mathcal{C}} \langle \mathbf{s}, \mathbf{g}\rangle\,, \tag{6}$$

and $\mathbf{d}^{\text{AS}}$ is the away direction

$$\mathbf{d}^{\text{AS}} = \mathbf{x} - \mathbf{q} \text{ with } \mathbf{q} \in \text{argmin}_{\mathbf{q}\in S} \langle \mathbf{q}, \mathbf{g}\rangle\,. \tag{7}$$

The FDFW only requires the current point $\mathbf{x}$ and gradient $-\mathbf{g}$ to select a descent direction (i.e., it does not need to keep track of the active set) and is defined as

$$\mathbf{d}^{F} = \mathbf{x} - \mathbf{x}_F \text{ with } \mathbf{x}_F \in \text{argmin}\{\langle \mathbf{g}, \mathbf{y}\rangle : \mathbf{y} \in \mathcal{F}(\mathbf{x})\}$$

for $\mathcal{F}(\mathbf{x})$ the minimal face of $\mathcal{C}$ containing $\mathbf{x}$. The selection criterion is then analogous to the one used by the AFW:

$$\mathbf{d}^{\text{FD}} \in \text{argmax}\{\langle \mathbf{g}, \mathbf{d}\rangle : \mathbf{d} \in \{\mathbf{d}^{F}, \mathbf{d}^{\text{FW}}\}\}\,. \tag{8}$$

From a theoretical point of view, this new algorithmic framework enables us to give:

- a local linear convergence rate for any choice of block selection strategy and FW-like direction. This result is obtained under a Kurdyka-Łojasiewicz (KL) property (see, e.g., [3], [6] and [7]) and a tailored angle condition (see, e.g., [34]). Thanks to the way we handle short steps in our framework we are thus able to extend the analysis given for FW variants to the block-coordinate case and then to close the relevant gap in the theory highlighted in [31].

- an active set identification result (see, e.g., [11, 12, 14]) for a specific structure of the Cartesian product defining the feasible set $\mathcal{C}$ and suitable choices of projection-free strategy (i.e., AFW direction is used). In particular, we prove that our framework identifies in finite time the support of a solution. Such a theoretical feature allows to reduce the dimension of the problem at hand and, consequently, the overall computational cost of the optimization procedure.

This is, to the best of our knowledge, the first time that both a (bad step free) linear convergence rate and an active set identification result are given for block-coordinate FW variants. In particular, we solve the open question from [31] discussed above.

We also report some preliminary numerical results on a specific class of structured problems with a block separable feasible set. Those results show that the proposed framework outperforms the classic block-coordinate FW and, thanks to its flexibility, it can be effectively embedded into a probabilistic global optimization framework thus significantly boosting its performances.

The paper is organized as follows. Section 2 describes the details of our new algorithmic framework. An in-depth analysis of its convergence properties is reported in Section 3. An active set identification result is reported in Section 4. Preliminary numerical results, focusing on the computational analysis of both the local identification and the convergence properties of our framework, are reported in Section 5. Finally, some concluding remarks are included in Section 6.

1.1 Notation

For a closed and convex set $C \subset \mathbb{R}^h$ we denote by $\pi(C, \mathbf{x})$ the projection of $\mathbf{x} \in \mathbb{R}^h$ onto C, and by $T_C(\mathbf{x})$ the tangent space to C at $\mathbf{x} \in C$. For $\mathbf{g} \in \mathbb{R}^h$ we also use $\pi_{\mathbf{x}}(\mathbf{g})$ as a shorthand for $\|\pi(T_C(\mathbf{x}), \mathbf{g})\|$. We denote by $\hat{\mathbf{y}}$ the vector $\frac{\mathbf{y}}{\|\mathbf{y}\|}$ for $\mathbf{y} \neq \mathbf{o}$, and $\hat{\mathbf{y}} = \mathbf{o}$ otherwise. We finally denote by $\bar{B}_r(\mathbf{x})$ and $B_r(\mathbf{x})$ the closed and open balls of radius r centered at $\mathbf{x}$.

2 A new block-coordinate projection-free method

The block-coordinate framework we consider here applies the Short Step Chain (SSC) procedure from [34], described below as Algorithm 2, to some of the blocks at every iteration. A detailed scheme is specified as Algorithm 1; recall notation $\mathbf{x} = (\mathbf{x}^{(1)}, ..., \mathbf{x}^{(m)})$ with $\mathbf{x}^{(i)} \in \mathcal{C}_{(i)}$, all $i \in [1\!:\!m]$.

Algorithm 1 Block coordinate method with Short Step Chain (SSC) procedure

```
1: $\mathbf{x}_0 \in \mathcal{C}$, $k = 0$.
2: If $\mathbf{x}_k$ is stationary, then STOP
3: Choose $\mathcal{M}_k \subset [1\!:\!m]$.
4: For all $i \notin \mathcal{M}_k$ set $\mathbf{x}_{k+1}^{(i)} = \mathbf{x}_k^{(i)}$
5: For all $i \in \mathcal{M}_k$ set $\mathbf{x}_{k+1}^{(i)} = \text{SSC}(\mathbf{x}_k^{(i)}, -\nabla f(\mathbf{x}_k)^{(i)}))$
6: $k = k + 1$. Go to step 2.
```

In Algorithm 1, we perform two main operations at each iteration. First, in Step 3, we pick a suitable subset of blocks $\mathcal{M}_k$ according to a given block selection strategy. We then update (Steps 4 and 5) the variables related to the selected blocks by means of the SSC procedure, while keeping all the variables in the other blocks unchanged.

We now briefly recall the SSC procedure from [34], designed to recycle the gradient in consecutive bad steps until suitable stopping conditions are met, in Algorithm 2.

By $\mathcal{A}$ we indicate a projection-free strategy to generate first-order feasible descent directions for smooth functions on the block where the SSC is applied (e.g., FW, PFW, AFW directions). Since the gradient, $-\mathbf{g}$, is constant during the SSC procedure, it is easy to see that the procedure represents an application of $\mathcal{A}$ to minimize the linearized objective $f_{\mathbf{g}}(\mathbf{z}) = \langle -\mathbf{g}, \mathbf{z} - \bar{\mathbf{x}} \rangle + f(\bar{\mathbf{x}})$, with suitable stepsizes and stopping condition. More specifically, after a stationarity check (see Steps 2–4), the stepsize α_j is the minimum of an auxiliary stepsize $\beta_j > 0$ and the maximal stepsize $\alpha_{\max}^{(j)}$

Algorithm 2 Short Step Chain procedure – SSC($\bar{\mathbf{x}}, \mathbf{g}$)

```
**Initialization.** $\mathbf{y}_0 = \bar{\mathbf{x}}$, $j = 0$
Select $\mathbf{d}_j \in \mathcal{A}(\mathbf{y}_j, \mathbf{g})$, $\alpha_{\max}^{(j)} \in \alpha_{\max}(\mathbf{y}_j, \mathbf{d}_j)$
**if** $\mathbf{d}_j = 0$ **then return** $\mathbf{y}_j$
**end if**
compute an auxiliary step size $\beta_j$
let $\alpha_j = \min(\alpha_{\max}^{(j)}, \beta_j)$
$\mathbf{y}_{j+1} = \mathbf{y}_j + \alpha_j \mathbf{d}_j$
**if** $\alpha_j = \beta_j$ **then return** $\mathbf{y}_{j+1}$
**end if**
$j = j + 1$, go to Step 2
```

(which we always assume to be strictly positive). The point $\mathbf{y}_{j+1}$ generated at Step 7 is always feasible since $\alpha_j \leq \alpha_{\max}^{(j)}$. Notice that if the method $\mathcal{A}$ used in the SSC performs a FW step (see equation (6) for the definition of FW step), then the SSC terminates, with $\alpha_j = \beta_j$ or with $\mathbf{y}_{j+1}$ a global minimizer of $f_{\mathbf{g}}$.

The auxiliary step size β_j (see Step 5 of the SSC procedure) is thus defined as the maximal feasible stepsize (at $\mathbf{y}_j$) for the trust region

$$\Omega_j = B_{\|\mathbf{g}\|/2L}(\bar{\mathbf{x}} + \frac{\mathbf{g}}{2L}) \cap B_{\langle \mathbf{g}, \hat{\mathbf{d}}_j \rangle / L}(\bar{\mathbf{x}}) \,. \tag{9}$$

This guarantees the sufficient decrease condition

$$f(\mathbf{y}_j) \leq f(\mathbf{x}_k) - \frac{L}{2}\|\mathbf{x}_k - \mathbf{y}_j\|^2 \tag{10}$$

and hence a monotone decrease of f in the SSC. For further details see [34].

2.1 Block selection strategies

As briefly mentioned in the introduction, we will consider three different block selection strategies in our analysis. The first one is a parallel or Jacobi-like strategy (see, e.g., [5]). In this case, we select all the blocks at each iteration. As we already observed, this is computationally cheaper than handling the whole variable space at once. Furthermore, multicore architectures might eventually be considered to perform those tasks in parallel. A definition of the strategy is given below:

Definition 1 (Parallel selection)**.** *Set* $\mathcal{M}_k = [1\!:\!m]$.

The second strategy is a variant of the GS rule (see, e.g., [30]), where we first perform SSC in all blocks and then select a block that violates optimality conditions at most. The formal definition is reported below.

Definition 2 (Gauss-Southwell (GS) selection)**.** *Set* $\mathcal{M}_k = \{i(k)\}$*, with*

$$i(k) \in \operatorname{argmax}_{i \in [1:m]} \langle \mathbf{g}^{(i)}, \mathrm{SSC}(\mathbf{x}_k^{(i)}, -\nabla f(\mathbf{x}_k)^{(i)}) - \mathbf{x}_k^{(i)} \rangle.$$

Finally, we have random sampling (see, e.g., [24]). Here we randomly generate one index at each iteration with uniform probability distribution. The definition we have in this case is the following:

Definition 3 (Random sampling)**.** *Set* $\mathcal{M}_k = \{i(k)\}$*, with* $i(k)$ *index chosen uniformly at random in* $[1\!:\!m]$.

3 Convergence analysis

In this section, we analyze the convergence properties of our algorithmic framework. In particular, we show that under a suitably defined angle condition on the blocks and a local KL condition on the objective function, we get, for any block selection strategy used, a linear convergence rate.

