The Pivoting Framework: Frank-Wolfe Algorithms with Active Set Size Control

Elias Wirth
TU Berlin

Mathieu Besançon
Université Grenoble Alpes
Inria, CNRS, LIG

Sebastian Pokutta
TU Berlin
Zuse Institute Berlin

Abstract

We propose the pivoting meta algorithm (PM) to enhance optimization algorithms that generate iterates as convex combinations of vertices of a feasible region $\mathcal{C} \subseteq \mathbb{R}^n$, including Frank-Wolfe (FW) variants. PM guarantees that the active set (the set of vertices in the convex combination) of the modified algorithm remains as small as $\dim(\mathcal{C}) + 1$ as stipulated by Carathéodory's theorem. PM achieves this by reformulating the active set expansion task into an equivalent linear program, which can be efficiently solved using a single pivot step akin to the primal simplex algorithm; the convergence rate of the original algorithms are maintained. Furthermore, we establish the connection between PM and active set identification, in particular showing under mild assumptions that PM applied to the away-step Frank-Wolfe algorithm (AFW) or the blended pairwise Frank-Wolfe algorithm (BPFW) bounds the active set size by the dimension of the optimal face plus 1. We provide numerical experiments to illustrate practicality and efficacy on active set size reduction.

1 Introduction

We study constrained convex optimization problems

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

where $\mathcal{C} \subseteq \mathbb{R}^n$ is a compact convex set with vertex set $\mathcal{V} = \operatorname{vert}(\mathcal{C})$ and $f\colon \mathcal{C} \to \mathbb{R}$ is a convex and smooth function. When projecting onto $\mathcal{C}$ is computationally challenging, we can address (OPT) using the projection-free *Frank-Wolfe algorithm* (FW) (Frank and Wolfe, 1956), a.k.a. the *conditional gradients algorithm* (Levitin and Polyak, 1966). The FW algorithm, presented in Algorithm 1, only requires first-order access to the function f and a *linear minimization oracle* (LMO) for the feasible region $\mathcal{C}$. The LMO returns a point in $\operatorname{argmin}_{\mathbf{x}\in\mathcal{C}}\langle \mathbf{c}, \mathbf{x}\rangle$ when given $\mathbf{c} \in \mathbb{R}^n$. FW possesses favorable attributes such as ease of implementation, affine invariance (Lacoste-Julien and Jaggi, 2013; Kerdreux et al., 2021; Peña, 2023), and its iterates are sparse convex combinations of vertices of $\mathcal{C}$. At each iteration, the FW algorithm calls the LMO to obtain a new FW vertex $\mathbf{v}^{(t)} \in \mathcal{V}$. As presented in Algorithm 1, the current iterate $\mathbf{x}^{(t)}$ is updated with line-search $\eta^{(t)} = \operatorname{argmin}_{\eta\in[0,1]} f(\mathbf{x}^{(t)} + \eta(\mathbf{v}^{(t)} - \mathbf{x}^{(t)}))$ in Line 4 to obtain $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \eta^{(t)}(\mathbf{v}^{(t)} - \mathbf{x}^{(t)})$. Alternative step-size rules exist, including short step, adaptive (Pedregosa et al., 2018; Pokutta, 2024), and open-loop variants $\eta^{(t)} = \frac{\ell}{t+\ell}$ for some $\ell \in \mathbb{N}_{>0}$ (Dunn and Harshbarger, 1978; Wirth et al., 2023b,c).

One drawback of the vanilla FW algorithm is its sublinear convergence rate when potentially higher rates are possible, e.g., when the feasible region is a polytope, the objective is strongly convex, and the optimizer lies in the relative interior of a face of $\mathcal{C}$ (see e.g.,Wolfe (1970); Bach (2021); Wirth et al. (2023b)). Consequently, several variants have been proposed to achieve linear convergence rates in such scenarios (see e.g., Holloway (1974); Guélat and Marcotte (1986); Lacoste-Julien and Jaggi (2015); Garber and Meshi (2016); Tsuji et al. (2022)). Most of these variants store the current iterate $\mathbf{x}^{(t)} = \sum_{\mathbf{s}\in\mathcal{S}^{(t)}} \alpha_{\mathbf{s}}^{(t)}\mathbf{s}$ as a convex combination of vertices, where $\boldsymbol{\alpha}^{(t)} \in \Delta_{|\mathcal{V}|}$, and $\mathcal{S}^{(t)} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(t)} > 0\}$ denote the *weight vector* and *active set*, respectively. Explicit access to a convex decomposition of $\mathbf{x}^{(t)}$ enables re-optimization over the active set or aggressive removal of weight from specific vertices and is crucial for achieving linear convergence rates. Moreover, variants often enhance the sparsity-inducing properties of vanilla FW, which is advantageous in, e.g., deriving bounds for the approximate Carathéodory theorem

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

(Combettes and Pokutta, 2023), approximate vanishing ideal computations (Wirth and Pokutta, 2022; Wirth et al., 2023a), data-driven identification of nonlinear dynamics (Carderera et al., 2021), deep neural network training (Pokutta et al., 2020; Macdonald et al., 2022), kernel herding (Bach et al., 2012; Tsuji et al., 2022; Wirth et al., 2023b), robust matrix recovery (Mu et al., 2016), and tensor completion (Guo et al., 2017; Bugg et al., 2022).

Maintaining the active set introduces computational overhead, especially in high-dimensional or dense vertex scenarios due to memory constraints. To improve efficiency, several methods aim to reduce the active set size. One approach alternates between adding vertices and performing correction steps, either fully or partially re-optimizing the set. Examples include the *fully-corrective Frank-Wolfe algorithm* (Holloway, 1974; Rao et al., 2015) and the *Blended Frank-Wolfe algorithm (BFW)* (Braun et al., 2019), also known as *Blended Conditional Gradients (BCG)*. Another approach uses *drop steps* to prune vertices from the active set. Key algorithms include the *away-step Frank-Wolfe algorithm* (AFW) (Wolfe, 1970; Guélat and Marcotte, 1986; Lacoste-Julien and Jaggi, 2015), detailed in Algorithm 6, the *pairwise Frank-Wolfe algorithm* (Lacoste-Julien and Jaggi, 2015), the *decomposition-invariant Frank-Wolfe algorithm* (Garber and Meshi, 2016), and the *Blended Pairwise Frank-Wolfe algorithm (BPFW)* (Tsuji et al., 2022), outlined in Algorithm 7, also called *Blended Pairwise Conditional Gradients (BPCG)*.

In FW variants, the size of the active set is typically only bounded solely by the number of iterations performed and the number of vertices in the feasible region, that is, $|\mathcal{S}^{(t)}| \leq \min\{t+1, |\mathcal{V}|\}$. Notably, this bound does not depend on the ambient dimension n. This observation is somewhat surprising considering the well-known *Carathéodory theorem*, which guarantees that any vector $\mathbf{x} \in \mathcal{C}$ can always be expressed as a convex combination of at most $n+1$ vertices of $\mathcal{C}$. To the best of our knowledge, the only approach proposed to bound the number of vertices to match that of the Carathéodory theorem is the *incremental representation reduction algorithm* (IRR) algorithm of Beck and Shtern (2017), to which we compare our approach in Subsection 4.1.

Theorem 1.1 (Carathéodory, 1907)**.** *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a compact convex set. Then, any $\mathbf{x} \in \mathcal{C}$ can be represented as a convex combination of at most $n+1$ vertices of $\mathcal{C}$.*

In settings where the number of vertices is significantly larger than the dimension, such as, for example, in the convex hull membership problem (Filippozzi et al., 2023), a dimension-dependent upper bound on the size of the active set is preferable to a bound based solely on the number of vertices of $\mathcal{C}$.

Algorithm 1: Frank-Wolfe algorithm (FW) with line-search

Input: $\mathbf{x}^{(0)} \in \mathcal{V}$.
Output: $\mathbf{x}^{(T)} \in \mathcal{C}$.

for $t = 0, 1, \ldots, T-1$ **do**
  $\mathbf{v}^{(t)} \leftarrow \operatorname{argmin}_{\mathbf{v} \in \mathcal{V}} \langle \nabla f(\mathbf{x}^{(t)}), \mathbf{v} - \mathbf{x}^{(t)} \rangle$
  $\eta^{(t)} \leftarrow \operatorname{argmin}_{\eta \in [0,1]} f(\mathbf{x}^{(t)} + \eta(\mathbf{v}^{(t)} - \mathbf{x}^{(t)}))$
  $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} + \eta^{(t)}(\mathbf{v}^{(t)} - \mathbf{x}^{(t)})$
end

1.1 Contributions

In this paper, we address the existing gap in the literature by introducing the *pivoting meta algorithm* (PM) presented in Algorithm 4. Our contributions can be summarized as follows:

Active-Set Reduction First, PM is designed to enhance a family of optimization algorithms, including various existing variants of the Frank-Wolfe algorithm (FW). Our main result, Theorem 4.2, demonstrates the key advantage of PM: PM applied to certain optimization algorithms ensures that the cardinality of the active set remains bounded by $n+1$ while preserving the convergence rate guarantees of the original algorithm. To achieve this, PM transforms the task of adding a new vertex $\mathbf{v}^{(t)} \in \mathcal{V} \setminus \mathcal{S}^{(t)}$ to an active set $\mathcal{S}^{(t)}$ into an equivalent linear programming problem. This problem can be solved using a single pivot step, similar to the primal simplex algorithm (Bertsimas and Tsitsiklis, 1997). However, this modification introduces the additional computational complexity of solving an $(n+2) \times (n+2)$ linear system in iterations performing an FW step, typically via an LU-decomposition. We highlight however, how this computational burden is alleviated by the fact that only sparse rank-one updates are performed on the system to solve, making it amenable to efficient factorization updates as performed in modern simplex solvers.

Active set identification Second, we establish a connection between PM and *active set identification*, the process of identifying the face containing a solution of (OPT) and not to be confused with the active set of FW algorithms. When the feasible region is a polytope and the minimizers lie in the relative interior of a face $\mathcal{C}^*$ of $\mathcal{C}$, Bomze et al. (2020) provided sufficient conditions that guarantee the existence of an iteration $R \in \{0, 1, \ldots, T\}$, where T is the number of iterations AFW is run for, s.t. for all $t \in \{R, R+1, \ldots, T\}$, the active set $\mathcal{S}^{(t)}$ produced by AFW is contained in the optimal face $\mathcal{C}^*$, implying that $\mathbf{x}^{(t)} \in \mathcal{C}^*$. In the same setting, we prove that applying PM to AFW guarantees an

active set size of at most $\dim(\mathcal{C}^*)+1$ for all iterations $t \in \{R, R+1, \dots, T\}$.[1] Since our result holds for a more general setting, we further improve the upper bound on the size of the active set of $n+1$ to $\dim(\mathcal{C})+1$ for any compact and convex feasible region $\mathcal{C}$.

Numerical Experiments Finally, we provide an algorithmic implementation by applying `PM` to `AFW` and the blended pairwise Frank-Wolfe algorithm (`BPFW`) and comparing the method to `AFW` and `BPFW`.

Some numerical experiments, proofs, and the discussion of implementation details have been relegated to the supplementary material due to space constraints.

2 Preliminaries

For $n \in \mathbb{N}$, let $[n] := \{1, \dots, n\}$. Vectors are denoted in bold. Given $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, let $\mathbf{x} \geq \mathbf{y}$ denote that $x_i \geq y_i$ for all $i \in [n]$. Given $\mathbf{x} \in \mathbb{R}^n$, let $\tilde{\mathbf{x}} \in \mathbb{R}^{n+2}$ be defined as $(\mathbf{x}^\intercal, 0, 1)^\intercal \in \mathbb{R}^{n+2}$. We denote the ith unit vector of dimension n by $\mathbf{e}_i \in \mathbb{R}^n$. Given a vector $\mathbf{x} \in \mathbb{R}^n$, denote its support by $\operatorname{supp}(\mathbf{x}) := \{i \in [n] \mid x_i \neq 0\}$. Given a matrix $M \in \mathbb{R}^{m \times n}$, we refer to the (i,j)th entry of M as $M_{i,j}$ and the ith row and column of M as $M_{i,:}$ and $M_{:,i}$, respectively. Furthermore, let $M_{:i,:j}$ denote the restriction of matrix M to rows $1, \dots, i$ and columns $1, \dots, j$. We define the matrix

$$D_n := \begin{pmatrix} I_n & \mathbf{0} \\ \mathbf{1}^\intercal & 1 \\ \mathbf{1}^\intercal & 1 \end{pmatrix} \in \mathbb{R}^{(n+2)\times(n+1)},$$

where I_n is the n-dimensional identity matrix, $\mathbf{0}$ and $\mathbf{1}$ the all-zero and all-one vectors. Let $\Delta_n = \{\mathbf{x} \in \mathbb{R}^n_{\geq 0} \mid \|\mathbf{x}\|_1 = 1\}$ denote the probability simplex. Throughout, let $\mathcal{C} \subseteq \mathbb{R}^n$ be a nonempty compact convex set and let $\operatorname{aff}(\mathcal{C})$, $\dim(\mathcal{C})$, and $\mathcal{V} = \operatorname{vert}(\mathcal{C})$ denote the affine hull, dimension, and set of vertices of $\mathcal{C}$, respectively. If $\mathcal{C}$ is a polytope, let $\operatorname{faces}(\mathcal{C})$ denote the sets of faces of $\mathcal{C}$. For a set $F \subseteq \mathbb{R}^n$, $\operatorname{conv}(F)$ and $\operatorname{rel.int}(F)$ are the convex hull and relative interior of F, respectively. For $F, G \subseteq \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^n$, $\operatorname{dist}(F, G) = \inf_{\mathbf{y}\in F, \mathbf{z}\in G} \|\mathbf{y}-\mathbf{z}\|_2$ is the Euclidean distance between F and G and $\operatorname{dist}(\mathbf{x}, F) = \inf_{\mathbf{y}\in F} \|\mathbf{y}-\mathbf{x}\|_2$ is the Euclidean point-set distance between $\mathbf{x}$ and F. A continuously differentiable function $f\colon \mathcal{C} \to \mathbb{R}$ is *L-smooth* over $\mathcal{C}$ with $L > 0$ if $f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \; \forall \mathbf{x}, \mathbf{y} \in \mathcal{C}$. The function is *$\mu$-strongly convex* with $\mu > 0$ if $f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \; \forall \mathbf{x}, \mathbf{y} \in \mathcal{C}$.

The pyramidal width (Lacoste-Julien and Jaggi, 2015) is equivalent to the definition below by Pena and Rodriguez (2019).

[1] The result by Bomze et al. (2020) is not limited to `AFW`. In Appendix 9, we derive a similar result for `BPFW`.

