On a Frank-Wolfe Approach for Abs-smooth Functions

Timo Kreimeier, Sebastian Pokutta,
Andrea Walther, and Zev Woodstock

March 7, 2023

Abstract

We propose an algorithm which appears to be the first bridge between the fields of conditional gradient methods and abs-smooth optimization. Our nonsmooth nonconvex problem setting is motivated by machine learning, since the broad class of abs-smooth functions includes, for instance, the squared ℓ_2-error of a neural network with ReLU or hinge Loss activation. To overcome the nonsmoothness in our problem, we propose a generalization to the traditional Frank-Wolfe gap and prove that first-order minimality is achieved when it vanishes. We derive a convergence rate for our algorithm which is *identical* to the smooth case. Although our algorithm necessitates the solution of a subproblem which is more challenging than the smooth case, we provide an efficient numerical method for its partial solution, and we identify several applications where our approach fully solves the subproblem. Numerical and theoretical convergence is demonstrated, yielding several conjectures.

Keywords: Frank-Wolfe algorithm, Active Signature Method, abs-smooth functions, nonsmooth optimization, convergence rate

1 Introduction

Many applications, see, e.g., [29, 38], involve the minimization of a function $f\colon \mathbb{R}^n \to \mathbb{R}$ subject to a compact and convex constraint $C \subset \mathbb{R}^n$. That is, one has to solve problems of the form

$$\min_{x \in C} \; f(x)\,. \tag{1}$$

We address (1) for the case when f is an *abs-smooth* function (see Definition 1.1). In a nutshell, the class of abs-smooth functions captures all nonsmooth functions whose nondifferentiability arise as a result of the absolute value function. Hence, this class also includes smooth functions, max, min, and compositions/linear combinations thereof. Since functions remain in this class when used recursively [16], one can readily show that many important objective functions from machine learning, e.g., the squared ℓ^2-loss of a neural network possessing the ReLU or hinge loss activation, reside in this class.

It was shown in [16] that, for abs-smooth functions, one can generate local piecewise linear models with approximation properties similar in quality to a Taylor expansion up to second order; furthermore, all required information to define these approximants are derivatives in the classical sense and can be computed easily by an extended version of algorithmic differentiation (AD) [18]. By considering this subclass of nonsmooth nonconvex objectives, efficient optimization algorithms with guarantees such as first-order stationarity have been pioneered in the last decade [11, 16, 17, 19, 20, 39]. However, it appears to be an open question as to whether or not one can enforce closed convex constraints in an abs-smooth optimization routine. In fact, even if one restricts to the subclass of *piecewise linear* objective functions, it is currently only possible to enforce piecewise linear constraints [25].

To address this gap, we will draw upon the theory of Frank-Wolfe (or *conditional gradient*) algorithms. In contrast to more computationally-intensive methods which enforce the constraint C by evaluating proximity operators or projections, the Frank-Wolfe algorithm, see, e.g., [7, 12, 27], only needs to solve a linear optimization subproblem over C. In traditional settings, the objective function f is assumed to be smooth,

i.e., Lipschitz-continuously differentiable, and the *Linear Minimization Oracle* (LMO) computes for the gradient $c \equiv \nabla f(\bar{x}) \in \mathbb{R}^n$ at a current iterate $\bar{x} \in \mathbb{R}^n$, a point in $\arg\min_{v \in C} \langle c, v \rangle$. The Frank-Wolfe algorithm and its variants have gained popularity because this linear minimization often requires fewer numerical computations when compared to projection-based methods. For instance, computing a projection onto the spectrahedron $C = \{x \in \mathbb{S}^n_+ \mid \mathrm{Tr}(x) = 1\}$ requires a full eigendecomposition; on the other hand, linear minimization over C only requires computing one dominant eigenpair [14].