Our convergence framework makes use of the angle condition introduced in [33, 34]. Such a condition ensures that the slope of the descent direction selected by the method is optimal up to a constant. We now recall this angle condition. For $\mathbf{x} \in \mathcal{C}$ and $\mathbf{g} \in \mathbb{R}^n$ we first define the directional slope lower bound as

$$\mathrm{DSB}_{\mathcal{A}}(\mathcal{C}, \mathbf{x}, \mathbf{g}) = \inf_{\mathbf{d} \in \mathcal{A}(\mathbf{x}, \mathbf{g})} \frac{\langle \mathbf{g}, \mathbf{d} \rangle}{\pi_{\mathbf{x}}(\mathbf{g}) \|\mathbf{d}\|}, \tag{11}$$

if $\mathbf{x}$ is not stationary for $-\mathbf{g}$, otherwise we set $\mathrm{DSB}_{\mathcal{A}}(\mathcal{C}, \mathbf{x}, \mathbf{g}) = 1$. We then define the slope lower bound as

$$\mathrm{SB}_{\mathcal{A}}(\mathcal{C}, P) = \inf_{\substack{\mathbf{g} \in \mathbb{R}^n \\ \mathbf{x} \in P}} \mathrm{DSB}_{\mathcal{A}}(\mathcal{C}, \mathbf{x}, \mathbf{g}) = \inf_{\substack{\mathbf{g} : \pi_{\mathbf{x}}(\mathbf{g}) \neq 0 \\ \mathbf{x} \in P}} \mathrm{DSB}_{\mathcal{A}}(\mathcal{C}, \mathbf{x}, \mathbf{g}) \,. \tag{12}$$

We use $\mathrm{SB}_{\mathcal{A}}(\mathcal{C})$ as a shorthand for $\mathrm{SB}_{\mathcal{A}}(\mathcal{C}, \mathcal{C})$, and say that the angle condition holds for the method $\mathcal{A}$ if

$$\mathrm{SB}_{\mathcal{A}}(\mathcal{C}) = \tau > 0 \,. \tag{13}$$

Remark 1. *AFW, PFW and FDFW all satisfy the angle condition, when $\mathcal{C}$ is a polytope. A detailed proof of this result is reported in [34], together with some other interesting examples of sets where the condition is satisfied with explicit bounds.*

We now report the local KL condition used to analyze the convergence of our algorithm.

Assumption 1. *Given a stationary point $\mathbf{x}_* \in \mathcal{C}$, there exists $\eta, \delta > 0$ such that for every $\mathbf{x} \in B_\delta(\mathbf{x}_*)$ with $f(\mathbf{x}_*) < f(\mathbf{x}) < f(\mathbf{x}_*) + \eta$ we have*

$$\pi_{\mathbf{x}}(-\nabla f(\mathbf{x})) \geq \sqrt{2\mu}[f(\mathbf{x}) - f(\mathbf{x}_*)]^{\frac{1}{2}} \,. \tag{14}$$

When dealing with convex programming problems, a Hölderian error bound with exponent 2 on the solution set implies condition (14), see [8, Corollary 6]. Therefore, our assumption holds when dealing with μ-strongly convex functions (see, e.g., [23]). It is however important to highlight that this error bound holds in a variety of both convex and non-convex settings (see [34] for a detailed discussion on this matter). An interesting example for our analysis is the setting where f is (non-convex) quadratic, i.e., $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{b}^\top \mathbf{x}$, and $\mathcal{C}$ is a polytope.

We now report our main convergence result. A detailed proof is included in the appendix.

Theorem 1. *Let Assumption 1 hold at $\mathbf{x}_*$. Let us consider the sequence $\{\mathbf{x}_k\}$ generated by Algorithm 1. Assume that:*

- *the angle condition* (13) *holds in every block for the same $\tau > 0$;*
- *the SSC procedure always terminates in a finite number of steps.*
- *$f(\mathbf{x}_*)$ is a minimum in the connected component of $\{\mathbf{x} \in \mathcal{C} : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ containing $\mathbf{x}_0$.*

Then, there exists $\tilde{\delta} > 0$ such that, if $\mathbf{x}_0 \in B_{\tilde{\delta}}(\mathbf{x}_*)$:

- *for the parallel block selection strategy, we have*
$$f(\mathbf{x}_k) - f(\mathbf{x}_*) \le (q_P)^k [f(\mathbf{x}_0) - f(\mathbf{x}_*)]\,, \tag{15}$$
and $\mathbf{x}_k \to \tilde{\mathbf{x}}_*$ *with*
$$\|\mathbf{x}_k - \tilde{\mathbf{x}}_*\| \le \frac{\sqrt{2 - 2q_P}}{\sqrt{L}(1 - \sqrt{q_P})}\,(q_P)^{\frac{k}{2}} [f(\mathbf{x}_0) - f(\tilde{\mathbf{x}}_*)]\,, \tag{16}$$
for
$$q_P = 1 - \frac{\mu\tau^2}{4L(1+\tau)^2}.$$
- *for the GS block selection strategy, we have*
$$f(\mathbf{x}_k) - f(\mathbf{x}_*) \le (q_{GS})^k [f(\mathbf{x}_0) - f(\mathbf{x}_*)]\,, \tag{17}$$
and $\mathbf{x}_k \to \tilde{\mathbf{x}}_*$ *with*
$$\|\mathbf{x}_k - \tilde{\mathbf{x}}_*\| \le \frac{\sqrt{2 - 2q_{GS}}}{\sqrt{L}(1 - \sqrt{q_{GS}})}\,(q_{GS})^{\frac{k}{2}} [f(\mathbf{x}_0) - f(\tilde{\mathbf{x}}_*)]\,, \tag{18}$$
for
$$q_{GS} = 1 - \frac{\mu\tau^2}{4mL(1+\tau)^2}\,,$$
- *for the random block selection strategy we have, under the additional condition that*
$$\min\{f(\mathbf{x}) : \|\mathbf{x} - \mathbf{x}_*\| = \delta\} > f(\mathbf{x}_*) \tag{19}$$
holds for some $\delta > 0$*, that*
$$\mathbb{E}[f(\mathbf{x}_k) - f(\mathbf{x}_*)] \le (q_R)^k [f(\mathbf{x}_0) - f(\mathbf{x}_*)]\,, \tag{20}$$
and $\mathbf{x}_k \to \tilde{\mathbf{x}}_*$ *almost surely with*
$$\mathbb{E}[\|\mathbf{x}_k - \tilde{\mathbf{x}}_*\|] \le \frac{\sqrt{2 - 2q_R}}{\sqrt{L}(1 - \sqrt{q_R})}\,(q_R)^{\frac{k}{2}} [f(\mathbf{x}_0) - f(\tilde{\mathbf{x}}_*)] \tag{21}$$
for $q_R = q_{GS}$*.*

It is easy to see that the SSC always terminates in a finite number of steps for the AFW, PFW, FDFW directions (see, e.g., [34]). Furthermore, we have a mild assumption on the stationary point $\mathbf{x}_*$, which is often satisfied in practice.

Remark 2. *If the feasible set $\mathcal{C}$ is a polytope and if we assume that the objective function f satisfies condition* (14) *on every point generated by the algorithm, with fixed $f(\mathbf{x}_*)$, then Algorithm 1 with AFW (PFW or FDFW) in the SSC converges at the rates given above. Condition* (14) *holds in case of μ-strongly convex functions, and hence we have that in those cases our algorithm globally converges with the rates given in Theorem 1.*

Remark 3. *Both the parallel and the GS strategy give the same rate with different constants. In particular, the constant ruling the GS case depends on the number of blocks used (the larger the number of blocks, the worse the rate) and is larger than the one we have for the parallel case.*

Remark 4. *The random block selection strategy has the same rate as the GS strategy, but it is given in expectation. In particular, the constant ruling the rate is the same as the GS one, hence depends on the number of blocks used. Note that a further technical assumption* (19) *on $\mathbf{x}_*$ is needed in this case.*

4 Active set identification

We now report an active set identification result for our framework. We only focus on Algorithm 1 with AFW in the SSC and assume that strict complementarity holds and that the sets in the Cartesian product have a specific structure:

$$\mathcal{C} = \Delta^{n_1} \times ... \times \Delta^{n_m}, \tag{22}$$

so that the set $\mathcal{C}_{(i)}$ is the $(n_i - 1)$-dimensional standard simplex

$$\Delta^{n_i} = \{\mathbf{x} \in \mathbb{R}^{n_i}_+ : \mathbf{x}^\top \mathbf{e}^{(i)} = 1\}, \quad i \in [1\!:\!m],$$

for $\mathbf{e} \in \mathbb{R}^n$ the vector with components all equal to 1. We now report our main identification result. A detailed proof is included in the appendix.

Theorem 2. *Under the above assumptions on $\mathcal{C}$, let $\mathcal{A}^{(i)}$ be the AFW for $i \in [1\!:\!m]$, and let strict complementarity conditions hold at $\mathbf{x}_* \in \mathcal{C}$.*

- *If $\{\mathbf{x}_k\}$ is generated by Algorithm 1 with parallel selection, then there exists a neighborhood U of $\mathbf{x}_*$ such that if $\mathbf{x}_k \in U$ then $\mathrm{supp}(\mathbf{x}_{k+1}) = \mathrm{supp}(\mathbf{x}_*)$.*
- *If $\{\mathbf{x}_k\}$ is generated by Algorithm 1 with randomized or GS selection, then there exists a neighborhood U of $\mathbf{x}_*$ such that if $\mathbf{x}_k \in U$ then $\mathrm{supp}(\mathbf{x}_{k+1}^{i(k)}) = \mathrm{supp}(\mathbf{x}_*^{i(k)})$.*

When the sequence generated by our algorithm converges to the point $\mathbf{x}_*$, it is then easy to see that the support of the iterate matches the final support of $\mathbf{x}_*$ for k large enough.

Corollary 3. *Under the above assumptions on $\mathcal{C}$, let $\mathcal{A}^{(i)}$ be the AFW for $i \in [1\!:\!m]$, and let strict complementarity conditions hold at $\mathbf{x}_* \in \mathcal{C}$. If $\mathbf{x}_k \to \mathbf{x}_*$ (almost surely), then for parallel and GS selection (for random sampling) we have $\mathrm{supp}(\mathbf{x}_k) = \mathrm{supp}(\mathbf{x}_*)$ for k large enough.*

This result has relevant practical implications, especially when handling sparse optimization problems. Since the algorithm iterates have a constant support when k is large, we can simply focus on the few support components and forget about the others in this case. We hence can exploit this by embedding sophisticated tools (like, e.g., caching strategies, second-order methods) in the algorithm, thus obtaining a significant speed up in the end.