Algorithm 2: Carathéodory-amenable algorithm (`CA`) [Template]

Input: $\mathbf{x}^{(0)} \in \mathcal{V}$.
Output: $\boldsymbol{\alpha}^{(T)} \in \Delta_{|\mathcal{V}|}$,
$\mathcal{S}^{(T)} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(T)} > 0\}$, and
$\mathbf{x}^{(T)} \in \mathcal{C}$, such that
$\mathbf{x}^{(T)} = \sum_{\mathbf{s}\in\mathcal{S}^{(T)}} \alpha_{\mathbf{s}}^{(T)}\mathbf{s}$.

$\boldsymbol{\alpha}^{(0)} \in \Delta_{|\mathcal{V}|}$ s.t. for all $\mathbf{s} \in \mathcal{V}$,
$\alpha_{\mathbf{s}}^{(0)} \leftarrow \begin{cases} 1, & \text{if } \mathbf{s} = \mathbf{x}^{(0)} \\ 0, & \text{if } \mathbf{s} \in \mathcal{V} \setminus \{\mathbf{x}^{(0)}\} \end{cases}$
$\mathcal{S}^{(0)} \leftarrow \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(0)} > 0\} = \{\mathbf{x}^{(0)}\}$
for $t = 0, 1, \dots, T-1$ **do**
  $(\boldsymbol{\alpha}^{(t+1)}, \mathcal{S}^{(t+1)}, \mathbf{x}^{(t+1)}) \leftarrow$
  `CCU`$(\boldsymbol{\alpha}^{(t)}, \mathcal{S}^{(t)}, \mathbf{x}^{(t)})$ ▷ see Algorithm 3
end

Definition 2.1 (Pyramidal width)**.** Let $\emptyset \neq \mathcal{C} \subseteq \mathbb{R}^n$ be a polytope with vertex set $\mathcal{V}$. The *pyramidal width* of $\mathcal{C}$ is $\omega := \min_{F\in\operatorname{faces}(\mathcal{C}), \emptyset\subsetneq F\subsetneq\mathcal{C}} \operatorname{dist}(F, \operatorname{conv}(\mathcal{V}\setminus F))$.

3 Amenable algorithms

We will begin by introducing two key concepts: a) *Carathéodory-amenable algorithms* (`CA`), outlined in Algorithm 2, which are well-suited for use with the pivoting meta-algorithm (`PM`); and b) *convex-combination-agnostic* properties, which are inherent characteristics of `CA`s that are preserved by the `PM`.

In particular, we demonstrate that `FW` algorithms such as vanilla `FW`, Algorithm 1, the away-step Frank-Wolfe algorithm (`AFW`), Algorithm 6, and the blended pairwise Frank-Wolfe algorithm (`BPFW`), Algorithm 7, are `CA`s. Furthermore, we prove that most convergence rates for `FW`, `AFW`, and `BPFW` are convex-combination-agnostic. Thus, `FW`, `AFW`, or `BPFW` modified with `PM` regularly enjoy the same convergence rates as the original algorithms.

3.1 Carathéodory-amenable algorithms

Consider an algorithm that represents each iterate $\mathbf{x}^{(t)}$ as a convex combination, i.e., $\mathbf{x}^{(t)} = \sum_{\mathbf{s}\in\mathcal{S}^{(t)}} \alpha_{\mathbf{s}}^{(t)}\mathbf{s}$. Here, $\boldsymbol{\alpha}^{(t)} \in \Delta_{|\mathcal{V}|}$ and $\mathcal{S}^{(t)} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(t)} > 0\}$ denote the *weight vector* and *active set* at iteration $t \in \{0, 1, \dots, T\}$, respectively. Most `FW` variants then perform one of the following types of updates:

`FW` **update:** The algorithm shifts weight from all vertices in the active set to a single vertex, that is, $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \eta^{(t)}(\mathbf{v}^{(t)} - \mathbf{x}^{(t)})$, where $\mathbf{v}^{(t)} \in \mathcal{V}$ and $\eta^{(t)} \in [0, 1]$. See, e.g., Line 4 in `FW` (Algorithm 1).

Algorithm 3: Convex-combination update (CCU), $(\boldsymbol{\beta}, \mathcal{T}, \mathbf{y}) = \texttt{CCU}(\boldsymbol{\alpha}, \mathcal{S}, \mathbf{x})$

Input: $\boldsymbol{\alpha} \in \Delta_{|\mathcal{V}|}$, $\mathcal{S} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}} > 0\}$, and $\mathbf{x} \in \mathcal{S}$, such that $\mathbf{x} = \sum_{\mathbf{s} \in \mathcal{S}} \alpha_{\mathbf{s}} \mathbf{s}$.

Output: $\boldsymbol{\beta} \in \Delta_{|\mathcal{V}|}$, $\mathcal{T} = \{\mathbf{s} \in \mathcal{V} \mid \beta_{\mathbf{s}} > 0\}$, and $\mathbf{y} \in \mathcal{C}$, such that $\mathbf{y} = \sum_{\mathbf{s} \in \mathcal{T}} \beta_{\mathbf{s}} \mathbf{s}$ and either $\mathcal{T} \subseteq \mathcal{S}$ or there exists exactly one $\mathbf{v} \in \mathcal{T} \setminus \mathcal{S}$ such that $\mathcal{T} \subseteq \mathcal{S} \cup \{\mathbf{v}\}$.

Away update: The algorithm shifts weight from a single vertex in the active set to all other vertices in the active set. That is, $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \eta^{(t)}(\mathbf{x}^{(t)} - \mathbf{a}^{(t)})$, where $\mathbf{a}^{(t)} \in \mathcal{S}^{(t)}$ and $\eta^{(t)} \in [0, \alpha^{(t)}_{\mathbf{a}^{(t)}}/(1 - \alpha^{(t)}_{\mathbf{a}^{(t)}})]$. See, e.g., Line 7 in AFW (Algorithm 6).

Pairwise update: Assuming that $\eta^{(t)} \neq 1$, the algorithm performs both an away and a FW update simultaneously, that is, $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \eta^{(t)}(\mathbf{v}^{(t)} - \mathbf{a}^{(t)})$, where $\mathbf{v}^{(t)} \in \mathcal{V}$, $\mathbf{a}^{(t)} \in \mathcal{S}^{(t)}$, $\mathbf{v}^{(t)} \neq \mathbf{a}^{(t)}$, and $\eta^{(t)} \in [0, \alpha^{(t)}_{\mathbf{a}^{(t)}}]$. See, e.g., Line 8 in BPFW (Algorithm 7). To see that a pairwise update is equivalent to an away update followed by an FW update, consider the auxiliary vector $\mathbf{y}^{(t)} = \mathbf{x}^{(t)} + \frac{\eta^{(t)}}{1-\eta^{(t)}}(\mathbf{x}^{(t)} - \mathbf{a}^{(t)})$ obtained after performing an away update with step-size $\frac{\eta^{(t)}}{1-\eta^{(t)}}$. Then, as required:

$$\begin{aligned}
\mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} + \eta^{(t)}(\mathbf{v}^{(t)} - \mathbf{a}^{(t)}) \\
&= \mathbf{x}^{(t)}(1 - \eta^{(t)} + \frac{\eta^{(t)}}{1 - \eta^{(t)}}(1 - \eta^{(t)})) \\
&\quad + \eta^{(t)}\mathbf{v}^{(t)} - \frac{\eta^{(t)}}{1 - \eta^{(t)}}(1 - \eta^{(t)})\mathbf{a}^{(t)} \\
&= \mathbf{y}^{(t)} + \eta^{(t)}(\mathbf{v}^{(t)} - \mathbf{y}^{(t)}).
\end{aligned}$$

These updates are captured by the more general *convex-combination updates* (CCUs) formalized in Algorithm 3. We refer to algorithms that perform repeated CCUs as Carathéodory-amenable algorithms (CAs), see Algorithm 2. Several comments are warranted. First, the running examples of this paper, FW, AFW, and BPFW, are CAs. Second, despite the focus of this work on FW algorithms, the class of CAs is general enough to potentially capture other methods, such as simplicial decompositions (Bettiol et al., 2024) or the nearest-extreme point variant (Garber and Wolf, 2021). We leave the applicability of our framework for algorithms forming iterates as linear or conic combinations (such as matching pursuit) to future research. Finally, we highlight that a CCU does not require $\mathbf{v}$ to be obtained via an (exact) LMO. Thus, the lazified variants of FW, AFW, and BPFW are also CAs.

3.2 Convex-combination-agnostic properties

We now formalize in the following definition the concept of properties of CAs that remain unchanged when one convex representation is replaced by another. As demonstrated in Section 4, the pivoting meta-algorithm (PM) ensures the preservation of these properties.

Definition 3.1 (Convex-combination-agnostic property)**.** Consider a CA and suppose that the output of the algorithm provably satisfies a property $\mathcal{P}$. We say that $\mathcal{P}$ is *convex-combination-agnostic* if property $\mathcal{P}$ also holds if CA is modified by replacing Line 4 in Algorithm 2 with the following two lines:

4a) $(\boldsymbol{\beta}^{(t+1)}, \mathcal{T}^{(t+1)}, \mathbf{x}^{(t+1)}) \leftarrow \texttt{CCU}(\boldsymbol{\alpha}^{(t)}, \mathcal{S}^{(t)}, \mathbf{x}^{(t)})$

4b) $(\boldsymbol{\alpha}^{(t+1)}, \mathcal{S}^{(t+1)}) \leftarrow$ constructed via any procedure that guarantees that $\mathbf{x}^{(t+1)} = \sum_{\mathbf{s} \in \mathcal{S}^{(t+1)}} \alpha^{(t+1)}_{\mathbf{s}} \mathbf{s}$, where $\boldsymbol{\alpha}^{(t+1)} \in \Delta_{|\mathcal{V}|}$ and $\mathcal{S}^{(t+1)} = \{\mathbf{s} \in \mathcal{T}^{(t)} \mid \alpha^{(t+1)}_{\mathbf{s}} > 0\}$.

Most properties are convex-combination-agnostic, including convergence rates of the FW variants that we consider as running examples here and which we will represent in the remainder of this section.

Theorem 3.2 (Sublinear convergence rate of FW)**.** *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a compact convex set of diameter $\delta > 0$ and let $f \colon \mathcal{C} \to \mathbb{R}$ be a convex and L-smooth function. Then, for the iterates of Algorithm 1 (FW) with line-search, short-step, or open-loop step-size rule $\eta^{(t)} = \frac{2}{t+2}$, the convergence guarantee $f(\mathbf{x}^{(t)}) - \min_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x}) \leq \frac{2L\delta^2}{t+2}$ is convex-combination-agnostic.*

Tsuji et al. (2022) showed convergence rate guarantees for BPFW similar to those of of AFW. Below, we present both the general sublinear rate as well as the linear rate for the case of $\mathcal{C}$ being a polytope.

Theorem 3.3 (Sublinear convergence rates of AFW and BPFW)**.** *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a compact convex set of diameter $\delta > 0$ and let $f \colon \mathcal{C} \to \mathbb{R}$ be a convex and L-smooth function. Then, for the iterates of Algorithms 6 (AFW) and 7 (BPFW) with line-search, the convergence guarantee $f(\mathbf{x}^{(t)}) - \min_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x}) \leq \frac{4L\delta^2}{t}$ is convex-combination-agnostic.*

Theorem 3.4 (Linear convergence rates of AFW and BPFW)**.** *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a polytope of diameter $\delta > 0$ and pyramidal width $\omega > 0$, and let $f \colon \mathcal{C} \to \mathbb{R}$ be a μ-strongly convex and L-smooth function. Then, for the iterates of Algorithms 6 (AFW) and 7 (BPFW) with line-search, the convergence guarantee $f(\mathbf{x}^{(t)}) - \min_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x}) \leq$*

Algorithm 4: Pivoting meta algorithm (PM), $(\boldsymbol{\alpha}^{(T)}, \mathcal{S}^{(T)}, \mathbf{x}^{(T)}) = \texttt{PM}(\mathbf{x}^{(0)})$

Input: $\mathbf{x}^{(0)} \in \mathcal{V}$.
Output: $\boldsymbol{\alpha}^{(T)} \in \Delta_{|\mathcal{V}|}$,
$\mathcal{S}^{(T)} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(T)} > 0\}$, and
$\mathbf{x}^{(T)} \in \mathcal{C}$, such that
$\mathbf{x}^{(T)} = \sum_{\mathbf{s} \in \mathcal{S}^{(T)}} \alpha_{\mathbf{s}}^{(T)} \mathbf{s}$.

1 $M^{(0)} \leftarrow \left(\tilde{\mathbf{x}}^{(0)}, D_n\right) \in \mathbb{R}^{(n+2)\times(n+2)}$
2 $\boldsymbol{\alpha}^{(0)} \in \Delta_{|\mathcal{V}|}$ s.t. for all $\mathbf{s} \in \mathcal{V}$,
$\alpha_{\mathbf{s}}^{(0)} \leftarrow \begin{cases} 1, & \text{if } \mathbf{s} = \mathbf{x}^{(0)} \\ 0, & \text{if } \mathbf{s} \in \mathcal{V} \setminus \{\mathbf{x}^{(0)}\} \end{cases}$
3 $\mathcal{S}^{(0)} \leftarrow \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(0)} > 0\} = \{\mathbf{x}^{(0)}\}$
4 **for** $t = 0, 1, \ldots, T-1$ **do**
5 $(\boldsymbol{\beta}^{(t+1)}, \mathcal{T}^{(t+1)}, \mathbf{x}^{(t+1)}) \leftarrow$ CCU$(\boldsymbol{\alpha}^{(t)}, \mathcal{S}^{(t)}, \mathbf{x}^{(t)})$ ▷ see Algorithm 3
6 $(M^{(t+1)}, \boldsymbol{\alpha}^{(t+1)}, \mathcal{S}^{(t+1)}) \leftarrow$ ASC$(M^{(t)}, \beta^{(t+1)}, \mathcal{T}^{(t+1)}, \mathcal{T}^{(t+1)} \setminus \mathcal{S}^{(t)})$ ▷ see Algorithm 5
7 **end**

$(f(\mathbf{x}^{(0)}) - \min_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x})) \exp(-\frac{t}{2} \min\{\frac{1}{2}, \frac{\mu\omega^2}{4L\delta^2}\})$ *is convex-combination-agnostic.*

4 The pivoting meta algorithm

We now introduce the *pivoting meta algorithm (*PM*)* in Algorithm 4, a drop-in modification applicable to existing CAs, as shown in Algorithm 2. This ensures that the size of the modified active set $\mathcal{S}^{(t)}$ remains bounded by $n+1$ for all $t \in \{0, 1, \ldots, T\}$, while preserving the convex-combination-agnostic properties of the original CA.