Frank-Wolfe approaches have been extensively studied for the smooth case and various results are available for a myriad of settings [8, 10, 12, 23]; see also [7] for an overview. However, despite their significant computational advantages, to-date, conditional gradient algorithms have rarely been studied outside of the smooth setting. The approach proposed in [32] extends Frank-Wolfe methods to cover objective functions with continuous, albeit non-Lipschitz, gradients. However, for our applications, the target function f is not differentiable at all. A typical approach to overcome this problem is to repeatedly minimize smoothed approximations of the objective function [31]. Then, the resulting algorithm essentially encodes a proximity-type operator, which (as demonstrated above) can be more costly than an LMO. Furthermore, as a smoothed version of a nonsmooth function grows in fidelity to the original, typically the Lipschitz constant of its gradient grows arbitrarily large, so smooth optimization methods, whose rates often depend on this smoothness constant, can exhibit poor behavior in practice. Another approach to the nonsmooth setting relies on the ability to compute a complete set of generalized derivatives of f at a given point [36], – a capability which is rarely available in practice. The obvious approach, i.e., using a subgradient instead of a gradient for the LMO model, has been shown to fail in general [30, Example 1]. There are subgradient-based approaches [4, 15, 33], but they are restricted to the convex case or a somewhat special function class. It appears that the analysis and theoretical tools from the abs-smooth literature including algorithmic differentiation and piecewise linear approximation have not been considered in conjunction with conditional gradient algorithms. To our knowledge, this is the first work bridging this gap in the literature.

In this article, we propose a generalization of the Frank-Wolfe algorithm for abs-smooth functions. Our analysis broadens the traditional nonconvex smooth setting of the Frank-Wolfe algorithm, which shows that an optimality criterion known as the *Frank-Wolfe gap*, whose definition relies on a gradient, asymptotically vanishes at a rate of $\mathcal{O}(1/\sqrt{t})$ [34]. Due to the nonsmoothness inherent to our problem class, we propose a generalization of the Frank-Wolfe gap for abs-smooth functions that captures the original Frank-Wolfe gap as a special case. We extend the current theory by proving that first-order optimality is achieved when our generalized Frank-Wolfe gap vanishes. Furthermore, we establish that our algorithm converges with a rate which is *identical* to that of the Frank-Wolfe algorithm when applied to nonconvex smooth objectives. This is consistent with previous results, since the smooth Frank-Wolfe algorithm arises as a particular case of our algorithm.

While the smooth Frank-Wolfe setting requires the solution of a linear minimization subproblem, it turns out that in general, our nonsmooth analogue of the Frank-Wolf algorithm necessitates the solution of a piecewise linear subproblem. In order to solve this task, we adapt the Active Signature Method of [17] to produce a locally minimal solution of our subproblem. We also establish that for several applications, including all constrained LASSO problems, our subroutine Algorithm 3 yields a global solution to the piecewise linear subproblem. We show that our new subroutine is computationally faster than other state-of-the-art methods on constrained piecewise linear problems, and we also benchmark our full algorithm on several standard nonsmooth test problems.

It is important to note that the ingredients of our method can be easily provided once the function to be optimized is given as computer program. Hence, it is readily applicable to such functions even if the requirements of the convergence theory provided in this paper can not be verified apriori. This is in contrast to many algorithms for solving nonsmooth optimization problems.

This article is structured as follows. In the remaining part of this section, we introduce abs-smooth functions and their properties. In Section 2, we present our Frank-Wolfe algorithm for abs-smooth functions (Section 2.1), propose the generalized Frank-Wolfe gap, derive guarantees of first-order optimality (Section 2.2), and provide convergence guarantees for our algorithm (Section 2.3). We also discuss the wide applicability of our approach to yield abs-smooth versions of many variants of the vanilla Frank-Wolfe algorithm in Remark 2.6. Section 3 is dedicated to the analysis and solution of our algorithm's piecewise linear subproblem. Our strategy for solving the piecewise linear subproblem is discussed in Section 3.1,

and potential relaxations are discussed in Section 3.2. Finally, numerical results are shown in Section 4. A summary and outlook are contained in Section 5.

1.1 Abs-smooth functions

Throughout we will consider the following class of target functions.