5 Numerical results

We report here some preliminary numerical results for a non-convex quadratic optimization problem referred to as Multi-StQP [15] on a product of (here identical) simplices, that is

$$\min \left\{\mathbf{x}^\top \mathsf{Q}\mathbf{x} : \mathbf{x} \in (\Delta^l)^m\right\}. \tag{23}$$

The matrix Q was generated in such a way that the solutions of problem (23) had components sparse but different from vertices. This is in fact the setting where FW variants have proved to be more effective [14, 34]. In order to obtain the desired property, we consider a perturbation of a stochastic StQP [10]. Given $\{\bar{\mathsf{Q}}_i\}_{i\in[1:m]}$ representing m possible StQPs, with $\bar{\mathsf{Q}}_i \in \mathbb{R}^{l\times l}$ for $i \in [1 : m]$, the corresponding stochastic StQP with sample space $[1\!:\!m]$ is given by

$$\max \left\{\sum_{i=1}^{m} p_i \mathbf{y}_i^\top \bar{\mathsf{Q}}_i \mathbf{y}_i : \mathbf{y}_i \in \Delta^l \quad \text{for all } i \in [1\!:\!m]\right\}. \tag{24}$$

with p_i probability of the StQP i. Equivalently, (24) is an instance of problem (23) with $\mathsf{Q} = \bar{\mathsf{Q}}$, for

$$\bar{\mathsf{Q}} = \begin{bmatrix} -p_1\bar{\mathsf{Q}}_1 & 0 & \cdots & 0 \\ 0 & -p_2\bar{\mathsf{Q}}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -p_m\bar{\mathsf{Q}}_m \end{bmatrix}. \tag{25}$$

In our tests, we added to the stochastic StQP a perturbation coupling the blocks. More precisely, the matrix Q was set equal to $\bar{\mathsf{Q}} + \varepsilon\tilde{\mathsf{Q}}$, for $\tilde{\mathsf{Q}}$ a random matrix with standard Gaussian independent entries. The coefficient ε was set equal to $\frac{1}{2m^2}$. We set $\bar{\mathsf{Q}}_i = \bar{\mathsf{A}}_i + \alpha\mathsf{I}_l$, for $\alpha = 0.5$ and $\bar{\mathsf{A}}_i$ the adjacency matrix of an Erdős-Rényi random graph, where each couple of vertices has probability p of being connected by an edge, independently from the other couples. Hence, for $i \in [1:m]$ the problem

$$\min\left\{-\mathbf{y}^\top\bar{\mathsf{Q}}_i\mathbf{y} : \mathbf{y} \in \Delta^l\right\} \tag{26}$$

is a regularized maximum-clique formulation, where each maximal clique corresponding to a unique strict local maximizer with support equal to its vertices, and conversely (see [9] and references therein). The probability p is set as follows

$$p = \binom{l}{s}^{\frac{2}{s(s-1)}}, \tag{27}$$

for s the nearest integer to $0.4l$, so that the expected number of cliques with size $\approx 0.4l$ is 1, (see, e.g., [2]). Notice that the perturbation term $\tilde{\mathsf{Q}}$ ensures that problem (23) cannot be solved by optimizing each block separately.

We remark here that different ways to build large StQPs starting from smaller instances and preserving the structure of their solutions have been discussed in [16]. However, while the resulting problems decouple on the feasible set of the larger problem, they still decouple on the product of the feasible sets of the smaller instances, and for our purposes are equivalent to the block diagonal structure.

We tested four methods in total: AFW + SSC with parallel, GS and randomized updates (PAFW + SSC, GSAFW + SSC, BCAFW + SSC respectively), and FW with randomized updates (BCFW, coinciding with the block coordinate FW introduced in [24]). Our tests focused on the local identification and on the convergence properties of our methods.

The code was written in `Python` using the `numpy` package, and the tests were performed on an Intel Core i9-12900KS CPU 3.40GHz, 32GB RAM. The codes relevant to the numerical tests are available at the following link:
`https://github.com/DamianoZeffiro/Projection-free-product-domain`.

5.1 Multistart

We first considered a multistart approach, where the results are averaged across 20 runs, choosing 4 starting points for each of 5 random initializations of the objective.

We measure both optimality gap (error estimate) and sparsity (number of nonzero components, ℓ_0 norm) of the iterates, reporting average and standard deviation in the plots. The estimated global optimum used in the optimality gap is obtained by subtracting 10^{-5} from the best local solution found by the algorithms. We mostly consider the performance with respect to block gradient computations, with one gradient counted each time the SSC is performed in one of the blocks, as in previous works (see, e.g., [24]). In some tests involving the GSAFW + SSC, we consider instead

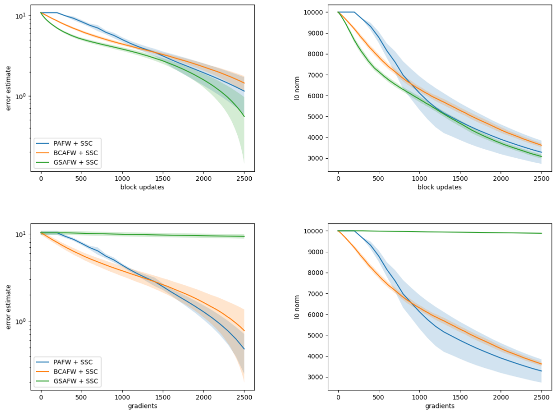

Figure 1: Comparison using multistart between GSAFW + SSC, PAFW + SSC and BCAFW + SSC. $l = m = 100$.

block updates, with one block update counted each time the algorithms modifies the current iterate in one of the blocks.

We first compare PAFW + SSC, BCAFW + SSC and GSAFW + SSC (Figure 1). As expected, while GSAFW + SSC shows good performance with respect to block updates, it has a very poor performance with respect to block gradient computations, since at every iteration m gradients must be computed to update a single block. We then compared PAFW + SSC, BCAFW + SSC and BCFW. The results (Figure 2) clearly show that PAFW + SSC and BCAFW + SSC outperform BCFW. All these findings are consistent with the theoretical results described in Section 7.2.

5.2 Monotonic basin hopping

We then consider the monotonic basin hopping approach (see, e.g., [26, 29]) described in Algorithm 3. The method computes a local optimizer $\mathbf{x}_{*,i}$ close to the current iterate $\bar{\mathbf{x}}_i$ (Step 2). There $\mathcal{M}$ is a local optimization algorithm, and given as input $\mathcal{M}$ and $\bar{\mathbf{x}}_i$, the subroutine LO returns the result of applying $\mathcal{M}$ starting from $\bar{\mathbf{x}}_i$, with a suitable stopping criterion which in our case is given by a limit on the number of gradient computations, set to $10m$. The sequence of best points found in the first i iterations $\{\bar{\mathbf{x}}_{*,i}\}$ is updated in Step 3, and in Step 5, $\bar{\mathbf{x}}_{i+1}$ is chosen in a neighborhood of $\bar{\mathbf{x}}_{*,i}$. The neighborhood $B(\mathbf{x}, \gamma)$ for $\mathbf{x} \in \mathcal{C}$ and $\gamma \in (0, 1]$ is given by

$$B(\mathbf{x}, \gamma) = \{\mathbf{x} + \gamma(\mathbf{y} - \mathbf{x}) : \mathbf{y} \in \mathcal{C}\} . \tag{28}$$

In the tests, we chose $\mathbf{y}$ uniformly at random in $\mathcal{C}$ and set $\bar{\mathbf{x}}_{i+1} = \bar{\mathbf{x}}_i + \gamma(\mathbf{y} - \bar{\mathbf{x}}_i)$, with $\gamma = 0.25$. The methods we consider as subroutines in Step 2 are PAFW + SSC, BCAFW + SSC and BCFW. We set $i_{\max} = 9$, and perform 10 runs of Algorithm 3, randomly initializing the starting point. We plot once again average and standard deviation for $\{f(\bar{\mathbf{x}}_{*,i}) - \tilde{f}^*\}$ with $\tilde{f}^*$ estimating the global optimum (obtained by subtracting 10^{-1} from the best solution found by the methods).

The results again show that PAFW + SSC and BCAFW + SSC find better solutions than BCFW, with BCAFW + SSC outperforming PAFW + SSC in most instances if $l \leq m$.

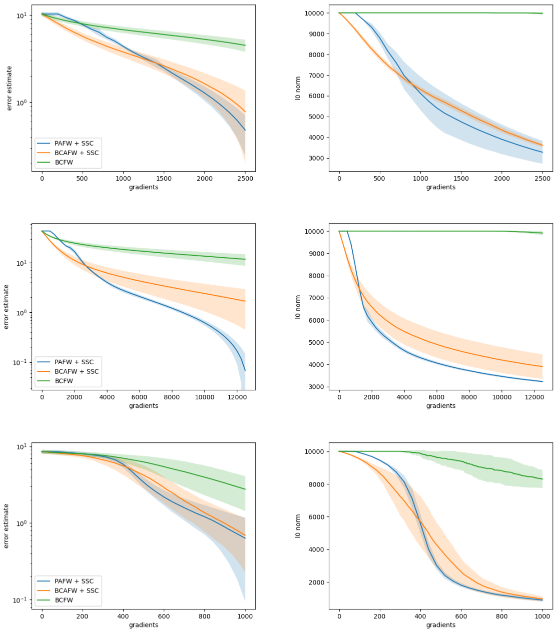

Figure 2: Comparison using multistart between BCFW, PAFW + SSC and BCAFW + SSC. $l = m = 100$ in the first row, $l = 40$ and $m = 250$ in the second row, $l = 250$ and $m = 40$ in the third row.

Algorithm 3 Monotonic Basin Hopping Strategy

```
1: $\bar{\mathbf{x}}_0 \in \mathcal{C}$, $i_{\max} \in \mathbb{N}$, $\gamma \in [0,1]$, $i = 0$. Set $\bar{\mathbf{x}}_{*,-1} = \bar{\mathbf{x}}_0$
2: Compute a local optimizer $\mathbf{x}_{*,i} = \mathrm{LO}(\mathcal{M}, \bar{\mathbf{x}}_i)$
3: If $f(\mathbf{x}_{*,i}) < f(\bar{\mathbf{x}}_{*,i-1})$ then set $\bar{\mathbf{x}}_{*,i} = \mathbf{x}_{*,i}$, else set $\bar{\mathbf{x}}_{*,i} = \bar{\mathbf{x}}_{*,i-1}$.
4: If $i = i_{\max}$, then STOP.
5: Randomly chose $\bar{\mathbf{x}}_{i+1}$ in $B(\bar{\mathbf{x}}_{*,i}, \gamma)$.
6: Set $i = i + 1$. Go to step 2.
```

6 Conclusions

For a quite general optimization problem on product domains, we offer a seemingly new convergence theory, which ensures both convergence of objective values and (local) linear convergence of the iterates under widely accepted conditions, for block-coordinate FW variants. Convergence is global for μ-strongly convex objectives, but we mainly focus on the non-convex case. In case of randomized selection of the blocks, all results are in expectation, and need a further technical assumption. As usual, constants and rates are specified in terms of the Lipschitz constant L for the gradient map, the constant μ used in the local Kurdyka-Łojasiewicz-condition, and the parameter τ in the so-called angle condition.

The results are complemented by an active set identification result for a specific structure of the product domain and suitable choices of a projection-free strategy (FW-approach with away steps for the search direction): it is proved that our framework identifies the support of a solution in a

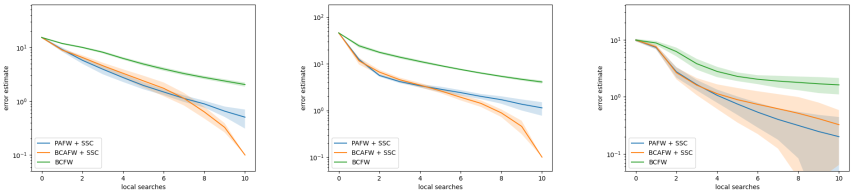

Figure 3: Comparison using Monotonic Basin Hopping with BCFW, PAFW + SSC and BCAFW + SSC. From left to right: $l = m = 100$; $l = 40$ and $m = 250$; $l = 250$ and $m = 40$.

finite number of iterations.

To the best of our knowledge, this is the first time that both a linear convergence rate and an active set identification result are given for (bad step-free) block-coordinate FW variants, in an effort to narrow the research gap observed in [31].

In our preliminary experiments, numerical evidence clearly points out the advantages of our strategy to exploit structural knowledge. On randomly generated non-convex Multi-StQPs where easy instances were carefully avoided, our approach (AFW with parallel or randomized updates, both combined with the Short Step Chain strategy SSC) is dominating the block-coordinate FW method with randomized updates.