The main idea behind PM is to modify the active set obtained from CCU in Line 5 with the *active set cleanup algorithm* (ASC), presented in Algorithm 5. ASC takes as arguments the new weight vector $\boldsymbol{\beta}^{(t+1)} \in \Delta_{|\mathcal{V}|}$, the new active set $\mathcal{T}^{(t+1)} = \{\mathbf{s} \in \mathcal{V} \mid \beta_{\mathbf{s}}^{(t+1)} > 0\}$, the set difference between the upcoming and the current active set $\mathcal{D}^{(t)} := \mathcal{T}^{(t+1)} \setminus \mathcal{S}^{(t)} \subseteq \mathcal{T}^{(t+1)}$, and a matrix $M^{(t)} \in \mathbb{R}^{(n+2)\times(n+2)}$, such that the following hold:

1. $M^{(t)}$ is invertible, $M_{n+1,:}^{(t)} \geq \mathbf{0}^\intercal$, and $M_{n+2,:}^{(t)} \geq \mathbf{1}^\intercal$.
2. For all $i \in [n+2]$, $M_{n+1,i}^{(t)} = 0$ implies that there exists an $\mathbf{s} \in \mathcal{S}^{(t)}$ such that $\tilde{\mathbf{s}} = M_{:,i}^{(t)}$.
3. For all $\mathbf{s} \in \mathcal{S}^{(t)}$ there exists an $i \in [n+2]$ such that $M_{:,i}^{(t)} = \tilde{\mathbf{s}}$,

In Line 6 of PM, ASC then constructs the matrix $M^{(t+1)}$, the modified weight vector $\boldsymbol{\alpha}^{(t+1)} \in \Delta_{|\mathcal{V}|}$ and the modified active set $\mathcal{S}^{(t+1)} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(t+1)} > 0\}$, such that $\mathbf{x}^{(t+1)} = \sum_{\mathbf{s} \in \mathcal{T}^{(t+1)}} \beta_{\mathbf{s}}^{(t+1)} \mathbf{s} = \sum_{\mathbf{s} \in \mathcal{S}^{(t+1)}} \alpha_{\mathbf{s}}^{(t+1)} \mathbf{s}$, $|\mathcal{S}^{(t+1)}| \leq n+1$, and the three properties above are now satisfied for $t+1$.

By the definition of CCUs, $\mathcal{D}^{(t)}$ is either empty or contains exactly one vertex $\mathbf{v}^{(t)} \in \mathcal{T}^{(t+1)} \setminus \mathcal{S}^{(t)}$. The former case is straightforward as the size of the active set does not increase. In the latter case, ASC performs a pivoting update akin to the simplex algorithm that guarantees that $|\mathcal{S}^{(t+1)}| \leq n+1$. Performing pivot updates necessitates maintaining the matrix $M^{(t)}$ throughout PM's execution. We formalize the properties of ASC below.

Proposition 4.1 (Properties of ASC)**.** *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a compact convex set, let $\mathcal{V} = \operatorname{vert}(\mathcal{C})$, let $N \in \mathbb{R}^{(n+2)\times(n+2)}$, let $\boldsymbol{\beta} \in \Delta_{|\mathcal{V}|}$, let $\mathcal{T} = \{\mathbf{s} \in \mathcal{V} \mid \beta_{\mathbf{s}} > 0\}$, and let $\mathcal{D} \subseteq \mathcal{T}$ such that $|\mathcal{D}| \leq 1$. Assume that the following hold:*

1. *N is invertible, $N_{n+1,:} \geq \mathbf{0}^\intercal$, and $N_{n+2,:} \geq \mathbf{1}^\intercal$.*
2. *For all $i \in [n+2]$, $N_{n+1,i} = 0$ implies that there exists an $\mathbf{s} \in \mathcal{T} \setminus \mathcal{D}$ such that $\tilde{\mathbf{s}} = N_{:,i}$.*
3. *For all $\mathbf{s} \in \mathcal{T} \setminus \mathcal{D}$, there exists an $i \in [n+2]$ such that $N_{:,i} = \tilde{\mathbf{s}}$.*

Let $(M, \boldsymbol{\alpha}, \mathcal{S}) = \texttt{ASC}(N, \boldsymbol{\beta}, \mathcal{T}, \mathcal{D})$, where $M \in \mathbb{R}^{(n+2)\times(n+2)}$, $\boldsymbol{\alpha} \in \Delta_{|\mathcal{V}|}$ and $\mathcal{S} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}} > 0\}$. Then we have:

4. *M is invertible, $M_{n+1,:} \geq \mathbf{0}^\intercal$, and $M_{n+2,:} \geq \mathbf{1}^\intercal$.*
5. *For all $i \in [n+2]$, $\lambda_i > 0$ implies that there exists an $\mathbf{s} \in \mathcal{S} \cap \mathcal{T}$ such that $\tilde{\mathbf{s}} = M_{:,i} = Q_{:,i}$.*
6. *$\mathbf{x} := \sum_{\mathbf{s} \in \mathcal{T}} \beta_{\mathbf{s}} \mathbf{s} = \sum_{\mathbf{s} \in \mathcal{S}} \alpha_{\mathbf{s}} \mathbf{s}$.*
7. *For all $i \in [n+2]$, $M_{n+1,i} = 0$ implies that there exists an $\mathbf{s} \in \mathcal{S}$ such that $\tilde{\mathbf{s}} = M_{:,i}$.*
8. *For all $\mathbf{s} \in \mathcal{S}$, there exists an $i \in [n+2]$ such that $M_{:,i} = \tilde{\mathbf{s}}$.*
9. *It holds that $\mathcal{S} \subseteq \mathcal{T}$ and $|\mathcal{S}| \leq n+1$.*

As a direct consequence, we formalize the properties of PM in the main result of the paper below.

Theorem 4.2 (Properties of PM)**.** *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a compact convex set and $\mathbf{x}^{(0)} \in \operatorname{vert}(\mathcal{C}) = \mathcal{V}$. Given a specific* CA*, let $(\boldsymbol{\alpha}^{(T)}, \mathcal{S}^{(T)}, \mathbf{x}^{(T)}) = \texttt{PM}(\mathbf{x}^{(0)})$ denote the output of its modification with* PM*. Then, for all $t \in \{0, 1, \ldots, T\}$ the following hold:*

1. *$\boldsymbol{\alpha}^{(t)} \in \Delta_{|\mathcal{V}|}$ and $\mathcal{S}^{(t)} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(t)} > 0\}$.*

Algorithm 5: Active set cleanup algorithm (ASC), $(M, \boldsymbol{\alpha}, \mathcal{S}) = \texttt{ASC}(N, \boldsymbol{\beta}, \mathcal{T}, \mathcal{D})$

Input: $N \in \mathbb{R}^{(n+2)\times(n+2)}$ invertible, $\boldsymbol{\beta} \in \Delta_{|\mathcal{V}|}$, $\mathcal{T} = \{\mathbf{s} \in \mathcal{V} \mid \beta_{\mathbf{s}} > 0\}$, and $\mathcal{D} \subseteq \mathcal{T}$ such that $|\mathcal{D}| \leq 1$.
Output: $M \in \mathbb{R}^{(n+2)\times(n+2)}$ invertible, $\boldsymbol{\alpha} \in \Delta_{|\mathcal{V}|}$, and $\mathcal{S} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}} > 0\}$.

1 **if** $\mathcal{D} = \emptyset$ **then**
2 $\quad \boldsymbol{\lambda} \in \Delta_{n+2}$ s.t. for all $i \in [n+2]$,
$$\lambda_i \leftarrow \begin{cases} \beta_{\mathbf{s}}, & \text{if } N_{:,i} = \tilde{\mathbf{s}} \text{ for some } \mathbf{s} \in \mathcal{T} \\ 0, & \text{else} \end{cases}$$
3 $\quad Q \leftarrow N$
4 **else**
5 $\quad \mathbf{v} \in \mathcal{D}$ ▷ `v is unique`
6 $\quad A \leftarrow (N, \tilde{\mathbf{v}}) \in \mathbb{R}^{(n+2)\times(n+3)}$
7 $\quad \boldsymbol{\mu} \in \Delta_{n+3}$ s.t. for all $i \in [n+3]$,
$$\mu_i \leftarrow \begin{cases} \beta_{\mathbf{s}}, & \text{if } A_{:,i} = \tilde{\mathbf{s}} \text{ for some } \mathbf{s} \in \mathcal{T} \\ 0, & \text{else} \end{cases}$$
8 $\quad \mathbf{r} \leftarrow -N^{-1}\tilde{\mathbf{v}} \in \mathbb{R}^{n+2}$
9 $\quad k \in \operatorname{argmin}_{i \in [n+2], r_i < 0} -\mu_i / r_i$
10 $\quad \theta^* \leftarrow -\mu_k / r_k \geq 0$
11 $\quad \boldsymbol{\lambda} \in \Delta_{n+2}$ s.t. for all $i \in [n+2]$,
$$\lambda_i \leftarrow \begin{cases} \theta^*, & \text{if } i = k \\ \mu_i + \theta^* r_i, & \text{if } i \neq k \end{cases}$$
12 $\quad Q \in \mathbb{R}^{(n+2)\times(n+2)}$ s.t. for all $i \in [n+2]$,
$$Q_{:,i} \leftarrow \begin{cases} \tilde{\mathbf{v}}, & \text{if } i = k \\ N_{:,i}, & \text{if } i \neq k \end{cases}$$
13 **end**
14 $\ell \in \{i \in [n+2] \mid Q_{n+1,i} \neq 0\}$
15 $M \in \mathbb{R}^{(n+2)\times(n+2)}$ s.t. for all $i \in [n+2]$,
$$M_{:,i} \leftarrow \begin{cases} Q_{:,i} + Q_{:,\ell}, & \text{if } \lambda_i = 0 \ \&\ Q_{n+1,i} = 0 \\ Q_{:,i}, & \text{else} \end{cases}$$
16 $\boldsymbol{\alpha} \in \Delta_{|\mathcal{V}|}$ s.t. for all $\mathbf{s} \in \mathcal{V}$,
$$\alpha_{\mathbf{s}} \leftarrow \begin{cases} \lambda_i, & \text{if } \exists\, i \in [n+2] \text{ s.t. } \tilde{\mathbf{s}} = M_{:,i} \\ 0, & \text{for all other } \mathbf{s} \in \mathcal{V} \end{cases}$$
17 $\mathcal{S} \leftarrow \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}} > 0\}$

2. *$M^{(t)} \in \mathbb{R}^{(n+2)\times(n+2)}$ is invertible, $M^{(t)}_{n+1,:} \geq \mathbf{0}^\intercal$, and $M^{(t)}_{n+2,:} \geq \mathbf{1}^\intercal$.*

3. *$\mathbf{x}^{(t)} = \sum_{\mathbf{s}\in\mathcal{T}^{(t)}} \beta^{(t)}_{\mathbf{s}}\mathbf{s} = \sum_{\mathbf{s}\in\mathcal{S}^{(t)}} \alpha^{(t)}_{\mathbf{s}}\mathbf{s}$. For $t = 0$, let $\mathcal{T}^{(0)} := \mathcal{S}^{(0)}$ and $\boldsymbol{\beta}^{(0)} := \boldsymbol{\alpha}^{(0)}$.*

4. *$\mathcal{S}^{(t)} \subseteq \mathcal{T}^{(t)}$ and $|\mathcal{S}^{(t)}| \leq n + 1$.*

Moreover, PM's output satisfies the same convex-combination-agnostic properties CA's output would satisfy.

We will later see that the dimension dependence in Property 4 of Theorem 4.2 can be replaced with $\dim(\mathcal{C})$, see Corollary 5.2.

4.1 Comparison to IRR

A first proposal for the reduction of the active set cardinality in CA algorithms motivated by Carathéodory's theorem was presented in Beck and Shtern (2017) as the *incremental representation reduction algorithm* (IRR). It maintains two matrices throughout the algorithm, T and W, with T a product of elementary matrices and W a matrix in row echelon form. The latter will, in general, not be sparse, and the whole analysis of the authors is not considering the potential sparsity of vertices, and thus of the resulting matrices. In contrast, PM operates directly on a matrix formed by the (extended) vertices and explicitly includes the linear system solving step for which the algorithmic details remain at the discretion of the implementation, instead of relying on the formation of a row-echelon matrix which is rarely the preferred choice to solve sparse linear systems. The workspace required by PM only consists of $M^{(t)}$ and at most one additional column, which is of a fixed size of only $\mathcal{O}(n_v(n+3))$, with n_v the number of structural non-zero terms in a single vertex, assuming Line 15 does not merge vertex columns into non-vertex ones in too many iterations. This bound is typically much lower than IRR's space requirements of order $\mathcal{O}(n^2)$ for many applications in which the support of vertices is small.

Finally, as detailed in Section 5, numerical errors in rank-one updates quickly accumulate beyond a practical level to compute the weights of the active set. IRR requires maintaining the row-echelon matrix throughout iterations and does not specify a mechanism to start from a given non-singleton active set, which means numerical errors inevitably accumulate throughout the algorithm. PM on the other hand can leverage rank-one updates and at any step recompute a factorization from scratch to solve the sparse linear system $M\mathbf{r} = \mathbf{v}$, or leverage any alternative algorithm for linear systems, making it more flexible for numerically challenging instances. Furthermore, PM requires significant computational work only when $\mathcal{T}^{(t+1)}$ contains a vertex not already contained in $\mathcal{S}^{(t)}$, that is, when a new vertex is introduced into the active set. On steps operating on known vertices only, see Line 1 of ASC (Algorithm 5), the only subroutine in $\mathcal{O}(n_v n_d)$ is Line 15, with n_d the number of vertices dropped at the given step. In practice, n_d remains quite small as one rarely drops a lot of vertices from the active set. In contrast, IRR always incurs a computational overhead of $\mathcal{O}(n^2)$.

5 Active set identification results

We will now present improvements to the bound $|\mathcal{S}^{(t)}| \leq n+1$ established for PM in Theorem 4.2. First, we refine the bound to $|\mathcal{S}^{(t)}| \leq \dim(\mathcal{C})+1$ for any iteration t. Then, we establish that if there exists an iteration R such that $\mathcal{S}^{(t)} \subseteq \mathcal{C}^* \subseteq \mathcal{C}$ for all $t \geq R$, then $|\mathcal{S}^{(t)}| \leq \dim(\mathcal{C}^*)+1$ instead of $|\mathcal{S}^{(t)}| \leq n+1$ for all $t \geq R$. This result is tied to so-called active set identification properties of some FW variants, known for a long time in specific settings (Guélat and Marcotte, 1986), and recentlt generalized in Bomze et al. (2020) for AFW when $\mathcal{C}$ is a polytope and some other mild assumptions are met.[2]

Below, we present the main result of this section.

Theorem 5.1 (Active set identification with PM)**.** *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a compact convex set and $\mathbf{x}^{(0)} \in \operatorname{vert}(\mathcal{C}) = \mathcal{V}$. Given a specific Algorithm 2 (CA), let $(\boldsymbol{\alpha}^{(T)}, \mathcal{S}^{(T)}, \mathbf{x}^{(T)}) = \texttt{PM}(\mathbf{x}^{(0)})$ denote the output of its modification with Algorithm 4 (PM). Suppose that there exists an iteration $R \in \{0, 1, \ldots, T\}$ such that $\mathcal{S}^{(t)} \subseteq \mathcal{C}^*$ for all $t \in \{R, R+1 \ldots, T\}$, where $\emptyset \neq \mathcal{C}^* \subseteq \mathcal{C}$. Then, $|\mathcal{S}^{(t)}| \leq \dim(\mathcal{C}^*)+1$ for all $t \in \{R, R+1 \ldots, T\}$.*

The result above implies that the bound $|\mathcal{S}^{(t)}| \leq n+1$ in Theorem 4.2 can be refined to $|\mathcal{S}^{(t)}| \leq \dim(\mathcal{C})+1$ for any iteration t.