Definition 1.1 ($\mathcal{C}^d_{\text{abs}}(\mathbb{R}^n)$, abs-smooth functions)**.** *For any $d \in \mathbb{N}$, the set of locally Lipschitz continuous functions $f : \mathbb{R}^n \to \mathbb{R},\ y = f(x)$, defined by an* abs-smooth form

$$\begin{aligned} z &= F(x, z, |z|) \ , \\ y &= \varphi(x, z) \ , \end{aligned} \tag{2}$$

with $F \in \mathcal{C}^d(\mathbb{R}^{n+s+s}, \mathbb{R}^s)$ and $\varphi \in \mathcal{C}^d(\mathbb{R}^{n+s}, \mathbb{R})$, such that z_i is determined only by the values of z_j, $1 \le j < i$, is denoted by $\mathcal{C}^d_{\text{abs}}(\mathbb{R}^n)$. A function $f \in \mathcal{C}^d_{\text{abs}}(\mathbb{R}^n)$ is called abs-smooth*. The components z_i, $1 \le i \le s$, of z are called* switching variables*.*

Despite the fact that this definition seems to be rather technical, the class of abs-smooth functions is quite broad. For example, it encompasses a large subset of piecewise smooth functions in the sense of Scholtes [37] since the evaluation of $\max(.,.)$ and $\min(.,.)$ can be expressed using the absolute value function. Furthermore, abs-smooth functions capture a wide variety of nonsmooth functions used in machine learning applications, e.g., the ℓ_1-norm, i.e.,

$$f : \mathbb{R}^n \mapsto \mathbb{R}, \quad f(x) = \sum_{i=1}^{n} |x_i| = z_{n+1}, \quad z_i = x_i, \ \ 1 \le i \le n \ , \ \ z_{n+1} = \sum_{i=1}^{n} |z_i| \ ,$$

and the ReLU activation function given by

$$f(x) = \max(x, 0) = 0.5(x + |x|) = 0.5(x + z_2) = 0.5(z_1 + z_2), \ z_1 = x, \ z_2 = |z_1| \ .$$

Using the recursive evaluation procedure from [16], it follows that the composition of abs-smooth functions remains abs-smooth. Therefore, using the absolute value function and smooth univariate functions as building blocks, one can combine them via recursion and linear combination to construct a wide variety of abs-smooth functions, e.g., min, max, and the hinge Loss function. As a result, provided a neural network uses abs-smooth activation functions, its resulting squared ℓ_2-loss is also abs-smooth.

Also complementarity conditions or equilibrium constraints can be formulated in an abs-smooth form via

$$0 \le x \perp y \ge 0 \quad \Longleftrightarrow \quad 0 = f(x, y) = \min(x, y) = 0.5(x + y + |x - y|), \ z_1 = x - y, \ z_2 = |z_1| \ .$$

Furthermore, all functions that can be evaluated by a straight-line code using only smooth operations and the absolute value are abs-smooth. Finally, it is important to note that, for the application of the generalized Frank-Wolfe method proposed in this paper, the user does not have to state the function evaluation in the form (2), since correspondingly adapted AD tools can generate this representation in a completely automated fashion.

The main advantage of formulating all these applications as abs-smooth functions is the localization of the nonsmoothness as argument of an otherwise smooth function. In this way, the nonsmoothness can be explicitly exploited in combination with standard smooth optimization theory. For example, if an abs-smooth function is nonsmooth at a given point x then at least one of the switching variables as argument of the absolute value is evaluated at zero – motivating also the name for these variables.

Example 1.2 (Simple example)**.** *The function $f : \mathbb{R} \mapsto \mathbb{R}, f(x) = \max(0, x, 2x + 1)$ is abs-smooth since it can be stated in the following form*

$$f(x) = \max(0, x, 2x + 1) = 0.25(3x + 1 + |x + 1| + |3x + 1 + |x + 1||)$$

such that one obtains as one abs-smooth representation

$$\begin{aligned} z_1 &= x + 1 \,, \\ z_2 &= 3x + 1 + |z_1| \,, \\ z_3 &= |z_1| + |z_2| \,, \\ f(x) &= 0.25(3x + 1 + z_3) \,. \end{aligned}$$

As can be seen, at the only nonsmooth point $x = -0.5$, the switching variable z_2 is zero. Note, that it can happen that a switching variable is evaluated at zero but the function itself is smooth.