We tested resilience of our reported observations by employing two experimental setups, pure multistart and monotonic basin hopping. The same effects seem to prevail.

Instance construction was motivated by a stochastic variant of the StQP, varying both domain dimension l and number m of possible scenarios. In case $l \leq m$ there seems to be a slight edge towards the combination of AFW with randomized updates and SSC, compared to the parallel variant. This effect does not seem to happen with large l in comparison to m, but would not change superiority over traditional block-coordinate FW methods.

7 Appendix

7.1 Proofs

In the rest of this section, we always assume that the SSC terminates in a finite number of steps and that the angle condition holds.

Lemma 4. *For a fixed $i \in [1\!:\!m]$, let $\{\mathbf{w}_k\} = \{\mathbf{x}_k^{(i)}\}$, and let $\mathbf{w}_{k+1} = \mathrm{SSC}(\mathbf{w}_k, \mathbf{g})$. Then there exists $\tilde{\mathbf{w}}_k \in \{\mathbf{y}_j\}_{j=0}^T$ such that*

$$\|\mathbf{w}_{k+1} - \mathbf{w}_k\| \geq \frac{\tau}{L} \|\pi(T_{\mathcal{C}_{(i)}}(\tilde{\mathbf{w}}_k), \mathbf{g})\| \tag{29}$$

and

$$\|\tilde{\mathbf{w}}_k - \mathbf{w}_k\| \leq \|\mathbf{w}_{k+1} - \mathbf{w}_k\|\,, \tag{30}$$

$$\langle \mathbf{g}, \tilde{\mathbf{w}}_k - \mathbf{w}_k \rangle \leq \langle \mathbf{g}, \mathbf{w}_{k+1} - \mathbf{w}_k \rangle\,. \tag{31}$$

Furthermore, we have

$$L\|\mathbf{y} - \mathbf{w}_k\|^2 \leq \langle \mathbf{g}, \mathbf{y} - \mathbf{w}_k \rangle \tag{32}$$

for $y \in \{\mathbf{w}_{k+1}, \tilde{\mathbf{w}}_k\}$.

Proof. Let $\bar{B} = \bar{B}_{\frac{\|\mathbf{g}\|}{2L}}(\bar{\mathbf{x}} + \frac{\mathbf{g}}{2L})$ and let T be such that $\mathbf{w}_{k+1} = \mathbf{y}_T$. By [34, (4.4)] we have that (32) holds for every $\mathbf{z} \in \bar{B}$ (in place of $\mathbf{y}$), and therefore as desired for every

$$\mathbf{y} \in \{\mathbf{w}_{k+1}, \tilde{\mathbf{w}}_k\} \subset \{\mathbf{y}_j : j \in [0\!:\!T]\} \subset \bar{B}\,.$$

Let now $\tilde{p}_j = \|\pi(T_{\mathcal{C}_{(i)}}(\mathbf{y}_j), \mathbf{g})\|$. Notice that, if $\tilde{\mathbf{w}}_k = \mathbf{y}_l$, then

$$\frac{\tau}{L}\,\|\pi(T_{\mathcal{C}_{(i)}}(\tilde{\mathbf{w}}_k), \mathbf{g})\| = \frac{\tau}{L}\,\tilde{p}_l \leq \frac{1}{L}\langle \mathbf{g}, \hat{\mathbf{d}}_l\rangle\,, \tag{33}$$

where the inequality follows from $\frac{\langle \mathbf{g}, \hat{\mathbf{d}}_l\rangle}{\tilde{p}_l} \geq \mathrm{DSB}_{\mathcal{A}}(\mathcal{C}_{(i)}, \mathbf{y}_l, \mathbf{g}) \geq \mathrm{SB}_{\mathcal{A}}(\mathcal{C}_{(i)}) = \tau$. Thus for proving (29), in the rest of the proof it will be enough to prove

$$\langle \mathbf{g}, \hat{\mathbf{d}}_l\rangle \leq L\|\mathbf{w}_{k+1} - \mathbf{w}_k\|\,. \tag{34}$$

Furthermore, since by definition of the SSC, the scalar product $\langle \mathbf{g}, \mathbf{y}_j\rangle$ is increasing in j, we have

$$\langle \mathbf{g}, \tilde{\mathbf{w}}_k - \mathbf{w}_k\rangle = \langle \mathbf{g}, \mathbf{y}_l - \mathbf{w}_k\rangle \leq \langle \mathbf{g}, \mathbf{y}_T - \mathbf{w}_k\rangle = \langle \mathbf{g}, \mathbf{w}_{k+1} - \mathbf{w}_k\rangle\,.$$

We distinguish four cases, according to how the SSC terminates. In the first two, we show we can choose the last step, $\tilde{\mathbf{w}} = \mathbf{y}_T$; in the third, the penultimate choice $\tilde{\mathbf{w}} = \mathbf{y}_{T-1}$ satisfies all conditions, and in the fourth case, an intermediate step is an appropriate choice. We abbreviate $B_j = \bar{B}_{\langle \mathbf{g}, \hat{\mathbf{d}}_j\rangle/L}(\mathbf{w}_k)$.

Case 1: $T = 0$ or $\mathbf{d}_T = \mathbf{o}$. Since there are no descent directions, $\mathbf{w}_{k+1} = \mathbf{y}_T$ must be stationary for the gradient $-\mathbf{g}$. Equivalently, $\tilde{p}_T = \|\pi(T_{\mathcal{C}_{(i)}}(\mathbf{w}_{k+1}), \mathbf{g})\| = 0$. Finally, it is clear that if $T = 0$ then $\mathbf{d}_0 = \mathbf{o}$, since $\mathbf{y}_0$ must be stationary for $-\mathbf{g}$. Thus taking $\tilde{\mathbf{w}}_k = \mathbf{y}_T$ the desired properties follow.

Before examining the remaining cases we remark that if the SSC terminates in Phase II, then $\alpha_{T-1} = \beta_{T-1}$ must be maximal w.r.t. the conditions $\mathbf{y}_T \in B_{T-1}$ or $\mathbf{y}_T \in \bar{B}$. If $\alpha_{T-1} = 0$ then $\mathbf{y}_{T-1} = \mathbf{y}_T$, and in this case we cannot have $\mathbf{y}_{T-1} \in \partial\bar{B}$, otherwise the SSC would terminate in Phase II of the previous cycle. Therefore necessarily $\mathbf{y}_T = \mathbf{y}_{T-1} \in \mathrm{int}(B_{T-1})^c$ (Case 2). If $\beta_{T-1} = \alpha_{T-1} > 0$ we must have $\mathbf{y}_{T-1} \in \mathcal{C}_{T-1} = B_{T-1} \cap \bar{B}$, and $\mathbf{y}_T \in \partial B_{T-1}$ (Case 3) or $\mathbf{y}_T \in \partial\bar{B}$ (Case 4) respectively.

Case 2: $\mathbf{y}_{T-1} = \mathbf{y}_T \in \mathrm{int}(B_{T-1})^c$. We can rewrite the condition as

$$\langle \mathbf{g}, \hat{\mathbf{d}}_{T-1}\rangle \leq L\|\mathbf{y}_{T-1} - \mathbf{w}_k\| = L\|\mathbf{y}_T - \mathbf{w}_k\|\,, \tag{35}$$

which is exactly (34). Then $\tilde{\mathbf{w}}_k = \mathbf{w}_{k+1} = \mathbf{y}_T$ satisfies the desired conditions.

Case 3: $\mathbf{y}_T = \mathbf{y}_{T-1} + \beta_{T-1}\mathbf{d}_{T-1}$ and $\mathbf{y}_T \in \partial B_{T-1}$. Then from $\mathbf{y}_{T-1} \in B_{T-1}$ it follows

$$L\|\mathbf{y}_{T-1} - \mathbf{w}_k\| \leq \langle \mathbf{g}, \hat{\mathbf{d}}_{T-1}\rangle\,, \tag{36}$$

and $\mathbf{y}_T \in \partial B_{T-1}$ implies

$$\langle \mathbf{g}, \hat{\mathbf{d}}_{T-1}\rangle = L\|\mathbf{y}_T - \mathbf{w}_k\|\,, \tag{37}$$

which is (34) for $l = T - 1$. Combining (36) with (37) we also obtain

$$L\|\mathbf{y}_{T-1} - \mathbf{w}_k\| \leq L\|\mathbf{y}_T - \mathbf{w}_k\|\,, \tag{38}$$

so that in particular we can take $\tilde{\mathbf{w}}_k = \mathbf{y}_{T-1}$.

Case 4: $\mathbf{y}_T = \mathbf{y}_{T-1} + \beta_{T-1}\mathbf{d}_{T-1}$ and $\mathbf{y}_T \in \partial\bar{B}$.

The condition $\mathbf{w}_{k+1} = \mathbf{y}_T \in \partial\bar{B}$ can be rewritten as

$$L\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 - \langle \mathbf{g}, \mathbf{w}_{k+1} - \mathbf{w}_k\rangle = 0\,. \tag{39}$$

For every $j \in [0\,{:}\,T]$ we have

$$\mathbf{w}_{k+1} = \mathbf{y}_j + \sum_{i=j}^{T-1} \alpha_i \mathbf{d}_i \,. \tag{40}$$

We now want to prove that for every $j \in [0\,{:}\,T]$

$$\|\mathbf{w}_{k+1} - \mathbf{w}_k\| \geq \|\mathbf{y}_j - \mathbf{w}_k\| \,. \tag{41}$$

Indeed, we have

$$\begin{aligned} L\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 = \langle \mathbf{g}, \mathbf{w}_{k+1} - \mathbf{w}_k \rangle = \langle \mathbf{g}, \mathbf{y}_j - \mathbf{w}_k \rangle + \sum_{i=j}^{T-1} \alpha_i \langle \mathbf{g}, \mathbf{d}_i \rangle \\ \geq \langle \mathbf{g}, \mathbf{y}_j - \mathbf{w}_k \rangle \geq L\|\mathbf{y}_j - \mathbf{w}_k\|^2 \,, \end{aligned}$$

where we used (39) in the first equality, (40) in the second, $\langle g, \mathbf{d}_j \rangle \geq 0$ for every j in the first inequality and $\mathbf{y}_j \in \bar{B}$ in the second equality, which proves (41).
We also have

$$\frac{\langle \mathbf{g}, \mathbf{w}_{k+1} - \mathbf{w}_k \rangle}{\|\mathbf{w}_{k+1} - \mathbf{w}_k\|} = \frac{\langle \mathbf{g}, \sum_{j=0}^{T-1} \alpha_j \mathbf{d}_j \rangle}{\|\sum_{j=0}^{T-1} \alpha_j \mathbf{d}_j\|} \geq \frac{\langle \mathbf{g}, \sum_{j=0}^{T-1} \alpha_j \mathbf{d}_j \rangle}{\sum_{j=0}^{T-1} \alpha_j \|\mathbf{d}_j\|} \geq \min \left\{ \frac{\langle \mathbf{g}, \mathbf{d}_j \rangle}{\|\mathbf{d}_j\|} : j \in [0\,{:}\,T-1] \right\} . \tag{42}$$

Thus for $\tilde{T} \in \operatorname{argmin} \left\{ \frac{\langle \mathbf{g}, \mathbf{d}_j \rangle}{\|\mathbf{d}_j\|} : j \in [0\,{:}\,T-1] \right\}$ we have

$$\langle g, \hat{\mathbf{d}}_{\tilde{T}} \rangle \leq \frac{\langle \mathbf{g}, \mathbf{w}_{k+1} - \mathbf{w}_k \rangle}{\|\mathbf{w}_{k+1} - \mathbf{w}_k\|} = L\|\mathbf{w}_{k+1} - \mathbf{w}_k\| \,, \tag{43}$$

where we used (42) in the first inequality and (39) in the second (equality).
In particular $\tilde{\mathbf{w}}_k = \mathbf{y}_{\tilde{T}}$ satisfies the desired properties, where $\|\tilde{\mathbf{w}}_k - \mathbf{w}_k\| \leq \|\mathbf{w}_{k+1} - \mathbf{w}_k\|$ by (41) and (34) holds by (43). □

We denote by $\overline{\text{SSC}}(\mathbf{w}_k, \mathbf{g})$ a point $\tilde{\mathbf{w}}_k$ with the properties stated in the above lemma. It is also useful to define U_0 as the connected component of $\{\mathbf{x} \in \mathcal{C} : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ containing $\mathbf{x}_0$.