Corollary 5.2. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a compact convex set and $\mathbf{x}^{(0)} \in \operatorname{vert}(\mathcal{C}) = \mathcal{V}$. Given a specific Algorithm 2 (CA), let $(\boldsymbol{\alpha}^{(T)}, \mathcal{S}^{(T)}, \mathbf{x}^{(T)}) = \texttt{PM}(\mathbf{x}^{(0)})$ denote the output of its modification with Algorithm 4 (PM). Then, $|\mathcal{S}^{(t)}| \leq \dim(\mathcal{C})+1$ for all $t \in \{0, 1, \ldots, T\}$.*

We now focus on the application of Theorem 5.1 to the setting characterized in Bomze et al. (2020).

Consider the optimization problem (OPT) with $\mathcal{C} \subseteq \mathbb{R}^n$ a polytope, $f\colon \mathcal{C} \to \mathbb{R}$ a convex and L-smooth function, and the set of optimal solutions $\operatorname{argmin}_{\mathbf{x}\in\mathcal{C}} f(\mathbf{x})$ lying in the relative interior of a face $\mathcal{C}^* \subseteq \mathcal{C}$. This setting is particularly relevant to the FW community, encompassing applications such as sparse signal recovery, sparse regression, and sparse logistic regression, where FW variants construct iterates that are sparse convex combinations of vertices of the feasible region. Second, vanilla FW with line-search or short-step is known to exhibit zig-zagging behavior and, in some cases, converges at a rate of at most $\Omega(1/t^{1+\epsilon})$ for any $\epsilon > 0$ (Wolfe, 1970). This motivated the development of accelerated variants (Lacoste-Julien and Jaggi, 2015; Garber and Meshi, 2016; Braun et al., 2019; Combettes and Pokutta, 2020; Garber and Wolf, 2021) and the exploration of alternative step sizes (Wirth et al., 2023b,c) to surpass the lower bound established by Wolfe (1970). Finally, the identification of the optimal face $\mathcal{C}^*$ is a crucial step known as active set identification; not to be confused with the active sets maintained by the CA's. Once $\mathcal{C}^*$ is identified, the optimization problem (OPT) simplifies to one over $\mathcal{C}^*$, which often has a much lower dimension compared to $\mathcal{C}$ (Bomze et al., 2019, 2020).

Recently, Bomze et al. (2020) provided insights into the settings where the away-step Frank-Wolfe algorithm (AFW) guarantees the identification of the active set after a finite number of iterations. Here, we present a simplified version[3] of their result, Bomze et al. (2020, Theorem C.1), along with our observation regarding its convex-combination-agnostic nature.

Theorem 5.3 (Theorem C.1, Bomze et al., 2020)**.** *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a polytope, let $f\colon \mathcal{C} \to \mathbb{R}$ be a convex and L-smooth function, and suppose that $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{C}} f(\mathbf{x})$ is unique and $\mathbf{x}^* \in \operatorname{rel.int}(\mathcal{C}^*)$, where $\mathcal{C}^* \in \operatorname{faces}(\mathcal{C})$. Then, for T large enough, for the iterations of Algorithm 6 (AFW) with line-search, there exists an iteration $R \in \{0, 1, \ldots, T\}$ such that $\mathbf{x}^{(t)} \in \mathcal{C}^*$ for all $t \in \{R, R+1, \ldots, T\}$. This property is convex-combination-agnostic.*

We similarly establish the active set identification property for BPFW and its convex-combination agnosticity in Theorem 9.4. We obtain the following active set identification result for PM applied to AFW and BPFW.

Corollary 5.4 (Active set identification with PM applied to AFW and BPFW)**.** *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a polytope, let $f\colon \mathcal{C} \to \mathbb{R}$ be a convex and L-smooth function, and suppose that $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{C}} f(\mathbf{x})$ is unique and $\mathbf{x}^* \in \operatorname{rel.int}(\mathcal{C}^*)$, where $\mathcal{C}^* \in \operatorname{faces}(\mathcal{C})$. Let $(\boldsymbol{\alpha}^{(t)}, \mathcal{S}^{(t)}, \mathbf{x}^{(t)}) = \texttt{PM}(\mathbf{x}^{(0)})$ denote the iterations of Algorithms 6 (AFW) or 7 (BPFW) with line-search modified with Algorithm 4 (PM). Then, there exists an iteration $R \geq 0$ such that $|\mathcal{S}^{(t)}| \leq \dim(\mathcal{C}^*)+1$ for all $t \geq R$.*

6 Numerical experiments

We will now provide a brief set of numerical experiments. We discuss numerical robustness and stability in Section 10 and more extensive experiments as well as a detailed description of the setup in Section 11 in the supplementary materials.

We assess PM on efficiency in terms of function value and FW gap convergence, and sparsity of the obtained solutions, compared to the standard and lazified versions (see (Braun et al., 2017) for details) of AFW and BPFW, which also produce notably sparse solutions. Our

[2] The result of Bomze et al. (2020) is proved in their paper for AFW. In Appendix 9, we prove that a similar result also holds for BPFW.

[3] Bomze et al. (2020) also provide sufficient conditions for their result to hold when there are multiple optimizers in the relative interior of an optimal face.

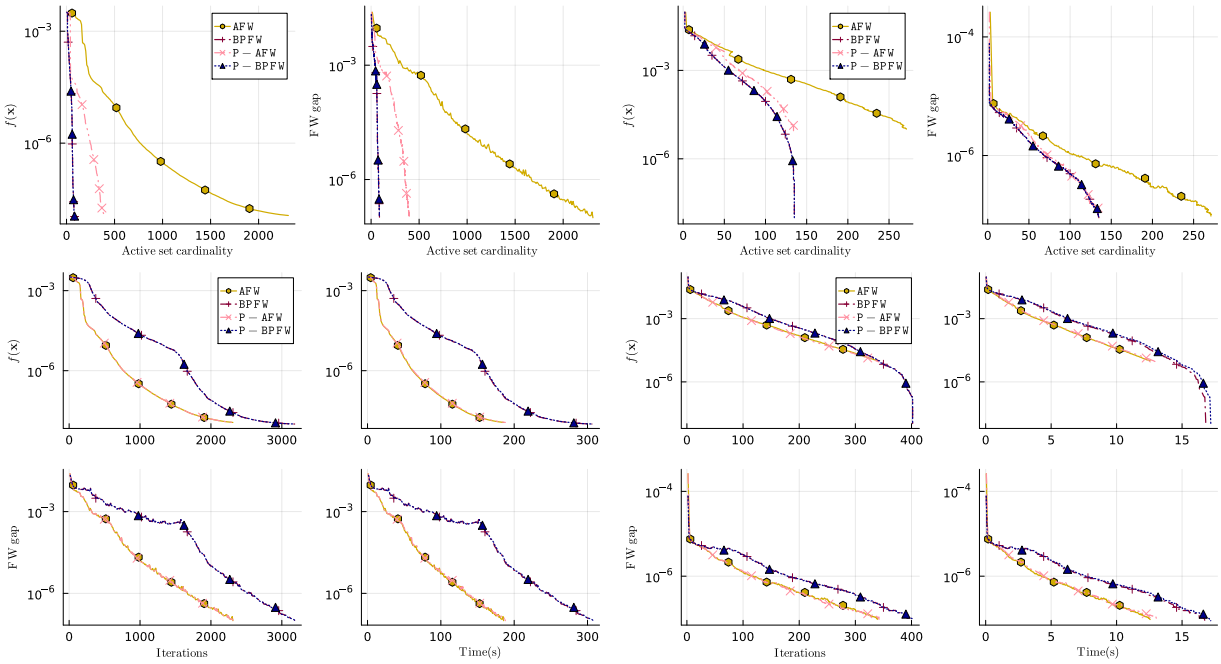

Figure 1: (Left) Logistic regression, $\tau = 60$. (Right) Signal recovery, $\tau_f = 20$, $m = 6000$, $n = 14000$. All variants are non-lazified. As can be seen in both cases, `PM` can significantly improve sparsity while maintaining identical convergence rates and in cases where the algorithm (such as e.g., BPFW) produces already sparse iterates it does not harm the algorithm.

algorithm is implemented in Julia (Bezanson et al., 2017) v1.9.2 and builds on the `FrankWolfe.jl` package (Besançon et al., 2022). The sparse linear systems are solved with the LU decomposition of the `UMFPACK` library (Davis, 2004). Plots are log-linear, function values are shifted so that the minimum on each plot reaches 10^{-8}. The prefix `L-` denotes the lazified version of an algorithm, the prefix `P-` for the `PM` variant.

6.1 Sparse logistic regression

We run all algorithms on a logistic regression problem with an ℓ_1-norm ball constraint:

$$\min_{\mathbf{x}:\|\mathbf{x}\|_1 \leq \tau} \frac{1}{m} \sum_{i=1}^{m} \log(1 + \exp(-y_i \mathbf{a}_i^\intercal \mathbf{x})).$$

where m is the number of samples, $\tau > 0$ is the ℓ_1-norm ball radius, $y_i \in \{-1, 1\}$ encodes the class and $\mathbf{a}_i$ the feature vector for the ith sample. We use the Gisette dataset (Guyon et al., 2004), which contains 5000 features, we run logistic regression on the validation set containing only 1000 samples and thus more prone to overfitting without a sparsity-inducing regularization constraint. The convergence of the pivoting variants of both `AFW` and `BPFW` converge similarly in both function value and FW gap as their standard counterparts as shown in Figure 1 (left). Even though `BPFW` typically maintains a smaller active set, it converges at a slower rate than the away-step `FW` variants, both in function value and FW gap. `P-AFW` drastically improves the sparsity of the `AFW` iterates, while maintaining the same convergence in function value and FW gap. This highlights one key property of our meta algorithm: it can be adapted to several `FW` variants, benefiting from their convergence rate while improving their sparsity.

6.2 Sparse signal recovery

We assess `PM` on a sparse signal recovery problem:

$$\min_{\mathbf{x}:\|\mathbf{x}\|_1 \leq \tau} \|A\mathbf{x} - \mathbf{y}\|_2^2,$$

with $A \in \mathbb{R}^{m \times n}$, $\mathbf{y} \in \mathbb{R}^m$, $n > m$. We generate the entries of the sensing matrix A i.i.d. from a standard Gaussian distribution and $\mathbf{y}$ by adding Gaussian noise with unit standard deviation to $A\mathbf{x}_{\text{true}}$, with $\mathbf{x}_{\text{true}}$ an underlying sparse vector, with 30% of non-zero terms, all taking entries sampled from a standard Gaussian distribution. The radius τ is computed as $\tau = \|\mathbf{x}_{\text{true}}\|_1 / \tau_f$ for different values of τ_f.

Figure 1 (right) illustrates the results of the non-lazified version of `BPFW` and `AFW` and their pivoting counterparts. `P-AFW` converges at the same rate as `AFW` in function value and FW gap while being faster than both `BPFW` variants, terminating before them, while maintaining an active set twice as sparse as `AFW`.

Acknowledgments

We thank Zev Woodstock for providing valuable feedback for an earlier version of this manuscript. Research reported in this paper was partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH$^+$ (EXC-2046/1, project ID 390685689, BMS Stipend). Mathieu Besançon was partially supported by MIAI at Université Grenoble Alpes (ANR-19-P3IA-0003).

References

Bach, F. (2021). On the effectiveness of Richardson extrapolation in data science. *SIAM Journal on Mathematics of Data Science*, 3(4):1251–1277.

Bach, F., Lacoste-Julien, S., and Obozinski, G. (2012). On the equivalence between herding and conditional gradient algorithms. In *Proceedings of the International Conference on Machine Learning*, pages 1355–1362. PMLR.

Beck, A. and Shtern, S. (2017). Linearly convergent away-step conditional gradient for non-strongly convex functions. *Mathematical Programming*, 164:1–27.

Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to linear optimization*, volume 6. Athena scientific Belmont, MA.

Besançon, M., Carderera, A., and Pokutta, S. (2022). Frankwolfe. jl: A high-performance and flexible toolbox for Frank–Wolfe algorithms and conditional gradients. *INFORMS Journal on Computing*, 34(5):2611–2620.

Bettiol, E., Buchheim, C., De Santis, M., and Rinaldi, F. (2024). An oracle-based framework for robust combinatorial optimization. *Journal of Global Optimization*, 88(1):27–51.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98.

Bomze, I. M., Rinaldi, F., and Bulo, S. R. (2019). First-order methods for the impatient: Support identification in finite time with convergent Frank–Wolfe variants. *SIAM Journal on Optimization*, 29(3):2211–2226.

Bomze, I. M., Rinaldi, F., and Zeffiro, D. (2020). Active set complexity of the away-step Frank–Wolfe algorithm. *SIAM Journal on Optimization*, 30(3):2470–2500.

Braun, G., Carderera, A., Combettes, C. W., Hassani, H., Karbasi, A., Mokhtari, A., and Pokutta, S. (2022). Conditional gradient methods. *arXiv preprint arXiv:2211.14103*.

Braun, G., Pokutta, S., Tu, D., and Wright, S. (2019). Blended conditonal gradients. In *Proceedings of the International Conference on Machine Learning*, pages 735–743. PMLR.

Braun, G., Pokutta, S., and Zink, D. (2017). Lazifying conditional gradient algorithms. In *Proceedings of the International Conference on Machine Learning*, pages 566–575. PMLR.

Bugg, C. X., Chen, C., and Aswani, A. (2022). Nonnegative tensor completion via integer optimization. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Proceedings of Advances in Neural Information Processing Systems*.

Cai, T. T. and Zhang, A. (2013). Sparse representation of a polytope and recovery of sparse signals and low-rank matrices. *IEEE transactions on information theory*, 60(1):122–132.

Carathéodory, C. (1907). Über den variabilitätsbereich der koeffizienten von potenzreihen, die gegebene werte nicht annehmen. *Mathematische Annalen*, 64(1):95–115.

Carderera, A., Pokutta, S., Schütte, C., and Weiser, M. (2021). CINDy: Conditional gradient-based identification of non-linear dynamics–noise-robust recovery. *arXiv preprint arXiv:2101.02630*.

Combettes, C. and Pokutta, S. (2020). Boosting Frank-Wolfe by chasing gradients. In *Proceedings of the International Conference on Machine Learning*, pages 2111–2121. PMLR.

Combettes, C. W. and Pokutta, S. (2023). Revisiting the approximate Carathéodory problem via the Frank-Wolfe algorithm. *Mathematical Programming*, 197(1):191–214.

Condat, L. (2016). Fast projection onto the simplex and the l1 ball. *Mathematical Programming*, 158(1-2):575–585.

Davis, T. A. (2004). Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):196–199.

Dunn, J. C. and Harshbarger, S. (1978). Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62(2):432–444.

Filippozzi, R., Gonçalves, D. S., and Santos, L.-R. (2023). First-order methods for the convex hull membership problem. *European Journal of Operational Research*, 306(1):17–33.

Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110.

Garber, D. and Meshi, O. (2016). Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 1009–1017. PMLR.

Garber, D. and Wolf, N. (2021). Frank-Wolfe with a nearest extreme point oracle. In *Proceedings of International Conference on Learning Theory*, pages 2103–2132. PMLR.

Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H. (1987). Maintaining LU factors of a general sparse matrix. *Linear Algebra and its Applications*, 88:239–270.

Goldfarb, D., Iyengar, G., and Zhou, C. (2017). Linear convergence of stochastic Frank-Wolfe variants. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 1066–1074. PMLR.

Guélat, J. and Marcotte, P. (1986). Some comments on Wolfe's 'away step'. *Mathematical Programming*, 35(1):110–119.

Guo, X., Yao, Q., and Kwok, J. (2017). Efficient sparse low-rank tensor completion using the Frank-Wolfe algorithm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Guyon, I., Gunn, S., Ben-Hur, A., and Dror, G. (2004). Result analysis of the nips 2003 feature selection challenge. *Proceedings of Advances in Neural Information Processing Systems*, 17.

Holloway, C. A. (1974). An extension of the Frank and Wolfe method of feasible directions. *Mathematical Programming*, 6(1):14–27.

Huangfu, Q. and Hall, J. J. (2015). Novel update techniques for the revised simplex method. *Computational Optimization and Applications*, 60:587–608.

Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the International Conference on Machine Learning*, pages 427–435. PMLR.

Kerdreux, T., Liu, L., Lacoste-Julien, S., and Scieur, D. (2021). Affine invariant analysis of Frank-Wolfe on strongly convex sets. In *Proceedings of the International Conference on Machine Learning*, pages 5398–5408. PMLR.

Lacoste-Julien, S. and Jaggi, M. (2013). An affine invariant linear convergence analysis for Frank-Wolfe algorithms. *arXiv preprint arXiv:1312.7864*.

Lacoste-Julien, S. and Jaggi, M. (2015). On the global linear convergence of Frank-Wolfe optimization variants. In *Proceedings of Advances in Neural Information Processing Systems*, pages 496–504.

Levitin, E. S. and Polyak, B. T. (1966). Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6(5):1–50.

Macdonald, J., Besançon, M. E., and Pokutta, S. (2022). Interpretable neural networks with Frank-Wolfe: Sparse relevance maps and relevance orderings. In *Proceedings of the International Conference on Machine Learning*, pages 14699–14716. PMLR.

Mu, C., Zhang, Y., Wright, J., and Goldfarb, D. (2016). Scalable robust matrix recovery: Frank–Wolfe meets proximal methods. *SIAM Journal on Scientific Computing*, 38(5):A3291–A3317.

Pedregosa, F., Askari, A., Negiar, G., and Jaggi, M. (2018). Step-size adaptivity in projection-free optimization. *arXiv preprint arXiv:1806.05123*.

Pena, J. and Rodriguez, D. (2019). Polytope conditioning and linear convergence of the Frank–Wolfe algorithm. *Mathematics of Operations Research*, 44(1):1–18.

Peña, J. F. (2023). Affine invariant convergence rates of the conditional gradient method. *SIAM Journal on Optimization*, 33(4):2654–2674.

Pokutta, S. (2024). The Frank-Wolfe algorithm: a short introduction. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 126:3–35.

Pokutta, S., Spiegel, C., and Zimmer, M. (2020). Deep neural network training with Frank-Wolfe. *arXiv preprint arXiv:2010.07243*.

Rao, N., Shah, P., and Wright, S. (2015). Forward–backward greedy algorithms for atomic norm regularization. *IEEE Transactions on Signal Processing*, 63(21):5798–5811.

Schork, L. and Gondzio, J. (2017). Permuting spiked matrices to triangular form and its application to the Forrest-Tomlin update. Technical report, Technical Report ERGO-17-002, University of Edinburgh.

Tsuji, K. K., Tanaka, K., and Pokutta, S. (2022). Pairwise conditional gradients without swap steps and sparser kernel herding. In *Proceedings of the International Conference on Machine Learning*, pages 21864–21883. PMLR.

Wirth, E., Kera, H., and Pokutta, S. (2023a). Approximate vanishing ideal computations at scale. In *Proceedings of the International Conference on Learning Representations*.

Wirth, E., Kerdreux, T., and Pokutta, S. (2023b). Acceleration of Frank-Wolfe algorithms with open-loop step-sizes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 77–100. PMLR.

Wirth, E., Pena, J., and Pokutta, S. (2023c). Accelerated affine-invariant convergence rates of the Frank-Wolfe algorithm with open-loop step-sizes. *arXiv preprint arXiv:2310.04096*.

Wirth, E. and Pokutta, S. (2022). Conditional gradients for the approximately vanishing ideal. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 2191–2209. PMLR.

Wolfe, P. (1970). Convergence theory in nonlinear programming. *Integer and Nonlinear Programming*, pages 1–36.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes/No/Not Applicable] Yes, all relevant properties of our algorithms are discussed and presented rigorously. See Sections 2–10.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable] Yes. See Sections 2–10.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable] Yes.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable] Yes, our theoretical results strive to be self-contained. See Sections 2–10.
   (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable] Yes, see Appendix 7.
   (c) Clear explanations of any assumptions. [Yes/No/Not Applicable] Yes. See Sections 2–10.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable] The instructions are in the README of the supplementary directory.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable] Yes, everything available in the supplementary materials.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/Not Applicable] Not Applicable
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable] Yes, beginning of Section 11.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable] Yes.
   (b) The license information of the assets, if applicable. [Yes/No/Not Applicable] Not applicable.
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable] Not applicable.
   (d) Information about consent from data providers/curators. [Yes/No/Not Applicable] Not applicable.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable] Not applicable.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes/No/Not Applicable] Not applicable.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable] Not applicable.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable] Not applicable.

7 Missing proofs

Proof of Theorem 3.2. The convergence result does not depend on any properties of convex combinations (Jaggi, 2013). □

Proof of Theorem 3.3. The proof in Tsuji et al. (2022) does not require any information on the convex combination except when determining an upper bound on the number of so-called drop steps, steps which drop certain vertices from the active set. Since the number of drop steps can only decrease when replacing the active set at iteration t with a potentially smaller active set, the convergence rate guarantee is convex-combination-agnostic. The proof in Tsuji et al. (2022) designed for `BPFW` also applies to `AFW`. □

Proof of Theorem 3.4. The argument is identical to the one in the proof of Theorem 3.3. □

Proof of Proposition 4.1. The proof is organized as follows: We first prove Properties 4–6 depending on whether the if (Line 1) or the else (Line 4) clauses are executed. Then, we prove Properties 7–9. First, suppose that $\mathcal{D} = \emptyset$. Then:

4. By Lines 3 and 15, M is obtained from $Q = N$ via elementary column additions. By Assumption 1, Property 4 is satisfied.
5. Let $i \in [n+2]$ such that $\lambda_i > 0$. By Lines 2 and 3, there exists an $\mathbf{s} \in \mathcal{T}$ such that $\tilde{\mathbf{s}} = Q_{:,i}$. By Line 15, $M_{:,i} = Q_{:,i}$. By Lines 16 and 17, $\mathbf{s} \in \mathcal{S}$, proving Property 5.
6. By Lines 1–3 and Assumption 3, $Q\boldsymbol{\lambda} = \tilde{\mathbf{x}}$. By Lines 15–17 and Property 5, $\sum_{\mathbf{s}\in\mathcal{S}} \alpha_{\mathbf{s}}\tilde{\mathbf{s}} = M\boldsymbol{\lambda} = Q\lambda = \tilde{\mathbf{x}}$. Thus, Property 6 is satisfied.

Second, suppose that $\mathcal{D} = \{\mathbf{v}\}$, that is, we consider Line 4. By Assumption 2, there does not exist an $i \in [n+2]$ such that $\tilde{\mathbf{v}} = N_{:,i}$. Let $P = \{\boldsymbol{\gamma} \in \mathbb{R}^{n+3} \mid A\boldsymbol{\gamma} = \tilde{\mathbf{x}}, \boldsymbol{\gamma} \geq \mathbf{0}\} \subseteq [0,1]^{n+3}$. By Lines 5–7, $\tilde{\mathbf{x}} = A\boldsymbol{\mu}$ and $\boldsymbol{\mu}$ is a feasible solution for the optimization problem

$$\begin{aligned} \min \ & -\mathbf{e}_{n+3}^\intercal \boldsymbol{\gamma} \\ \text{subject to } & \boldsymbol{\gamma} \in P. \end{aligned} \tag{7.1}$$

Intuitively, Algorithm 5 performs one pivoting step akin to the simplex algorithm starting from the feasible solution $\boldsymbol{\mu}$. We first prove that $\mathbf{r} = -N^{-1}\tilde{\mathbf{v}}$ has at least one negative entry. Suppose towards a contradiction that $\mathbf{r} \geq \mathbf{0}$. Thus, $\mathbf{d} = \begin{pmatrix} \mathbf{r} \\ 1 \end{pmatrix} \geq \mathbf{0}$. Since $A\mathbf{d} = -N\mathbf{r} + \tilde{\mathbf{v}} = N(-N^{-1}\tilde{\mathbf{v}}) + \tilde{\mathbf{v}} = \mathbf{0}$, the vector $\boldsymbol{\mu} + \theta\mathbf{d}$ is infeasible only if one of its components is negative for some $\theta \geq 0$. Since $\boldsymbol{\mu} \geq \mathbf{0}$ and $\mathbf{d} \geq \mathbf{0}$, we have $\boldsymbol{\mu} + \theta\mathbf{d} \geq \mathbf{0}$ for all $\theta \geq 0$, implying that $P \subseteq [0,1]^{n+3}$ is unbounded, a contradiction. Then:

4. Since $\mathbf{r} \not\geq \mathbf{0}$, it holds that $k \in \operatorname{argmin}_{i\in[n+2], r_i<0} -\frac{\mu_i}{r_i}$ as in Line 9 exists. Thus, in Line 12, Algorithm 5 constructs the matrix $Q = (N_{:,1}, \ldots, N_{:,k-1}, \tilde{\mathbf{v}}, N_{:,k+1}, \ldots, N_{:,n+2}) \in \mathbb{R}^{(n+2)\times(n+2)}$. The columns $N_{:,i}$, $i \neq k$, and $\tilde{\mathbf{v}}$ are linearly independent (Bertsimas and Tsitsiklis, 1997, Theorem 3.2). Thus, by Assumption 1 and Line 12, Q is invertible, $Q_{n+1,:} \geq \mathbf{0}^\intercal$, and $Q_{n+2,:} \geq \mathbf{1}^\intercal$. By Line 15, M is obtained from Q via elementary column additions. Thus, Property 4 is satisfied.
5. Let $i \in [n+2]$ such that $\lambda_i > 0$. By Bertsimas and Tsitsiklis (1997, Theorem 3.2), $Q\boldsymbol{\lambda} = \tilde{\mathbf{x}}$ and $\boldsymbol{\lambda} \geq \mathbf{0}$. Since $\tilde{x}_{n+1} = 0$, $Q_{n+1,:} \geq \mathbf{0}^\intercal$, and $\lambda_i > 0$, we have $Q_{n+1,i} = 0$. By Line 12 and Assumption 2, there exists an $\mathbf{s} \in \mathcal{T}$ such that $\tilde{\mathbf{s}} = Q_{:,i}$. By Line 15, $M_{:,i} = Q_{:,i}$. By Lines 16 and 17, $\mathbf{s} \in \mathcal{S}$, proving Property 5.
6. We have already established that $Q\boldsymbol{\lambda} = \tilde{\mathbf{x}}$. By Lines 15–17 and Property 5, $\sum_{\mathbf{s}\in\mathcal{S}} \alpha_{\mathbf{s}}\tilde{\mathbf{s}} = M\boldsymbol{\lambda} = Q\lambda = \tilde{\mathbf{x}}$. Thus, Property 6 is satisfied.

Finally, we prove Properties 7–9 irrespective of whether the if (Line 1) or the else (Line 4) clauses are executed:

7. Let $i \in [n+2]$ such that $M_{n+1,i} = 0$. By Line 15, $\lambda_i > 0$. By Property 5, there exists an $\mathbf{s} \in \mathcal{S}$ such that $\tilde{\mathbf{s}} = M_{:,i}$. Thus, Property 7 is satisfied.

8. Let $\mathbf{s} \in \mathcal{S}$. By Line 17, $\alpha_{\mathbf{s}} > 0$. By Line 16, there exists an $i \in [n+2]$ such that $M_{:,i} = \tilde{\mathbf{s}}$. Thus, Property 8 is satisfied.

9. Let $\mathbf{s} \in \mathcal{S}$. By Line 17, $\alpha_{\mathbf{s}} > 0$. By Line 16, there exists an $i \in [n+2]$ such that $\tilde{\mathbf{s}} = M_{:,i}$ and $\lambda_i > 0$. By Property 5, $\mathbf{s} \in \mathcal{T}$. Thus, $\mathcal{S} \subseteq \mathcal{T}$. By Property 4, M is invertible. Thus, there exists an $i \in [n+2]$ such that $M_{n+1,i} \neq 0$. Thus, for all $\mathbf{s} \in \mathcal{S}$, $\tilde{\mathbf{s}} \neq M_{:,i}$. By Property 8, $|\mathcal{S}| \leq n+1$. Thus, Property 9 is satisfied.