Lemma 5. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1. Let $\bar{\mathbf{x}}_k = [\overline{\text{SSC}}(\mathbf{x}_k^{(i)}, -\nabla f(\mathbf{x}_k)^{(i)})]_{i=1}^m$. Assume also that $f(\mathbf{x}_*)$ is a minimum in U_0. Then, $\{f(\mathbf{x}_k)\}$ is decreasing, and for every k, $\{\mathbf{x}_k, \bar{\mathbf{x}}_k\} \subset U_0$, with $f(\mathbf{y}) \in [f(\mathbf{x}_*), f(\mathbf{x}_0)]$ for $\mathbf{y} \in \{\mathbf{x}_k, \bar{\mathbf{x}}_k\}$.*

Proof. Let U_k be the minimal connected component of $\{\mathbf{x} \in \mathcal{C} : f(\mathbf{x}) \leq f(\mathbf{x}_k)]$ containing $\mathbf{x}_k$, let $\mathbf{g} = -\nabla f(\mathbf{x}_k)$ and let $\bar{B}_k^{\mathcal{C}} = \mathcal{C} \cap \prod_i \bar{B}_{\frac{\|\mathbf{g}^{(i)}\|}{2L}}(\mathbf{x}_k^{(i)} + \frac{\mathbf{g}^{(i)}}{2L})$. For $\mathbf{y} \in \bar{B}_k^{\mathcal{C}}$, we have $\mathbf{x}_k \in \bar{B}_k^{\mathcal{C}}$ and

$$\begin{aligned} & f(\mathbf{y}) \leq f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{y} - \mathbf{x}_k \rangle + \frac{L}{2}\|\mathbf{y} - \mathbf{x}_k\|^2 \\ & = f(\mathbf{x}_k) + \sum_{i=1}^m \langle \nabla f(\mathbf{x}_k)^{(i)}, \mathbf{y}^{(i)} - \mathbf{x}_k^{(i)} \rangle + \frac{L}{2}\|\mathbf{y}^{(i)} - \mathbf{x}_k^{(i)}\|^2 \leq f(\mathbf{x}_k) - \frac{L}{2} \sum_{i=1}^m \|\mathbf{y}^{(i)} - \mathbf{x}_k^{(i)}\|^2 \\ & = f(\mathbf{x}_k) - \frac{L}{2}\|\mathbf{x}_k - \mathbf{y}\|^2 \end{aligned} \tag{44}$$

where we used the standard Descent Lemma in the first inequality and the the second follows by definition of $\bar{B}_k^{\mathcal{C}}$. From (44) it follows that $\{f(\mathbf{x}_k)\}$ is decreasing, and that $\bar{B}_k^{\mathcal{C}} \subset \{\mathbf{x} \in \mathcal{C} : f(\mathbf{x}) \leq$

$f(\mathbf{x}_k)\}$. Furthermore, since $\bar{B}_k^{\mathcal{C}}$ is connected and contains $\mathbf{x}_k$, the stronger inclusion $\bar{B}_k^{\mathcal{C}} \subset U_k$ is also true. Thus $\{\mathbf{x}_{k+1}, \bar{\mathbf{x}}_k\} \subset \bar{B}_k^{\mathcal{C}} \subset U_k$, so that in particular $U_{k+1} \subset U_k$ since $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$, and by induction we can conclude $\{\mathbf{x}_{k+1}, \bar{\mathbf{x}}_k\} \subset U_0$. Finally, $f(\mathbf{y}) \in [f(\mathbf{x}_*), f(\mathbf{x}_k)]$ for $\mathbf{y} \in \{\mathbf{x}_{k+1}, \bar{\mathbf{x}}_k\}$, where the lower bound follows from the assumption that $f(\mathbf{x}_*)$ is a minimum in U_0, and the upper bound follows from (44). □

Lemma 6. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1, and assume that the angle condition holds for the method $\mathcal{A}^{(i)}$ with the same τ, for all $i \in [1\!:\!m]$. Let $\bar{\mathbf{x}}_k = [\overline{\mathrm{SSC}}(\mathbf{x}_k^{(i)}, -\nabla f(\mathbf{x}_k)^{(i)})]_{i=1}^m$ and $\tilde{\mathbf{x}}_{k+1} = [\mathrm{SSC}(\mathbf{x}_k^{(i)}, -\nabla f(\mathbf{x}_k)^{(i)})]_{i=1}^m$. If (14) holds at $\bar{\mathbf{x}}_k$, we then have, abbreviating $\mathbf{g} = -\nabla f(\mathbf{x}_k)$:*

$$\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^2 \geq \frac{\tau^2}{2(1+\tau^2)L^2}\|\pi(T_{\mathcal{C}}(\bar{\mathbf{x}}_k), -\nabla f(\bar{\mathbf{x}}_k))\|^2 \;\geq \frac{\tau^2\mu}{L^2(1+\tau^2)}[f(\bar{\mathbf{x}}_k) - f^*]\,, \tag{45}$$

$$\frac{1}{2}\langle \mathbf{g}, \tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\rangle \geq \frac{1}{3}[f(\mathbf{x}_k) - f(\bar{\mathbf{x}}_k)]\,. \tag{46}$$

Proof. Let $\bar{\mathbf{g}} = -\nabla f(\bar{\mathbf{x}}_k)$, $\bar{q}_{(i)} = \|\pi(T_{\mathcal{C}_{(i)}}(\bar{\mathbf{x}}_k^{(i)}), \bar{\mathbf{g}}^{(i)})\|$, and $q_{(i)} = \|\pi(T_{\mathcal{C}_{(i)}}(\bar{\mathbf{x}}_k^{(i)}), \mathbf{g}^{(i)})\|$. Observe that by the Lipschitz continuity of the gradient, we have the inequality

$$\bar{q}_{(i)} \leq q_{(i)} + L\|\bar{\mathbf{x}}_k^{(i)} - \mathbf{x}_k^{(i)}\| \tag{47}$$

and thus

$$\bar{q}_{(i)}^2 \leq 2q_{(i)}^2 + 2L^2\|\bar{\mathbf{x}}_k^{(i)} - \mathbf{x}_k^{(i)}\|^2 \leq \frac{2L^2(1+\tau^2)}{\tau^2}\|\tilde{\mathbf{x}}_{k+1}^{(i)} - \mathbf{x}_k^{(i)}\|^2\,, \tag{48}$$

where we applied Jensen's inequality to (47) in the first inequality, and (29) together with (30) in the second inequality.

Thus we can write

$$\begin{aligned}\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^2 &= \sum_{i=1}^m \|\tilde{\mathbf{x}}_{k+1}^{(i)} - \mathbf{x}_k^{(i)}\|^2 \geq \frac{\tau^2}{2L^2(1+\tau^2)}\sum_{i=1}^m \bar{q}_{(i)}^2 \\ &= \frac{\tau^2}{2L^2(1+\tau^2)}\|\pi(T_{\mathcal{C}}(\bar{\mathbf{x}}_k), -\nabla f(\bar{\mathbf{x}}_k))\|^2 \;\geq \frac{\tau^2\mu}{L^2(1+\tau^2)}[f(\bar{\mathbf{x}}_k) - f^*]\,,\end{aligned} \tag{49}$$

where we used (48) in the first inequality and the KL property in the second. This proves (45). Using the standard Descent Lemma, we can give the upper bound

$$f(\mathbf{x}_k) - f(\bar{\mathbf{x}}_k) \leq \langle \mathbf{g}, \bar{\mathbf{x}}_k - \mathbf{x}_k\rangle + \tfrac{L}{2}\|\bar{\mathbf{x}}_k - \mathbf{x}_k\|^2 \leq \tfrac{3}{2}\langle \mathbf{g}, \bar{\mathbf{x}}_k - \mathbf{x}_k\rangle = \tfrac{3}{2}\sum_{i=1}^m \langle \mathbf{g}^{(i)}, \bar{\mathbf{x}}_k^{(i)} - \mathbf{x}_k^{(i)}\rangle\,, \tag{50}$$

where we used (32) in the second inequality. We can finally prove (46):

$$\tfrac{1}{2}\langle \mathbf{g}, \tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\rangle \geq \tfrac{1}{2}\langle \mathbf{g}, \bar{\mathbf{x}}_k - \mathbf{x}_k\rangle = \tfrac{1}{2}\sum_{i=1}^m \langle \mathbf{g}^{(i)}, \bar{\mathbf{x}}_k^{(i)} - \mathbf{x}_k^{(i)}\rangle \geq \tfrac{1}{3}[f(\mathbf{x}_k) - f(\bar{\mathbf{x}}_k)]\,, \tag{51}$$

where we used (31) in the first inequality and (50) in the second one. □

Lemma 7. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1, and assume that the angle condition holds for the method $\mathcal{A}^{(i)}$ with the same τ, for all $i \in [1\!:\!m]$. Let $\bar{\mathbf{x}}_k = (\overline{\mathrm{SSC}}(\mathbf{x}_k^{(i)}, -\nabla f(\mathbf{x}_k)^{(i)}))_{i=1}^m$. Then, if the KL property (14) holds at $\bar{\mathbf{x}}_k$, for parallel updates*

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \frac{\tau^2\mu}{2L(1+\tau^2)}\left(f(\bar{\mathbf{x}}_k) - f^*\right), \tag{52}$$

for GS updates

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \frac{\tau^2\mu}{2L(1+\tau^2)}\,\frac{1}{m}\left(f(\bar{\mathbf{x}}_k) - f^*\right), \tag{53}$$

and for random updates

$$\mathbb{E}[f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})] \geq \frac{\tau^2\mu}{2L(1+\tau^2)}\,\frac{1}{m}\,\mathbb{E}[f(\bar{\mathbf{x}}_k) - f^*]\,. \tag{54}$$

Proof. We first prove the inequality for parallel updates. We have

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) &\geq \frac{L}{2}\,\|\mathbf{x}_k - \mathbf{x}_{k+1}\|^2 \geq \frac{L}{2}\,\frac{\tau^2}{2L^2(1+\tau^2)}\,\|\pi(T_{\mathcal{C}}(\bar{\mathbf{x}}_k), -\nabla f(\bar{\mathbf{x}}_k))\|^2 \\ &= \frac{\tau^2}{4L(1+\tau^2)}\|\pi(T_{\mathcal{C}}(\bar{\mathbf{x}}_k), -\nabla f(\bar{\mathbf{x}}_k))\|^2 \;\geq \frac{\tau^2\mu}{2L^2(1+\tau^2)}\,[f(\bar{\mathbf{x}}_k) - f^*]\,, \end{aligned} \tag{55}$$

where the first inequality follows from (44), the second inequality by (45) where with the notation introduced in Lemma 6 we have by definition $\mathbf{x}_{k+1} = \tilde{\mathbf{x}}_{k+1}$. For GS updates, we have

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) &\geq \langle g, \mathbf{x}_{k+1} - \mathbf{x}_k\rangle - \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \geq \frac{1}{2}\langle \mathbf{g}, \mathbf{x}_{k+1} - \mathbf{x}_k\rangle \\ &= \frac{1}{2}\max_{i\in[1:m]}\langle \mathbf{g}^{(i)}, \tilde{\mathbf{x}}_{k+1}^{(i)} - \mathbf{x}_k\rangle \geq \frac{1}{2m}\langle \mathbf{g}, \tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\rangle \geq \frac{L}{2m}\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^2 \\ &\geq \frac{\tau^2\mu}{2mL(1+\tau^2)}\,[f(\bar{\mathbf{x}}_k) - f^*]\,, \end{aligned} \tag{56}$$

where in the first inequality we used the standard Descent Lemma, (32) in the second inequality; the equality follows by definition of GS updates, in the fourth inequality we applied again (32), and (45) in the last one.

Finally, for random updates we have, denoting as $i(k) = j$ the event that the index chosen at the step k is j:

$$\begin{aligned} \mathbb{E}[f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})] &\geq \frac{L}{2}\mathbb{E}[\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2] = \frac{L}{2}\sum_{j=1}^{m}\mathbb{P}(i(k) = j)\mathbb{E}[\|\tilde{\mathbf{x}}_{k+1}^{(j)} - \mathbf{x}_k^{(j)}\|^2] \\ &= \frac{L}{2m}\sum_{j=1}^{m}\mathbb{E}[\|\tilde{\mathbf{x}}_{k+1}^{(j)} - \mathbf{x}_k^{(j)}\|^2] = \frac{L}{2m}\mathbb{E}[\|\tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^2] \geq \frac{\tau^2\mu}{2L(1+\tau^2)}\mathbb{E}[f(\bar{\mathbf{x}}_k) - f^*]\,, \end{aligned} \tag{57}$$

where the first inequality follows from (44), we used $\mathbb{P}(\{i(k) = j\}) = \frac{1}{m}$ in the second equality and (45) in the last inequality. □

Lemma 8. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1, and assume that the angle condition holds for the method $\mathcal{A}^{(i)}$ with the same τ, for all $i \in [1:m]$. Let $\bar{\mathbf{x}}_k = (\overline{\mathrm{SSC}}(\mathbf{x}_k^{(i)}, -\nabla f(\mathbf{x}_k)^{(i)}))_{i=1}^m$. Then for parallel updates*

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \frac{1}{3}[f(\mathbf{x}_k) - f(\bar{\mathbf{x}}_k)]\,, \tag{58}$$

for GS updates

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \frac{1}{3m}[f(\mathbf{x}_k) - f(\bar{\mathbf{x}}_k)]\,, \tag{59}$$

and for random updates

$$\mathbb{E}[f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})] \geq \frac{1}{3m}\mathbb{E}[f(\mathbf{x}_k) - f(\bar{\mathbf{x}}_k)] \tag{60}$$

Proof. For parallel updates, we have

$$\begin{aligned} & f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \langle \mathbf{g}, \mathbf{x}_{k+1} - \mathbf{x}_k \rangle - \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \geq \frac{1}{2}\langle \mathbf{g}, \mathbf{x}_{k+1} - \mathbf{x}_k \rangle \\ & = \frac{1}{2}\langle \mathbf{g}, \tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k \rangle \geq \frac{1}{3}[f(\mathbf{x}_k) - f(\bar{\mathbf{x}}_k)]\,, \end{aligned} \tag{61}$$

where we have used the standard descent Lemma in the first inequality, (32) in the second inequality, and (46) in the last inequality.
The proof follows analogously for GS updates, after noticing

$$\langle \mathbf{g}, \mathbf{x}_{k+1} - \mathbf{x}_k \rangle \geq \frac{1}{m}\langle \mathbf{g}, \tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k \rangle\,, \tag{62}$$

as showed in (56), and for random updates, using

$$\mathbb{E}[\langle \mathbf{g}, \mathbf{x}_{k+1} - \mathbf{x}_k \rangle] = \frac{1}{m}\mathbb{E}[\langle \mathbf{g}, \tilde{\mathbf{x}}_{k+1} - \mathbf{x}_k \rangle]\,, \tag{63}$$

respectively. □

Lemma 9. *Let $\{\mathbf{x}_k\}$ be a sequence generated by Algorithm 1, and assume that the angle condition holds for the method $\mathcal{A}^{(i)}$ with the same τ, for all $i \in [1\!:\!m]$. Then, if the KL property* (14) *holds at $\mathbf{x}_k$, for parallel updates*

$$f(\mathbf{x}_{k+1}) - f^* \leq q_P(f(\mathbf{x}_k) - f^*)\,, \tag{64}$$

for GS updates

$$f(\mathbf{x}_{k+1}) - f^* \leq q_{GS}(f(\mathbf{x}_k) - f^*)\,, \tag{65}$$

and for random updates

$$\mathbb{E}[f(\mathbf{x}_{k+1}) - f^*] \leq q_R\mathbb{E}[f(\mathbf{x}_k) - f^*]\,. \tag{66}$$

Proof. First observe that since $\tau \in [0, 1]$ and $\mu \leq L$ we have

$$\frac{\tau^2\mu}{2L(1+\tau^2)} \leq \frac{\mu}{3L} \leq \frac{1}{3}\,. \tag{67}$$

Then, combining (52) and (58), we can write

$$\begin{aligned} & 2[f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})] \geq \frac{1}{3}[f(\mathbf{x}_k) - f(\bar{\mathbf{x}}_k)] + \frac{\tau^2\mu}{2L(1+\tau^2)}\,[f(\bar{\mathbf{x}}_k) - f^*] \\ & \geq \frac{\tau^2\mu}{2L(1+\tau^2)}\,[f(\mathbf{x}_k) - f(\bar{\mathbf{x}}_k)] + \frac{\tau^2\mu}{2L(1+\tau^2)}\,[f(\bar{\mathbf{x}}_k) - f^*] \\ & = \frac{\tau^2\mu}{2L(1+\tau^2)}\,[f(\mathbf{x}_k) - f^*)]\,, \end{aligned} \tag{68}$$

and rearranging

$$f(\mathbf{x}_{k+1}) - f^* \leq q_P[f(\mathbf{x}_k) - f^*]\,. \tag{69}$$

The thesis follows for GS and random updates analogously. □

Proof of Theorem 1. We need to prove that the KL property (14) holds in $\{\bar{\mathbf{x}}_k\}$. The bounds on $f(\mathbf{x}_k) - f(\mathbf{x}_*)$ then follow immediately by induction from Lemma 9, and in turn the bounds on $\|\mathbf{x}_k - \mathbf{x}_*\|$ follow as in the proof of [34, Lemma 4.3].

For random updates, we can take $\tilde{\delta} < \delta$ small enough so that $f(\mathbf{x}_0) < f(\mathbf{x}_*)+\eta$. Then by construction the KL property (14) holds in U_0, and since $\{\bar{\mathbf{x}}_k\}$ is contained in U_0 by Lemma 5, (14) holds in particular in $\{\bar{\mathbf{x}}_k\}$.

For parallel updates, thanks to Lemma 5 we have that $\{f(\mathbf{x}_k)\}$ is decreasing and $f(\bar{\mathbf{x}}_k), f(\mathbf{x}_k) \geq f(\mathbf{x}_*)$. It can then be proved with an argument analogous to the proof of [34, Theorem 4.2] that for $\tilde{\delta}$ small enough, (14) holds in $\{\bar{\mathbf{x}}_k\}$. We include the argument here for completeness. Let $f_k = f(\mathbf{x}_k) - f(\mathbf{x}_*)$, and let $\tilde{\delta} < \delta/2$ defined as in the proof of [34, Theorem 4.2] so that

$$\tilde{\delta} < \frac{\delta}{2} < \delta - \frac{\sqrt{2f_0(1-q)}}{L(1-\sqrt{q})} - \sqrt{\frac{2}{L}}\sqrt{f_0}\,, \tag{70}$$

with $q = q_P$ here. We now want to prove $\bigcup_{[0:k-1]}\{\mathbf{x}_i, \bar{\mathbf{x}}_i\} \cup \{\mathbf{x}_k\} \subset B_\delta(\mathbf{x}_*)$ for every $k \in \mathbb{N}$, by induction on k. Notice that $\mathbf{x}_0 \in B_\delta(\mathbf{x}_*)$ by construction. To start with the inductive step,

$$\sum_{i=0}^{k-1} \|\mathbf{x}_i - \mathbf{x}_{i+1}\| \leq \sqrt{\frac{2}{L}} \sum_{i=0}^{k-1} \sqrt{f_i - f_{i+1}} \leq \frac{\sqrt{2f_0(1-q)}}{\sqrt{L}(1-\sqrt{q})} \tag{71}$$

where we used (44) in the first inequality, and the second can be derived from [34, Lemma 8.1] as in the proof of [34, Theorem 4.2]. But then

$$\begin{aligned}
\|\mathbf{x}_{k+1} - \mathbf{x}_*\| &\leq \|\mathbf{x}_0 - \mathbf{x}_*\| + \left(\sum_{i=0}^{k-1} \|\mathbf{x}_i - \mathbf{x}_{i+1}\|\right) + \|\mathbf{x}_k - \mathbf{x}_{k+1}\| \\
&\leq \tilde{\delta} + \frac{\sqrt{2f_0(1-q)}}{L(1-\sqrt{q})} + \sqrt{\frac{2}{L}}\sqrt{f_k - f_{k+1}} \\
&< \tilde{\delta} + \frac{\sqrt{2f_0(1-q)}}{L(1-\sqrt{q})} + \sqrt{\frac{2}{L}}\sqrt{f_k} < \delta\,,
\end{aligned} \tag{72}$$

where we used (71) together with (44) in the second inequality, $f_{k+1} = f(\mathbf{x}_{k+1}) - f(\mathbf{x}_*) \geq 0$ in the third inequality, and (70) together with $f_0 \geq f_k$ in the last inequality.