□

Proof of Theorem 4.2. Properties 1–4 follow from Proposition 4.1 and induction. Since `PM` modifies the `CA` as described in Definition 3.1, `PM` conserves convex-combination-agnostic properties of the corresponding `CA`. □

Proof of Theorem 5.1. Let $t \in \{R, R+1, \ldots, T\}$ and consider the convex combination representing the current iterate $\mathbf{x}^{(t)} = \sum_{s \in \mathcal{S}^{(t)}} \alpha_{\mathbf{s}}^{(t)} \mathbf{s}$, where $\boldsymbol{\alpha}^{(t)} \in \Delta_{|\mathcal{V}|}$ and $\mathcal{S}^{(t)} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(t)} > 0\}$. Suppose towards a contradiction that $|\mathcal{S}^{(t)}| \geq \dim(\mathcal{C}^*) + 2$. By Theorem 4.2, $M^{(t)}$ is invertible for all $t \in \{0, 1 \ldots, T\}$. Thus, for any subset $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{\dim(\mathcal{C}^*)+2}\} \subseteq \mathcal{S}^{(t)}$, it has to hold that $\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, \ldots, \tilde{\mathbf{p}}_{\dim(\mathcal{C}^*)+2}$ are linearly independent. Since $\dim(\mathcal{C}^*)+2$ vectors of dimension $\dim(\mathcal{C}^*)+1$ cannot be linearly independent, we have a contradiction. □

Proof of Theorem 5.3. The statement of Theorem 5.3 without the convex-combination-agnostic property is a simplified version derived from Bomze et al. (2020, Theorem C.1). The proof of Theorem C.1 is a reduction to Bomze et al. (2020, Theorem 4.3) that does not utilize any arguments involving the active set of `AFW`. Furthermore, it can be verified that Bomze et al. (2020, Theorem 4.3) and all the results it builds upon are convex-combination-agnostic. Thus, the convex-combination-agnostic property of the statement in Theorem 5.3 is established. □

Proof of Corollary 5.4. The result follows by combining Theorem 5.1 with Theorem 5.3 for `AFW`, and with Theorem 9.4 for `BPFW`, and noting that $\mathbf{x}^{(t)} \in \mathcal{C}^*$ for $\mathcal{C}^* \in \text{faces}(\mathcal{C})$ implies $\mathcal{S}^{(t)} \subseteq \mathcal{C}^*$. □

8 `FW` variants

The away-step Frank-Wolfe algorithm (`AFW`) (Wolfe, 1970; Guélat and Marcotte, 1986; Lacoste-Julien and Jaggi, 2015) is presented in Algorithm 6. The blended pairwise Frank-Wolfe algorithm (`BPFW`) (Tsuji et al., 2022) is presented in Algorithm 7.

9 Active set identification with the blended pairwise Frank-Wolfe algorithm

In this section, we prove for `BPFW` the active set identification property introduced in Bomze et al. (2020) and proved for `AFW`. We consider in this section that the feasible set is the probability simplex Δ_n. Generalization to polytopes follows by affine invariance (Bomze et al., 2020, Appendix C). This means in particular that each vertex is the unit vector $\mathbf{e}_i$ corresponding to a coordinate $i \in [n]$. For convenience, we introduce the set $\mathcal{S}_I$ consisting of unit vectors corresponding to the coordinates for a given active set $\mathcal{S}$:

$$\mathcal{S}_I = \{i \in [n] \mid \mathbf{e}_i \in \mathcal{S}\}.$$

We first introduce necessary definitions. The multiplier associated with the ith coordinate at a point $\mathbf{x} \in \Delta_n$ is:

$$\lambda_i(\mathbf{x}) = \langle \nabla f(\mathbf{x}), \mathbf{e}_i - \mathbf{x} \rangle,$$

allowing the partition of the coordinates into

$$I(\mathbf{x}) = \{i \in [n] \mid \lambda_i(\mathbf{x}) = 0\} \qquad \text{and} \qquad I^c(\mathbf{x}) = [n] \setminus I(\mathbf{x}). \tag{9.1}$$

Algorithm 6: Away-step Frank-Wolfe algorithm (`AFW`) with line-search

Input: $\mathbf{x}^{(0)} \in \mathcal{V}$.
Output: $\boldsymbol{\alpha}^{(T)} \in \Delta_{|\mathcal{V}|}$, $\mathcal{S}^{(T)} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(T)} > 0\}$, and $\mathbf{x}^{(T)} \in \mathcal{C}$ such that $\mathbf{x}^{(T)} = \sum_{\mathbf{s} \in \mathcal{S}^{(T)}} \alpha_{\mathbf{s}}^{(T)} \mathbf{s}$.

$\boldsymbol{\alpha}^{(0)} \in \Delta_{|\mathcal{V}|}$ s.t. for all $\mathbf{s} \in \mathcal{V}$, $\alpha_{\mathbf{s}}^{(0)} \leftarrow \begin{cases} 1, & \text{if } \mathbf{s} = \mathbf{x}^{(0)} \\ 0, & \text{if } \mathbf{s} \in \mathcal{V} \setminus \{\mathbf{x}^{(0)}\} \end{cases}$
$\mathcal{S}^{(0)} \leftarrow \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(0)} > 0\} = \{\mathbf{x}^{(0)}\}$
for $t = 0, \ldots, T-1$ **do**
  $\mathbf{a}^{(t)} \leftarrow \operatorname{argmax}_{\mathbf{v} \in \mathcal{S}^{(t)}} \langle \nabla f(\mathbf{x}^{(t)}), \mathbf{v} \rangle$ ▷ `away vertex`
  $\mathbf{v}^{(t)} \leftarrow \operatorname{argmin}_{\mathbf{v} \in \mathcal{V}} \langle \nabla f(\mathbf{x}^{(t)}), \mathbf{v} \rangle$ ▷ `FW vertex`
  if $\langle \nabla f(\mathbf{x}^{(t)}), \mathbf{v}^{(t)} - \mathbf{x}^{(t)} \rangle \geq \langle \nabla f(\mathbf{x}^{(t)}), \mathbf{x}^{(t)} - \mathbf{a}^{(t)} \rangle$ **then**
    $\mathbf{d}^{(t)} \leftarrow \mathbf{x}^{(t)} - \mathbf{a}^{(t)}$ ▷ `away step`
    $\eta^{(t)} \leftarrow \operatorname{argmin}_{\eta \in [0, \alpha_{\mathbf{a}^{(t)}}^{(t)} / (1 - \alpha_{\mathbf{a}^{(t)}}^{(t)})]} f(\mathbf{x}^{(t)} + \eta \mathbf{d}^{(t)})$
    $\boldsymbol{\alpha}^{(t+1)} \in \Delta_{|\mathcal{V}|}$ s.t. for all $\mathbf{s} \in \mathcal{V}$, $\alpha_{\mathbf{s}}^{(t+1)} \leftarrow \begin{cases} (1 + \eta^{(t)}) \alpha_{\mathbf{s}}^{(t)}, & \text{if } \mathbf{s} \in \mathcal{V} \setminus \{\mathbf{a}^{(t)}\} \\ (1 + \eta^{(t)}) \alpha_{\mathbf{s}}^{(t)} - \eta^{(t)}, & \text{if } \mathbf{s} = \mathbf{a}^{(t)} \end{cases}$
  else
    $\mathbf{d}^{(t)} \leftarrow \mathbf{v}^{(t)} - \mathbf{x}^{(t)}$ ▷ `FW step`
    $\eta^{(t)} \leftarrow \operatorname{argmin}_{\eta \in [0,1]} f(\mathbf{x}^{(t)} + \eta \mathbf{d}^{(t)})$
    $\boldsymbol{\alpha}^{(t+1)} \in \Delta_{|\mathcal{V}|}$ s.t. for all $\mathbf{s} \in \mathcal{V}$, $\alpha_{\mathbf{s}}^{(t+1)} \leftarrow \begin{cases} (1 - \eta^{(t)}) \alpha_{\mathbf{s}}^{(t)}, & \text{if } \mathbf{s} \in \mathcal{V} \setminus \{\mathbf{v}^{(t)}\} \\ (1 - \eta^{(t)}) \alpha_{\mathbf{s}}^{(t)} + \eta^{(t)}, & \text{if } \mathbf{s} = \mathbf{v}^{(t)} \end{cases}$
  end
  $\mathcal{S}^{(t+1)} \leftarrow \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(t+1)} > 0\}$
  $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} + \eta^{(t)} \mathbf{d}^{(t)}$
end

Given a stationary point of the problem $\mathbf{x}^*$ and iterates $\mathbf{x}^{(t)}$ generated by `BPFW`, we also define:

$$\begin{aligned} O^{(t)} &= \{i \in I^c(\mathbf{x}^*) \mid \mathbf{x}_i^{(t)} = 0\}, \\ J^{(t)} &= I^c(\mathbf{x}^*) \setminus O^{(t)}, \\ \delta^{(t)} &= \max_{i \in [n] \setminus O^{(t)}} \lambda_i(\mathbf{x}^*), \\ \delta_{\min} &= \min_{i \in I^c(\mathbf{x}^*)} \lambda_i(\mathbf{x}^*), \\ h^* &= \frac{\delta_{\min}}{3L + \max\{\delta^{(t)}, \delta_{\min}\}}. \end{aligned} \tag{9.2}$$

By complementary slackness, $\mathbf{x}_i^* = 0$ for all $i \in I^c(\mathbf{x}^*)$. We now introduce a first lemma restating the conditions for the selection of the type of step in `BPFW` in terms of multipliers.

Lemma 9.1 (`BPFW` step multipliers)**.** *Let $\mathcal{C} = \Delta_n \subseteq \mathbb{R}^n$ be the probability simplex and let $f \colon \mathcal{C} \to \mathbb{R}$ be a convex and L-smooth function. Then, for the iterates of Algorithm 7 (`BPFW`) with line-search and all $t \in \{0, 1, \ldots, T-1\}$, the following hold:*

1. *If $\max_{j \in \mathcal{S}_I^{(t)}} \lambda_j(\mathbf{x}^{(t)}) - \min_{i \in \mathcal{S}_I^{(t)}} \lambda_i(\mathbf{x}^{(t)}) \geq -\min_{i \in [n]} \lambda_i(\mathbf{x}^{(t)})$, then `BPFW` performs a pairwise step, with $\mathbf{d}^{(t)} = \mathbf{e}_i - \mathbf{e}_j$ for $i \in \operatorname{argmin}_{i \in \mathcal{S}_I^{(t)}} \lambda_i(\mathbf{x}^{(t)})$ and $j \in \operatorname{argmax}_{j \in \mathcal{S}_I^{(t)}} \lambda_j(\mathbf{x}^{(t)})$.*
2. *For every $j \in [n] \setminus \mathcal{S}_I^{(t)}$, if $\lambda_j(\mathbf{x}^{(t)}) > 0$, then $x_j^{(t+1)} = x_j^{(t)} = 0$.*

Proof. The first statement follows from the algorithm and the fact that the ith multiplier corresponds to the selection criterion for the away and `FW` vertices. For the second statement, note that a vertex not in the current

Algorithm 7: Blended pairwise Frank-Wolfe algorithm (`BPFW`) with line-search

Input: $\mathbf{x}^{(0)} \in \mathcal{V}$.
Output: $\boldsymbol{\alpha}^{(T)} \in \Delta_{|\mathcal{V}|}$, $\mathcal{S}^{(T)} = \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(T)} > 0\}$, and $\mathbf{x}^{(T)} \in \mathcal{C}$, such that $\mathbf{x}^{(T)} = \sum_{\mathbf{s} \in \mathcal{S}^{(T)}} \alpha_{\mathbf{s}}^{(T)} \mathbf{s}$.

$\boldsymbol{\alpha}^{(0)} \in \Delta_{|\mathcal{V}|}$ s.t. for all $\mathbf{s} \in \mathcal{V}$, $\alpha_{\mathbf{s}}^{(0)} \leftarrow \begin{cases} 1, & \text{if } \mathbf{s} = \mathbf{x}^{(0)} \\ 0, & \text{if } \mathbf{s} \in \mathcal{V} \setminus \{\mathbf{x}^{(0)}\} \end{cases}$
$\mathcal{S}^{(0)} \leftarrow \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(0)} > 0\} = \{\mathbf{x}^{(0)}\}$
for $t = 0, \ldots, T-1$ **do**
  $\mathbf{a}^{(t)} \leftarrow \operatorname{argmax}_{\mathbf{v} \in \mathcal{S}^{(t)}} \langle \nabla f(\mathbf{x}^{(t)}), \mathbf{v} \rangle$ ▷ `away vertex`
  $\mathbf{w}^{(t)} \leftarrow \operatorname{argmin}_{\mathbf{v} \in \mathcal{S}^{(t)}} \langle \nabla f(\mathbf{x}^{(t)}), \mathbf{v} \rangle$ ▷ `local FW vertex`
  $\mathbf{v}^{(t)} \leftarrow \operatorname{argmin}_{\mathbf{v} \in \mathcal{V}} \langle \nabla f(\mathbf{x}^{(t)}), \mathbf{v} \rangle$ ▷ `FW vertex`
  if $\langle \nabla f(\mathbf{x}^{(t)}), \mathbf{v}^{(t)} - \mathbf{x}^{(t)} \rangle \geq \langle \nabla f(\mathbf{x}^{(t)}), \mathbf{w}^{(t)} - \mathbf{a}^{(t)} \rangle$ **then**
    $\mathbf{d}^{(t)} \leftarrow \mathbf{w}^{(t)} - \mathbf{a}^{(t)}$ ▷ `local pairwise step`
    $\eta^{(t)} \leftarrow \operatorname{argmin}_{\eta \in [0, \alpha_{\mathbf{a}^{(t)}}^{(t)}]} f(\mathbf{x}^{(t)} + \eta \mathbf{d}^{(t)})$
    $\boldsymbol{\alpha}^{(t+1)} \in \Delta_{|\mathcal{V}|}$ s.t. for all $\mathbf{s} \in \mathcal{V}$, $\alpha_{\mathbf{s}}^{(t+1)} \leftarrow \begin{cases} \alpha_{\mathbf{s}}^{(t)}, & \text{if } \mathbf{s} \in \mathcal{V} \setminus \{\mathbf{a}^{(t)}, \mathbf{w}^{(t)}\} \\ \alpha_{\mathbf{s}}^{(t)} - \eta^{(t)}, & \text{if } \mathbf{s} = \mathbf{a}^{(t)} \\ \alpha_{\mathbf{s}}^{(t)} + \eta^{(t)}, & \text{if } \mathbf{s} = \mathbf{w}^{(t)} \end{cases}$
  else
    $\mathbf{d}^{(t)} \leftarrow \mathbf{v}^{(t)} - \mathbf{x}^{(t)}$ ▷ `FW step`
    $\eta^{(t)} \leftarrow \operatorname{argmin}_{\eta \in [0,1]} f(\mathbf{x}^{(t)} + \eta \mathbf{d}^{(t)})$
    $\boldsymbol{\alpha}^{(t+1)} \in \Delta_{|\mathcal{V}|}$ s.t. for all $\mathbf{s} \in \mathcal{V}$, $\alpha_{\mathbf{s}}^{(t+1)} \leftarrow \begin{cases} (1-\eta^{(t)})\alpha_{\mathbf{s}}^{(t)}, & \text{if } \mathbf{s} \in \mathcal{V} \setminus \{\mathbf{v}^{(t)}\} \\ (1-\eta^{(t)})\alpha_{\mathbf{s}}^{(t)} + \eta^{(t)}, & \text{if } \mathbf{s} = \mathbf{v}^{(t)} \end{cases}$
  end
  $\mathcal{S}^{(t+1)} \leftarrow \{\mathbf{s} \in \mathcal{V} \mid \alpha_{\mathbf{s}}^{(t+1)} > 0\}$
  $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} + \eta^{(t)} \mathbf{d}^{(t)}$
end

active set cannot be selected as a `FW` or away vertex in a pairwise step. Furthermore, the `FW` gap can be expressed as

$$g(\mathbf{x}) = -\min_{i \in [n]} \lambda_i(\mathbf{x}) \geq 0.$$

Thus, there always exists a coordinate i with $\lambda_i(\mathbf{x}) \leq 0$. If for $j \in [n]$, $\lambda_j(\mathbf{x}^{(t)}) > 0$, then the vertex $\mathbf{e}_j$ would not be selected as the vertex in a `FW` step either. □

Next, we derive upper and lower bounds on the multipliers.