We now have

$$\begin{aligned}
\|\tilde{\mathbf{x}}_k - \mathbf{x}_*\| &\leq \|\mathbf{x}_0 - \mathbf{x}_*\| + \left(\sum_{i=0}^{k-1} \|\mathbf{x}_i - \mathbf{x}_{i+1}\|\right) + \|\mathbf{x}_k - \tilde{\mathbf{x}}_k\| \\
&\leq \|\mathbf{x}_0 - \mathbf{x}_*\| + \left(\sum_{i=0}^{k-1} \|\mathbf{x}_i - \mathbf{x}_{i+1}\|\right) + \|\mathbf{x}_k - \mathbf{x}_{k+1}\| < \delta\,,
\end{aligned} \tag{73}$$

where we used $\|\tilde{\mathbf{x}}_k - \mathbf{x}_k\| \leq \|\mathbf{x}_{k+1} - \mathbf{x}_k\|$ in the second inequality and the last inequality follows as in (72). Thus $\tilde{\mathbf{x}}_k \in B_\delta(\mathbf{x}_*)$ as well, and the induction is complete. For GS updates the proof that $\{\tilde{\mathbf{x}}_k\} \subset B_\delta(\mathbf{x}_*)$ is analogous. □

7.2 An active set identification criterion

We prove in this section Theorem 2, proposing a general active set identification criterion for Algorithm 1 in the special case where the feasible set $\mathcal{C}$ is the product of simplices. With the notation introduced in Section 4, let $\mathcal{C}_* = \{\mathbf{x} \in \mathcal{C} : \text{supp}(\mathbf{x}) = \text{supp}(\mathbf{x}_*)\}$ and $S_* = \{\mathbf{x} \in \mathbb{R}^n : \text{supp}(\mathbf{x}) = \text{supp}(\mathbf{x}_*)\}$ be the subset of points in $\mathcal{C}$ and the subspace of directions with the same support of $\mathbf{x}_*$ respectively.

Definition 4. *We say that the method $\bar{A}$ has active set related directions in $\mathbf{x}_*$ if it can do a bounded number of consecutive maximal steps, and if for some neighborhood V of $\mathbf{x}_*$, $\mathbf{x} \to \mathbf{x}_*$, $\mathbf{g} \to -\nabla f(\mathbf{x}_*)$ and $\mathbf{d} \in \bar{A}(\mathbf{x}, \mathbf{g})$:*

- *if $\mathbf{x} \in \mathcal{C}_*$ then $\mathbf{d} \in S_*$ with $\alpha_{\max}(\mathbf{x}, \hat{\mathbf{d}}) = \Theta(1)$,*
- *if $\mathbf{x} \in \mathcal{C} \setminus \mathcal{C}_*$ then $\langle \mathbf{g}, \hat{\mathbf{d}} \rangle = \Theta(1)$.*

Lemma 10. *Under the assumptions of Definition 4:*

- *if $\mathbf{x} \in \mathcal{C} \setminus \mathcal{C}_*$ we have $\alpha_{\max}(\mathbf{x}, \hat{\mathbf{d}}) = o(1)$,*
- *if $\mathbf{x} \in \mathcal{C}_*$, then $\langle \mathbf{g}, \hat{\mathbf{d}} \rangle = o(1)$.*

Proof. Notice that

$$\begin{aligned} 0 &\leq \alpha_{\max}(\mathbf{x}, \hat{\mathbf{d}}) \langle \mathbf{g}, \hat{\mathbf{d}} \rangle = \langle \mathbf{g}, (\mathbf{x} + \alpha_{\max}(\mathbf{x}, \hat{\mathbf{d}})\hat{\mathbf{d}}) - \mathbf{x} \rangle \\ &= \langle -\nabla f(\mathbf{x}), (\mathbf{x} + \alpha_{\max}(\mathbf{x}, \hat{\mathbf{d}})\hat{\mathbf{d}}) - \mathbf{x} \rangle + o(1) \leq \langle -\nabla f(\mathbf{x}), \mathbf{x}_* - \mathbf{x} \rangle + o(1) = o(1) \,. \end{aligned} \tag{74}$$

Thus

$$\alpha_{\max}(\mathbf{x}, \hat{\mathbf{d}}) \leq \frac{o(1)}{\langle \mathbf{g}, \hat{\mathbf{d}} \rangle} = o(1) \,, \tag{75}$$

where in the equality we used $\langle \mathbf{g}, \hat{\mathbf{d}} \rangle = \Theta(1)$ by assumption. This proves the first part of the claim. As for the second part, we have

$$\langle \mathbf{g}, \hat{\mathbf{d}} \rangle = \langle -\nabla f(\mathbf{x}_*), \hat{\mathbf{d}} \rangle + o(1) = o(1) \,, \tag{76}$$

where we used $\langle -\nabla f(\mathbf{x}_*), \hat{\mathbf{d}} \rangle = 0$ in the second equality, guaranteed by stationarity conditions since $\mathbf{d} \in S_*$. □

Proposition 11. *Let Algorithm 1 be applied to a method with active set related directions in $\mathbf{x}_*$ as in Definition 4. Then there is a neighborhood U of $\mathbf{x}_*$ such that if $\mathbf{x}_k \in U$ then supp$(\mathbf{x}_{k+1})$ = supp$(\mathbf{x}_*)$.*

Proof. Let $\{\mathbf{y}_i : i \in [0\!:\!j]\}$ be the set of points generated by SSC$(\mathbf{x}_k, -\nabla f(\mathbf{x}_k))$, $\bar{T}$ the upper bound on the number of consecutive maximal steps, so that $T \leq \bar{T} + 1$, and let $\bar{B}$ and B_j as in the proof of Lemma 4. We assume without loss of generality that $\|\mathbf{d}_j\| = 1$ for $j \in [0\!:\!T]$.
We will show that for $\mathbf{x}_k$ sufficiently close to $\mathbf{x}_*$ certain inequalities, namely (78), (79) and (82) are satisfied, allowing us to deduce the identification property. Let

$$T^* = \max\{j \in [0\!:\!T] : \{\mathbf{y}_i : i \in [0\!:\!j]\} \subset \mathcal{C} \setminus \mathcal{C}_*\}$$

whenever $\mathbf{y}_0 \notin \mathcal{C}_*$, and $T^* = -1$ otherwise. We first claim $\bar{T} \geq T^* + 1$. This is clear by the definition of T^* if $T^* = -1$. Otherwise, for $j \in [0\!:\!T^*]$ let $\tilde{\mathbf{y}}_{j+1} = \mathbf{y}_j + \alpha_{\max}^{(j)} \mathbf{d}_j$. We now show $\tilde{\mathbf{y}}_{j+1} \in \mathcal{C}_j$. First, we check $\tilde{\mathbf{y}}_{j+1} \in \text{int}(\bar{B})$. On one hand we have

$$\begin{aligned} L\|\tilde{\mathbf{y}}_{j+1} - \mathbf{y}_0\|^2 &= L\|\alpha_{\max}^{(j)} \mathbf{d}_j + \sum_{i=0}^{j-1} \alpha_i \mathbf{d}_i\|^2 \leq L \left(\sum_{i=0}^{j-1} \alpha_{\max}^{(i)} \|\mathbf{d}_i\| \right)^2 \\ &= L \left(\sum_{i=0}^{j-1} \alpha_{\max}^{(i)} \right)^2 = O(\max_{i \in [0:j]} (\alpha_{\max}^{(i)})^2) \,, \end{aligned} \tag{77}$$

where we used $\|\mathbf{d}_i\| = 1$ by assumption in the second equality. On the other hand

$$\langle \mathbf{g}, \tilde{\mathbf{y}}_{j+1} - \mathbf{y}_0 \rangle = \alpha_{\max}^{(j)} \mathbf{d}_j + \sum_{i=0}^{j-1} \alpha_i \langle \mathbf{g}_i, \mathbf{d}_i \rangle = O(\max_{i \in [0:j]} \alpha_{\max}^{(i)}) \, . \tag{78}$$

Since $\mathbf{y}_{(i)} \in \mathcal{C} \setminus \mathcal{C}_*$, by the active set related property $\alpha_{\max}^{(i)} = o(1)$ for $\mathbf{x}_k \to \mathbf{x}_*$. Let now M_1 and M_2 be the implicit constants in (77) and (78). For $\mathbf{y}_0 = \mathbf{x}_k$ close enough to $\mathbf{x}_*$ we obtain

$$L\|\tilde{\mathbf{y}}_{j+1} - \mathbf{y}_0\|^2 \leq M_1 \max_{i \in [0:j]} (\alpha_{\max}^{(i)})^2 < M_2 \max_{i \in [0:j]} \alpha_{\max}^{(i)} \leq \langle \mathbf{g}, \tilde{\mathbf{y}}_{j+1} - \mathbf{y}_0 \rangle \, , \tag{79}$$

where we used (77) in the first inequality, $\max_{i \in [0:j]} \alpha_{\max}^{(i)} = o(1)$ in the second inequality and (78) in the last inequality. From (79), $\tilde{\mathbf{y}}_{j+1} \in \mathrm{int}\bar{B}$ follows easily as desired.
We now need to check $\tilde{\mathbf{y}}_{j+1} \in B_j$. Reasoning as above, on the one hand we have $\frac{\|\mathbf{g}\|}{2L} = \Theta(1)$ for $\mathbf{x}_k \to \mathbf{x}$ (setting aside the trivial case where $-\nabla f(\mathbf{x}_*) = 0$), and on the other hand $\|\tilde{\mathbf{y}}_{j+1} - \mathbf{y}_0\| = o(1)$ by (77), so that

$$\|\tilde{\mathbf{y}}_{j+1} - \mathbf{y}_0\| < \frac{\|\mathbf{g}\|}{2L} \tag{80}$$

for $\mathbf{y}_0$ close enough to $\mathbf{x}_*$ and $\tilde{\mathbf{y}}_{j+1} \in B_j$ as desired. Then $\tilde{\mathbf{y}}_{j+1} \in \mathrm{int}(\mathcal{C}_j)$ for $j \in [0 : T^*]$, or equivalently $\beta_j > \alpha_{\max}^{(j)}$ the SSC does always maximal steps in the first $T^* + 1$ iterations. In particular, it generates the point $\mathbf{y}_{T^*+1} \in \mathcal{C}_* \setminus \{\mathbf{y}_{T^*}\}$. The claim is thus proved.
If $\mathbf{y}_{T^*+1}$ is stationary for g, the SSC terminates at step 4 with output $\mathbf{y}_{T^*+1} \in \mathcal{C}_*$ and the thesis is proved. Otherwise, we claim that the SSC terminates with output $\mathbf{y}_{T^*+2} \in \mathcal{C}_*$ and $\beta_{T^*+1} < \alpha_{\max}^{(T^*+1)}$. First, observe that by assumption we must have $\mathbf{d}_{T^*+1} \in S_*$, and therefore $\mathbf{y}_{T^*+2} = \mathbf{y}_{T^*+1} + \alpha_{T^*+1}\mathbf{d}_{T^*+1} \in \mathcal{C}_*$. Second, we have $\alpha_{\max}^{(T^*+1)} = \Theta(1)$, and at the same time

$$\beta_{T^*+1} \leq \mathrm{diam}(\mathcal{C}_{T^*+1}) \leq \mathrm{diam}(B_{T^*+1}) \leq 2\langle \mathbf{g}_{T^*+1}, \mathbf{d}_{T^*+1} \rangle = o(1) \, . \tag{81}$$