Lemma 9.2 (Bounds on the multipliers)**.** *Let $\mathcal{C} = \Delta_n \subseteq \mathbb{R}^n$ be the probability simplex, let $f \colon \mathcal{C} \to \mathbb{R}$ be a convex and L-smooth function, and let $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x})$. Then, for the iterates of Algorithm 7 (`BPFW`) with line-search and all $t \in \{0, 1, \ldots, T-1\}$ such that $\|\mathbf{x}^* - \mathbf{x}^{(t)}\|_1 < h^*$, it holds that:*

$$\lambda_i(\mathbf{x}^{(t)}) < h^* \left(L + \frac{\delta^{(t)}}{2} \right) \qquad \text{for all } i \in I(\mathbf{x}^*) \tag{9.3}$$

$$\lambda_j(\mathbf{x}^{(t)}) > -h^* \left(L + \frac{\delta^{(t)}}{2} \right) + \delta_{\min} \qquad \text{for all } j \in I^c(\mathbf{x}^*). \tag{9.4}$$

Proof. Bomze et al. (2020, Lemma 3.1) proves that $|\lambda_i(\mathbf{x}^*) - \lambda_i(\mathbf{x}^{(t)})| < h^* \left(L + \frac{\delta^{(t)}}{2} \right)$ for any $i \in [n]$ and iteration t where the assumptions of the Lemma hold. The two subsequent inequalities follow from the definitions of $I(\mathbf{x}^*)$

in (9.1) and $\delta_{\min}$ in (9.2). In particular, Equation (9.4) can be obtained as follows:

$$\lambda_j(\mathbf{x}^{(t)}) > \lambda_j(\mathbf{x}^*) - h^* \left(L + \frac{\delta^{(t)}}{2}\right) \geq \min_{i \in I^c(\mathbf{x}^*)} \lambda_i(\mathbf{x}^*) - h^* \left(L + \frac{\delta^{(t)}}{2}\right) = \delta_{\min} - h^* \left(L + \frac{\delta^{(t)}}{2}\right).$$

□

In order to prove the active set identification property, we need a lower bound assumption on the step size. This assumption allows us in particular to prove that, once the iterate is close enough to a stationary point, a local pairwise step is always maximal and reduces the weight of the away vertex down to zero.

Assumption 1 (Step-size lower bound)**.** *During the execution of Algorithm 7 (*BPFW*) and any iteration $t \in \{0, 1, \ldots, T-1\}$ where a pairwise step is taken with $\mathbf{a}^{(t)} \in \mathcal{S}^{(t)}$ the away vertex, we assume that the following lower bound holds on the step size:*

$$\eta^{(t)} \geq \min \left\{ \alpha^{(t)}_{\mathbf{a}^{(t)}}, \frac{\langle \nabla f(\mathbf{x}^{(t)}), -\mathbf{d}^{(t)} \rangle}{L \|\mathbf{d}^{(t)}\|_2^2} \right\}.$$

Assumption 1 holds in particular for step-sizes computed with line-search and short-step rules Bomze et al. (2020, Remark 4.4). The authors also show that

$$\eta^{(t)} \geq \min \left\{ \alpha^{(t)}_{\mathbf{a}^{(t)}}, c \frac{\langle \nabla f(\mathbf{x}^{(t)}), -\mathbf{d}^{(t)} \rangle}{L \|\mathbf{d}^{(t)}\|_2^2} \right\} \tag{9.5}$$

for a given $c > 0$ is a sufficient condition for Assumption 1 to hold (with a modified Lipschitz constant L/c). This means in particular that the commonly-used adaptive step-size strategy introduced in Pedregosa et al. (2018) respects the weaker condition (9.5), with the factor c corresponding to the parameter τ of the backtracking subroutine Pedregosa et al. (2018, Algorithm 2).

We can now state Theorem 9.3, which proves the active set identification for BPFW. The proof follows the structure of that of Bomze et al. (2020, Theorem 3.3), where the same property is shown for AFW.

Theorem 9.3 (BPFW identification)**.** *Let $\mathcal{C} = \Delta_n \subseteq \mathbb{R}^n$ be the probability simplex, let $f \colon \mathcal{C} \to \mathbb{R}$ be a convex and L-smooth function, and let $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x} \in \mathcal{C}} f(\mathbf{x})$. Then, for the iterates of Algorithm 7 (*BPFW*) with line-search and all $t \in \{0, 1, \ldots, T-1\}$ such that $\|\mathbf{x}^{(t)} - \mathbf{x}^*\|_1 < h^*$, it holds that $|J^{(t+1)}| \leq \max\{0, |J^{(t)}| - 1\}$.*

Proof. If $I^c(\mathbf{x}^*) = \emptyset$, $J^{(t)} \subseteq I^c(\mathbf{x}^*) = \emptyset$ and $J^{(t+1)} \subseteq I^c(\mathbf{x}^*) = \emptyset$. We next assume $|I^c(\mathbf{x}^*)| > 0$.

We first consider the case in which $J^{(t)} = \emptyset$. Then, $[n] \setminus O^{(t)} = I(\mathbf{x}^*)$ and $\delta^{(t)} = 0$ since $\lambda_i(\mathbf{x}^*) = 0$ for all $i \in I(\mathbf{x}^*)$. Inequality (9.4) becomes:

$$\lambda_j(\mathbf{x}^{(t)}) > -h^* L + \delta_{\min} = \delta_{\min} - \frac{L \delta_{\min}}{3L + \delta_{\min}} > 0 \qquad \text{for all } j \in I^c(\mathbf{x}^*) = O^{(t)}.$$

Since $J^{(t)} = \emptyset$, it holds that $I^c(\mathbf{x}^*) = O^{(t)}$ and $x_j^{(t)} = 0$ for all $j \in I^c(\mathbf{x}^*)$. Following Lemma 9.1, Property 2, we conclude that $x_j^{(t+1)} = 0$ for all $j \in I^c(\mathbf{x}^*)$. Thus, $J^{(t+1)} = \emptyset$.

We now consider the case when $|J^{(t)}| > 0$ and first prove that the FW vertex i and the away vertex j are such that $i \in I(\mathbf{x}^*)$ and $j \in J^{(t)}$. Recall that $\mathcal{S}_I^{(t)} \subseteq [n] \setminus O^{(t)} = J^{(t)} \cup I(\mathbf{x}^*)$ and that both the FW and away vertex are selected from $\mathcal{S}_I^{(t)}$ in a pairwise step. Bomze et al. (2020, Theorem 3.3, Case 2) proves that if $|J^{(t)}| > 0$, then the away vertex is always selected in $J^{(t)}$ if

$$\|\mathbf{x}^{(t)} - \mathbf{x}^*\|_1 < \frac{\delta_{\min}}{2L + \delta_{\min}},$$

which holds by the definition of h^* since

$$\|\mathbf{x}^{(t)} - \mathbf{x}^*\|_1 < h^* \leq \frac{\delta_{\min}}{3L + \delta_{\min}} \leq \frac{\delta_{\min}}{2L + \delta_{\min}}.$$

We next demonstrate that the FW vertex is always selected in $I(\mathbf{x}^*)$ and not in $J^{(t)}$. Based on (9.3) and (9.4), a sufficient condition for the FW vertex to be selected in $I(\mathbf{x}^*)$ is:

$$\lambda_i(\mathbf{x}^{(t)}) < \lambda_j(\mathbf{x}^{(t)}) \text{ for all } i \in I(\mathbf{x}^*), j \in J^{(t)} \qquad \Leftrightarrow \qquad h^* < \frac{\delta_{\min}}{2L + \delta^{(t)}},$$

which follows from the definition of h^*. Finally, we need to prove that the pairwise step is maximal such that the away vertex is dropped. That is, given the FW and away vertices $i \in I(\mathbf{x}^*)$ and $j \in J^{(t)}$, respectively, the step-size is $\eta^{(t)} = \alpha^{(t)}_{\mathbf{e}_j}$, the weight of the away vertex $\mathbf{e}_j$ in the active set. Let us assume by contradiction that the step is not maximal, that is, $\eta^{(t)} < \alpha^{(t)}_{\mathbf{e}_j}$. Furthermore, recall that if a pairwise step is performed at iteration t, then $\mathbf{d}^{(t)} = \mathbf{e}_i - \mathbf{e}_j$ for two indices $i \neq j$, yielding $\|\mathbf{d}^{(t)}\|_2^2 = 2$. Then, by Assumption 1,

$$\begin{aligned}
\eta^{(t)} &\geq \frac{\langle \nabla f(\mathbf{x}^{(t)}), -\mathbf{d}^{(t)} \rangle}{L\|\mathbf{d}^{(t)}\|_2^2} \\
&= \frac{\lambda_j(\mathbf{x}^{(t)}) - \lambda_i(\mathbf{x}^{(t)})}{2L} \\
&\geq \frac{\delta_{\min} - h^*(2L + \delta^{(t)})}{2L} && \triangleright \text{ by subtracting (9.3) from (9.4)} \\
&= \frac{\delta_{\min}}{2L} - h^* - \frac{h^*\delta^{(t)}}{2L}.
\end{aligned}$$

Furthermore, since $\mathbf{x}^{(t)}$ and $\mathbf{x}^*$ lie in the simplex, by Bomze et al. (2020, Lemma A.1, Property 2), and the fact that $x^*_j = 0$ for all $j \in I^c(\mathbf{x}^*)$ by complementary slackness, we have:

$$x^{(t)}_j = |x^{(t)}_j - x^*_j| \leq \frac{\|\mathbf{x}^{(t)} - \mathbf{x}^*\|_1}{2} < \frac{h^*}{2}.$$

Together with the fact that $d^{(t)}_j = -1$ since j is the coordinate corresponding to the away vertex, we have:

$$\begin{aligned}
x^{(t+1)}_j &= x^{(t)}_j + \eta^{(t)} d^{(t)}_j \\
&< \frac{h^*}{2} - \frac{\delta_{\min}}{2L} + h^* + \frac{h^*\delta^{(t)}}{2L} \\
&= \frac{3Lh^*}{2L} - \frac{\delta_{\min}}{2L} + \frac{h^*\delta^{(t)}}{2L} \\
&= \frac{1}{2L}((3L + \delta^{(t)})h^* - \delta_{\min}) < 0,
\end{aligned}$$

a contradiction. We conclude that the step size is maximal. □

Theorem 9.3 can be used to generalize Bomze et al. (2020, Theorem 4.3) to BPFW under similar assumptions, meaning that BPFW identifies the active set in a finite number of iterations, since the proof of Bomze et al. (2020, Theorem 4.3) does not rely on the sequence generated by the algorithm. We also note that our proof does not extend to the pairwise Frank-Wolfe algorithm (Lacoste-Julien and Jaggi, 2015), since we rely on the BPFW property that at any iteration in which a pairwise step is performed, both the away and the FW vertex are selected from the active set.

Theorem 9.4 (Active set identification property of BPFW)**.** *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a polytope, let $f\colon \mathcal{C} \to \mathbb{R}$ be a convex and L-smooth function, and suppose that $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{C}} f(\mathbf{x})$ is unique and $\mathbf{x}^* \in \operatorname{rel.int}(\mathcal{C}^*)$, where $\mathcal{C}^* \in \operatorname{faces}(\mathcal{C})$. Then, for T large enough, for the iterations of Algorithm 7 (BPFW) with line-search, there exists an iteration $R \in \{0, 1, \ldots, T\}$ such that $\mathbf{x}^{(t)} \in \mathcal{C}^*$ for all $t \in \{R, R+1, \ldots, T\}$. This property is convex-combination-agnostic.*

Proof. By affine invariance of BPFW with line-search, this is equivalent to optimizing $f(A\mathbf{y})$, with $\mathbf{y} \in \Delta_n$, as developed in Bomze et al. (2020, Appendix C). The proof over the simplex follows that of Bomze et al. (2020, Theorem 4.3), relying on Theorem 9.3 to ensure the cardinality of the set $J^{(t)}$ decreases at each iteration once there exists a stationary point $\mathbf{x}^*$ of the problem with $\mathbf{x}^* \in \mathcal{X}$, such that $\|\mathbf{x}^{(t)} - \mathbf{x}^*\|_1 < h^*$. □

10 Algorithmic aspects

The computationally-demanding part of `PM` – compared to the baseline `FW` variants it builds on – amounts to solving the linear system $(N^{(t)})^{-1}\mathbf{v}$ in `FWU` when a new vertex is introduced. All modifications of $M^{(t)}$ throughout `PM`'s execution consist of column additions and substitutions, one could therefore construct an initial sparse LU decomposition of $M^{(t)}$ and perform rank-one updates (Gill et al., 1987; Huangfu and Hall, 2015). Preliminary experiments using the `BasicLU` sparse LU library (Schork and Gondzio, 2017) showed that the numerical accuracy was too low for our purpose, resulting in weights λ that presented a large iterate reconstruction error. We resorted to direct sparse LU solves instead whenever a new vertex is inserted in the matrix without maintaining an LU update. Furthermore, we compensate for inaccuracy of the linear solver by projecting back the weights λ onto Δ_{n+2} using the algorithm proposed in Condat (2016). As observed in the experiments, the per-iteration overhead caused by the pivoting operations is not deterrent enough to make the meta algorithm, even with this direct implementation, impractical. In future work, an extended precision version, along with a control of error tolerances, will allow the integration of rank-one update steps avoiding full factorizations. Leveraging higher-precision arithmetic will also allow us to exploit the pivoting framework for feasible regions yielding denser vertices and numerically challenging sparse linear systems to solve, such as the instance solved the K-sparse polytope.

A typical case where the conditioning of the matrix could degrade quickly occurred with K-sparse polytope instances with large enough K and τ values. One vertex $\mathbf{v}$ and its opposite $\mathbf{v}_n = -\mathbf{v}$ could be added to the active set, resulting in two columns such that:

$$\frac{\langle \tilde{\mathbf{v}}, \tilde{\mathbf{v}}_n \rangle}{\|\mathbf{v}\|^2} = \frac{1 - K\tau^2}{1 + K\tau^2} = -1 + \frac{2}{K\tau^2 + 1} \approx -1.$$

As more almost-colinear pairs of columns are added, the matrix M is closer to becoming singular, which may lead to imprecise or impossible linear system solves. The degradation in numerical accuracy is especially pronounced on problems yielding denser vertices.

In contrast, the operations performed on the active set weights in `FW` variants are more robust to numerical accuracy, consisting only of increasing and decreasing individual weights or scaling the whole weight vector. The availability of an extended-precision sparse linear solver supporting rank-one updates will directly benefit implementations of our algorithms.