Thus for $\mathbf{y}_0$ close enough to $\mathbf{x}_*$ we must have

$$\beta_{T^*+1} < \alpha_{\max}^{(T^*+1)} \, , \tag{82}$$

and the claim is proved. Since the SSC terminates either with $\mathbf{y}_{T^*+1}$ or $\mathbf{y}_{T^*+2}$, and both of these points are in $\mathcal{C}_*$, the thesis follows. □

Lemma 12. *For $\mathbf{x} \to \mathbf{x}_*$, $\mathbf{g} \to -\nabla f(\mathbf{x}_*)$, if $\mathbf{x} \in \mathcal{C}_*$, $\mathbf{d} \in S_*$ and $\alpha_{\max}(\mathbf{x}, \hat{\mathbf{d}})$ coincides with the maximal feasible stepsize, then $\alpha_{\max}(\mathbf{x}, \hat{\mathbf{d}}) = \Theta(1)$.*

Proof. We have

$$\alpha_{\max}(\mathbf{x}, \hat{\mathbf{d}}) = \min_{i : \hat{d}_i < 0} \frac{x_i}{|\hat{d}_i|} \geq \min_{i : \hat{d}_i < 0} x_i \geq \min_{i \in \mathrm{supp}(\mathbf{x}_*)} x_i = \Theta(1) \, , \tag{83}$$

where we used $|\hat{d}_i| \leq \|\hat{\mathbf{d}}\| \leq 1$ in the first inequality, $\mathrm{supp}(\mathbf{d}) \subseteq \mathrm{supp}(\mathbf{x}_*)$ in the second inequality, and $x_i \to x_{*,i} > 0$ in the third one. □

For $\mathbf{x} \in \mathcal{C}$, we define the expression

$$\lambda^{(i)}(\mathbf{x}, \mathbf{g}) = \langle \mathbf{g}^{(i)}, \mathbf{x}^{(i)} \rangle \, \mathbf{e}^{(i)} - \mathbf{g}^{(i)} \, , \quad i \in [1 \!:\! m] \, ,$$

and the Lagrangian multiplier vector

$$\lambda^{(i)}(\mathbf{x}) = \lambda^{(i)}(\mathbf{x}, -\nabla f(\mathbf{x})) = \nabla f(\mathbf{x})^{(i)} - \langle \nabla f(\mathbf{x})^{(i)}, \mathbf{x}^{(i)} \rangle \, \mathbf{e}^{(i)} \, , \quad i \in [1 \!:\! m] \, . \tag{84}$$

We notice that strict complementarity holds at a stationary point $\mathbf{x}_* \in \mathcal{C}$ for $\nabla f(\mathbf{x}_*)$ if and only if it holds for every $i \in [1 \!:\! m]$ at $\mathbf{x}_*^{(i)} \in \mathcal{C}^{(i)}$ and $\nabla f(\mathbf{x}_*)^{(i)}$.

Lemma 13. *Assume that strict complementarity holds at $\mathbf{x}_*$. Then the AFW applied to the simplex has active set related directions in $\mathbf{x}_*$ as in Definition 4.*

Proof. For $\mathbf{x} \to \mathbf{x}_*$ and $\mathbf{g} \to -\nabla f(\mathbf{x}_*)$ we have $\lambda(\mathbf{x}, \mathbf{g}) \to \lambda(\mathbf{x}_*)$, and therefore in particular $\lambda_i(\mathbf{x}, \mathbf{g}) \to 0$ for $i \in \operatorname{supp}(\mathbf{x}_*)$ while $\lambda_i(\mathbf{x}, \mathbf{g}) \to \lambda_i(\mathbf{x}_*) > 0$ for $i \in [1:n] \setminus \operatorname{supp}(\mathbf{x}_*)$. Therefore, for $\mathbf{x} \in \mathcal{C} \backslash \mathcal{C}_*$ close enough to $\mathbf{x}_*$ we must have $\max\{\lambda_i(\mathbf{x}, \mathbf{g}) : i \in \operatorname{supp}(\mathbf{x})\} > \max\{-\lambda_i(\mathbf{x}, \mathbf{g}) : i \in [1:n]\}$, so that by [12, Lemma 3.2(a)] we have that the descent direction selected by the AFW satisfies $\mathbf{d} = \mathbf{x} - \mathbf{e}_{\hat{i}}$ for some $\hat{i} \in \operatorname{argmax}\{\lambda_i(\mathbf{x}, \mathbf{g}) : i \in \operatorname{supp}(\mathbf{x})\} \subset [1:n] \setminus \operatorname{supp}(\mathbf{x}_*)$. Therefore

$$\langle \mathbf{g}, \hat{\mathbf{d}} \rangle = \langle \mathbf{g}, \frac{\mathbf{x} - \mathbf{e}_{\hat{i}}}{\|\mathbf{x} - \mathbf{e}_{\hat{i}}\|} \rangle = \frac{\lambda_{\hat{i}}(\mathbf{x}, \mathbf{g})}{\|\mathbf{x} - \mathbf{e}_{\hat{i}}\|} = \Theta(1) \tag{85}$$

for $\mathbf{x} \to \mathbf{x}_*$ and $\mathbf{g} \to -\nabla f(\mathbf{x}_*)$.
As for the case $\mathbf{x} \in \mathcal{C}_*$, then if $\mathbf{x}, \mathbf{g}$ are close enough to $\mathbf{x}_*$ we must have $\lambda_i(\mathbf{x}, \mathbf{g}) > 0$ for every i in $[1:n] \setminus \operatorname{supp}(\mathbf{x}_*)$. Therefore by [12, Lemma 3.2(b)] if y is obtained from $\mathbf{x}$ with a FW update we must have $\mathbf{y}_i = 0$ for $i \in [1:n] \setminus \operatorname{supp}(\mathbf{x}_*)$, which is equivalent to say that the update direction must be in S_*. The property $\alpha_{\max}(\mathbf{x}, \hat{\mathbf{d}}) = \Theta(1)$ follows by Lemma 12. □

Proof of Theorem 2. Follows by applying the property proved in Lemma 13 to each block selected by the method. □

References

[1] Carlos M Alaíz, Alvaro Barbero, and José R Dorronsoro. Group fused lasso. In *International Conference on Artificial Neural Networks*, pages 66–73. Springer, 2013.

[2] Noga Alon and Joel H Spencer. *The probabilistic method.* John Wiley & Sons, 2016.

[3] Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

[4] Amir Beck. *First-order methods in optimization.* SIAM, 2017.

[5] Dimitri Bertsekas and John Tsitsiklis. *Parallel and distributed computation: numerical methods.* Athena Scientific, 2015.

[6] Jérôme Bolte, Aris Daniilidis, Adrian Lewis, and Masahiro Shiota. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18(2):556–572, 2007.

[7] Jérôme Bolte, Aris Daniilidis, Olivier Ley, and Laurent Mazet. Characterizations of Łojasiewicz inequalities: subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010.

[8] Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507, 2017.

[9] Immanuel M Bomze, Marco Budinich, Panos M Pardalos, and Marcello Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization*, pages 1–74. Springer, 1999.

[10] Immanuel M Bomze, Markus Gabl, Francesca Maggioni, and Georg Pflug. Two-stage stochastic standard quadratic optimization. *European Journal of Operational Research*, 299(1):21–34, 2022.

[11] Immanuel M Bomze, Francesco Rinaldi, and Samuel Rota Bulo. First-order methods for the impatient: Support identification in finite time with convergent Frank-Wolfe variants. *SIAM Journal on Optimization*, 29(3):2211–2226, 2019.

[12] Immanuel M Bomze, Francesco Rinaldi, and Damiano Zeffiro. Active set complexity of the away-step Frank–Wolfe algorithm. *SIAM Journal on Optimization*, 30(3):2470–2500, 2020.

[13] Immanuel M Bomze, Francesco Rinaldi, and Damiano Zeffiro. Frank-Wolfe and friends: a journey into projection-free first-order optimization methods. *4OR*, 19(3):313–345, 2021.

[14] Immanuel M Bomze, Francesco Rinaldi, and Damiano Zeffiro. Fast cluster detection in networks by first order optimization. *SIAM Journal on Mathematics of Data Science*, 4(1):285–305, 2022.

[15] Immanuel M Bomze and Werner Schachinger. Multi-standard quadratic optimization: interior point methods and cone programming reformulation. *Computational Optimization and Applications*, 45(2):237–256, 2010.

[16] Immanuel M. Bomze, Werner Schachinger, and Reinhard Ullrich. The complexity of simple models – a study of worst and typical hard cases for the standard quadratic optimization problem. *Mathematics of Operations Research*, 43(2):347–692, 2017.

[17] Nicolas Boumal. An introduction to optimization on smooth manifolds. *Available online, May*, 3, 2020.

[18] Cyrille W Combettes and Sebastian Pokutta. Complexity of linear minimization and projection on some sets. *Operations Research Letters*, 49(4):565–571, 2021.

[19] Rina Foygel, Michael Horrell, Mathias Drton, and John Lafferty. Nonparametric reduced rank regression. *Advances in Neural Information Processing Systems*, 25, 2012.

[20] Takumi Fukunaga and Hiroyuki Kasai. Fast block-coordinate Frank-Wolfe algorithm for semi-relaxed optimal transport. *arXiv preprint arXiv:2103.05857*, 2021.

[21] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435, 2013.

[22] Stefanie Jegelka, Francis Bach, and Suvrit Sra. Reflection methods for user-friendly submodular optimization. *Advances in Neural Information Processing Systems*, 26, 2013.

[23] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

[24] Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. Block-coordinate Frank-Wolfe optimization for structural SVMs. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 53–61, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[25] Guanghui Lan. *First-order and Stochastic Optimization Methods for Machine Learning.* Springer, 2020.

[26] Robert H Leary. Global optimization on funneling landscapes. *Journal of Global Optimization*, 18(4):367–383, 2000.

[27] Larry J LeBlanc, Edward K Morlok, and William P Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation research*, 9(5):309–318, 1975.

[28] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2012.

[29] Marco Locatelli and Fabio Schoen. *Global optimization: theory, algorithms, and applications.* SIAM, 2013.

[30] Zhi-Quan Luo and Paul Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.

[31] Anton Osokin, Jean-Baptiste Alayrac, Isabella Lukasewitz, Puneet Dokania, and Simon Lacoste-Julien. Minding the gaps for block Frank-Wolfe optimization of structured svms. In *International Conference on Machine Learning*, pages 593–602. PMLR, 2016.

[32] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1):1–38, 2014.

[33] Francesco Rinaldi and Damiano Zeffiro. A unifying framework for the analysis of projection-free first-order methods under a sufficient slope condition. *arXiv preprint arXiv:2008.09781*, 2020.

[34] Francesco Rinaldi and Damiano Zeffiro. Avoiding bad steps in Frank Wolfe variants. *Computational Optimization and Applications*, 84:225–264, 2023.

[35] Yu-Xiang Wang, Veeranjaneyulu Sadhanala, Wei Dai, Willie Neiswanger, Suvrit Sra, and Eric Xing. Parallel and distributed block-coordinate Frank-Wolfe algorithms. In *International Conference on Machine Learning*, pages 1548–1557. PMLR, 2016.