We also highlight that a key characteristic of `PM` is that it maintains only the vertices required to represent the current iterate. One could solve the LPs over the final vertices to obtain a basic solution ensuring the Carathéodory upper bound on the cardinality, but it would require a constraint matrix including all vertices of the final active set, which can be arbitrarily larger than $n + 2$ during the solving process.

Finally, we emphasize that `PM` works with all algorithms leveraging convex combinations of vertices as long as their convergence respects the convex-combination-agnostic property introduced in Definition 3.1. A promising avenue for future research will be leveraging the pivoting framework introduced in this paper in the context of stochastic gradient estimators, see Braun et al. (2022, Chapter 4). The convergence of the stochastic Frank-Wolfe with and without momentum presented in the survey is convex combination agnostic, see Braun et al. (2022, Theorems 4.7, 4.8). While large-scale settings could be detrimental to storing the vertices as an active set, stochastic variants of `FW` algorithms can be leveraged for machine learning problems where the objective is a finite sum of loss terms evaluated over a large number of samples. In such a setting, the pivoting framework can maintain a sparse vertex decomposition of the iterates while maintaining the convergence guarantees of the underlying algorithm. Away and Pairwise `FW` variants have been proposed in the stochastic setting in Goldfarb et al. (2017). We note that the convergence rate is not convex-combination agnostic, since the number of vertices in the active set appears in a coefficient of the expected primal bound decrease rate, leading the authors to motivate a Carathéodory procedure after the active set update to bound the number of vertices, and thus improving their convergence rate, making a strong argument for the pivoting framework we propose.

11 Numerical experiments (extended)

We assess our algorithm on efficiency in terms of function value and FW gap convergence, and sparsity of the obtained solutions, compared to the standard and lazified versions (that is, using the active set vertices as a

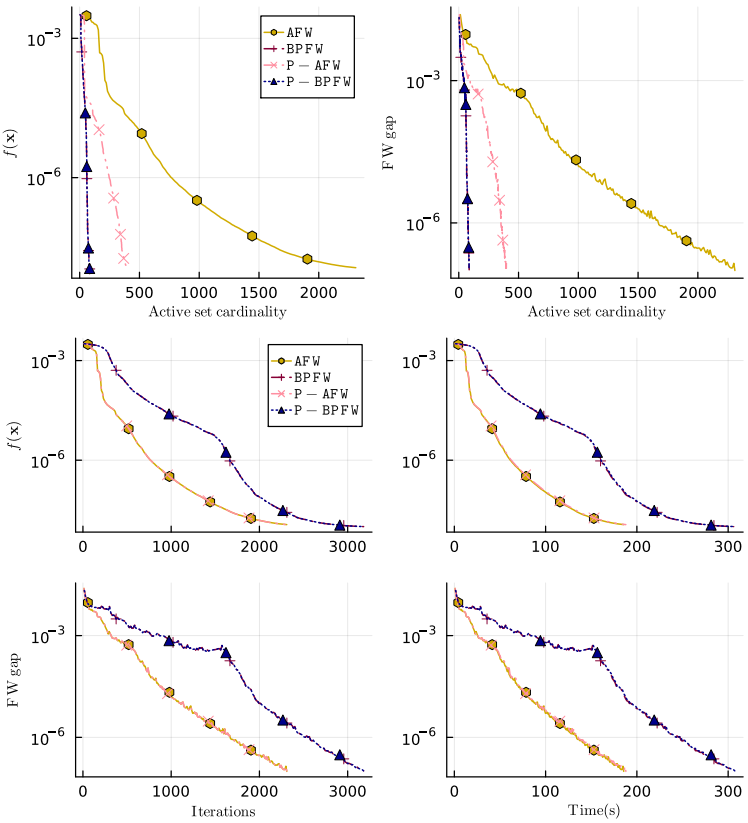

Figure 2: Logistic regression, $\tau = 60$.

weak separation oracle (Braun et al., 2017)) of `AFW` and `BPFW`, which also produce notably sparse solutions. The implementation of further `PM` variants is left for future research. Our algorithm is implemented in Julia (Bezanson et al., 2017) v1.9.2 and builds on the `FrankWolfe.jl` package (Besançon et al., 2022). The code for the algorithm and experiments is available on the repository github.com/ZIB-IOL/pivoting_frankwolfe.

The sparse linear systems are solved with the LU decomposition of the `UMFPACK` library (Davis, 2004). All computations are executed on a cluster with PowerEdge R650 nodes equipped with Intel Xeon Gold 6342 2.80GHx CPUs and 512GB RAM in exclusive mode. Plots are log-linear, function values are shifted so that the minimum on each plot reaches 10^{-8}. The prefix `L-` is used to denote the lazified version of an algorithm, the prefix `P-` for the `PM` variant.

11.1 Sparse logistic regression

We run all algorithms on a logistic regression problem with an ℓ_1-norm ball constraint:

$$\min_{\mathbf{x}} \frac{1}{m} \sum_{i=1}^{m} \log(1 + \exp(-y_i \mathbf{a}_i^\intercal \mathbf{x})$$
$$\text{s.t. } \|\mathbf{x}\|_1 \leq \tau,$$

where m is the number of samples, $\tau > 0$ is the ℓ_1-norm ball radius, $y_i \in \{-1, 1\}$ encodes the class and $\mathbf{a}_i$ the feature vector for the ith sample. We use the Gisette dataset (Guyon et al., 2004), which contains 5000 features, we run logistic regression on the validation set containing only 1000 samples and thus more prone to overfitting without a sparsity-inducing regularization constraint. The convergence of the pivoting variants of both `AFW` and `BPFW` converge similarly in both function value and FW gap as their standard counterparts as shown in Figure 2. Even though `BPFW` typically maintains a smaller active set, it converges at a slower rate than the away-step `FW` variants, both in function value and FW gap. `P-AFW` drastically improves the sparsity of the `AFW` iterates, while

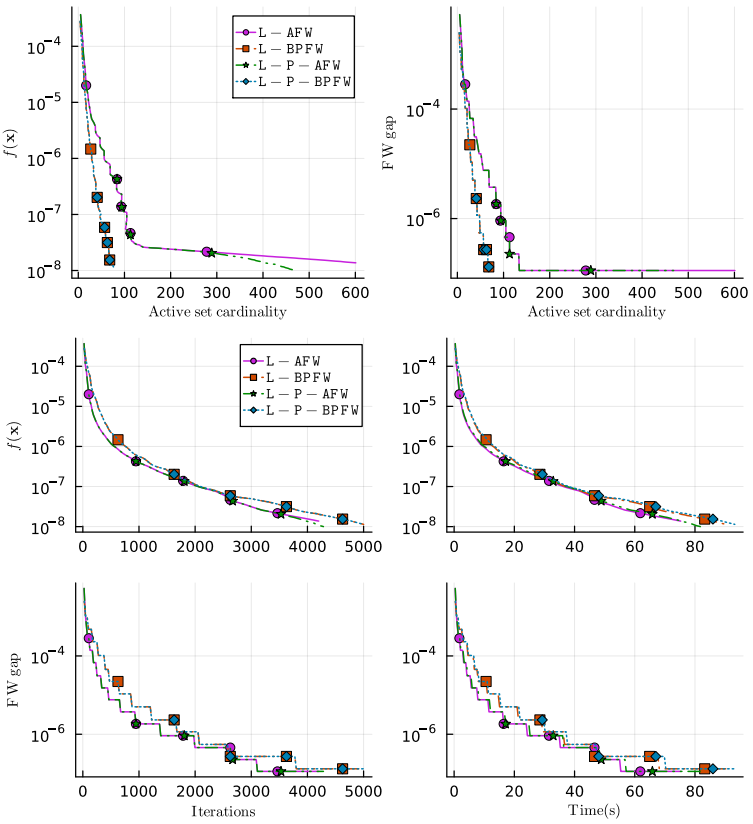

Figure 3: K-sparse logistic regression, $K = 10$ and $\tau = 1.0$.

maintaining the same convergence in function value and FW gap. This highlights one key property of our meta algorithm: it can be adapted to several `FW` variants, benefiting from their convergence rate while improving their sparsity.

We also fit logistic regression models on the K-sparse polytope (Cai and Zhang, 2013):

$$\mathcal{K}_{K,\tau} = \{\mathbf{x} \in \mathbb{R} \mid \|\mathbf{x}\|_1 \leq K\tau, \|\mathbf{x}\|_\infty \leq \tau\} = \operatorname{conv}(\{\mathbf{x} \in \mathbb{R} \mid \|\mathbf{x}\|_0 \leq K, |x_i| \in \{0, \tau\}\}),$$

the convex hull of vectors with at most K non-zero entries with values in $\{-\tau, \tau\}$ for a given radius $\tau > 0$.

The relative behavior of the lazified version of the algorithms is illustrated on the K-sparse polytope in Figure 3. The lazified `AFW` variants converge faster in both function value and FW gap than the `BPFW` variants, although the latter maintain much sparser iterates. `L-P-AFW` reduces the cardinality of the active set, especially when approaching the optimum.

We also study and compare the effect of the pivoting meta algorithm on the lazified and non-lazified version of `AFW`, illustrated on another regression on the K-sparse polytope in Figure 4. On this instance, the pivoting technique applied to the lazified `AFW` algorithm only slightly improves the active set cardinality and does not change convergence. Pivoting however sharply reduces the cardinality of the active set for the non-lazified variant. This however comes at the expense of a higher per-iteration runtime and higher numerical instabilities which affect in particular the FW gap convergence. Such artifacts occur when the determinant of the matrix M increases with new vertices, and the phenomenon is particularly pronounced for non-lazified variants which typically introduce more vertices. We discuss numerical robustness and stability in more detail in Subsection 10.

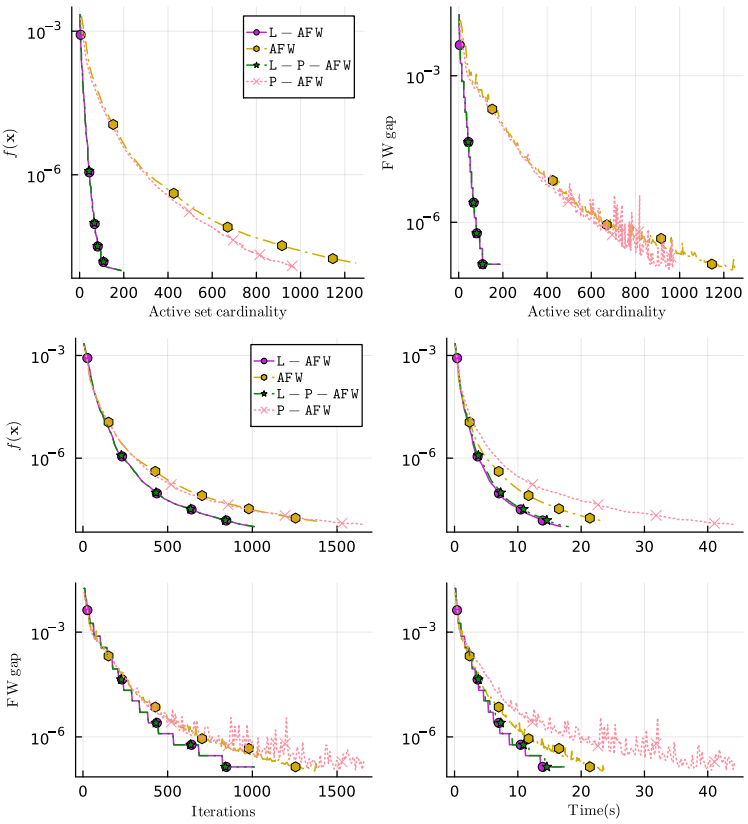

Figure 4: K-sparse logistic regression on the lazified and non-lazified `AFW`, $K = 10$ and $\tau = 4$.

11.2 Sparse signal recovery

We assess our algorithm on a sparse signal recovery problem:

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{y}\|_2^2$$
$$\text{s.t. } \|\mathbf{x}\|_1 \leq \tau,$$

with $A \in \mathbb{R}^{m \times n}$, $\mathbf{y} \in \mathbb{R}^m$, $n > m$. We generate the entries of the sensing matrix A i.i.d. from a standard Gaussian distribution and $\mathbf{y}$ by adding Gaussian noise with unit standard deviation to $A\mathbf{x}_{\text{true}}$, with $\mathbf{x}_{\text{true}}$ an underlying sparse vector, with 30% of non-zero terms, all taking entries sampled from a standard Gaussian distribution. The radius τ is computed as $\tau = \|\mathbf{x}_{\text{true}}\|_1/\tau_f$ for different values of τ_f.

Figure 5 illustrates the results of the non-lazified version of `BPFW` and `AFW` and their pivoting counterparts. `P-AFW` converges at the same rate as `AFW` in function value and FW gap while being faster than both `BPFW` variants, terminating before them, while maintaining an active set twice as sparse as `AFW`.

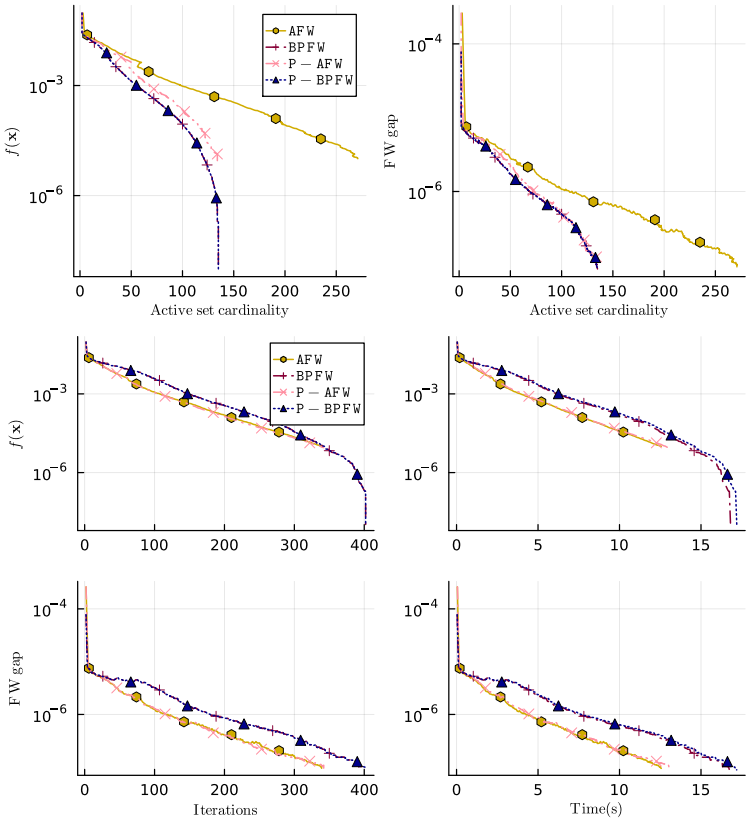

Figure 5: Signal recovery, $\tau_f = 20$, $m = 6000$, $n = 14000$. All variants are non-lazified.