Example 1.3 (Mifflin II)**.** *The example Mifflin II given by the function*

$$f : \mathbb{R}^2 \mapsto \mathbb{R}, \qquad f(x) = -x_1 + 2\left(x_1^2 + x_2^2 - 1\right) + 1.75\left|x_1^2 + x_2^2 - 1\right| \,, \tag{3}$$

see, e.g., [1]*, is a well-established test case for nonsmooth optimization. It is abs-smooth since it has the representation*

$$\begin{aligned} z_1 &= x_1^2 + x_2^2 - 1 \\ z_2 &= |z_1| \\ y &= -x_1 + 2z_1 + 1.75z_2 \,. \end{aligned}$$

1.2 The abs-linearization

For an abs-smooth function f and a point $\bar{x}$, Griewank proposed in [16] the so-called *abs-linearization* $\Delta f(\bar{x}; \cdot)$ which can be used to construct a *piecewise linear model*

$$f_{PL,\bar{x}}(\cdot) \equiv f(\bar{x}) + \Delta f(\bar{x}; \cdot - \bar{x}) \approx f(\bar{x} + \cdot) \,.$$

This model can be viewed as a generalized Taylor expansion at $\bar{x}$ which simultaneously accounts for nonsmoothness and maintains second-order accuracy:

Theorem 1.4. *Suppose f is abs-smooth on $\mathcal{D} \subset \mathcal{K} \subset \mathbb{R}^n$, $\mathcal{D}$ open, $\mathcal{K}$ compact and convex. Then there exists $\gamma > 0$ such that for all $x, \bar{x} \in \mathcal{D}$*

$$\|f(x) - f_{PL,\bar{x}}(x)\| = \|f(x) - f(\bar{x}) - \Delta f(\bar{x}; x - \bar{x})\| \leq \gamma \|x - \bar{x}\|^2 \,. \tag{4}$$

Proof. See [16, Proposition 1]. □

For detailed explanation of the methodologies for generating Δf, we refer to [16, 20] as well as the Algorithmic Differentiation tools like ADOL-C [3], CppAD [5], and Tapenade [21] which have been extended to generate abs-linearizations and local piecewise linear models in an automated fashion. It is important to emphasize that, once a function f is available as a C or Fortran program, its piecewise linear approximant $f_{PL;\bar{x}}(\cdot)$ at a given point $\bar{x}$ is accessible in an easy way, with the same numerical effort as obtaining a gradient computed by algorithmic differentiation. In contrast to many machine-learning frameworks which use SGD even in nonsmooth settings, our advantage is that the piecewise linearization approximates f by explicitly taking its nonsmoothness into account. For intuition of how the abs-linearization behaves geometrically, we provide the abs-linearization of Example 1.3.

Example 1.5 (Mifflin II, continued)**.** *For the Mifflin II function, its abs-linearization is given by $\Delta f(\bar{x}; \cdot) : \mathbb{R}^2 \mapsto \mathbb{R}$ with*

$$\Delta f(\bar{x}; \Delta x) = -\Delta x_1 + 2\left(2\bar{x}_1\Delta x_1 + 2\bar{x}_2\Delta x_2\right) + 1.75\left(\left|\bar{x}_1^2 + \bar{x}_2^2 - 1 + 2\bar{x}_1\Delta x_1 + 2\bar{x}_2\Delta x_2\right| - \left|\bar{x}_1^2 + \bar{x}_2^2 - 1\right|\right) \,. \tag{5}$$

The function itself together with its piecewise linear model $f_{PL,\bar{x}}(.)$ at the point $\bar{x} = (-1.8, 1.8)$ are illustrated in Figure 1.

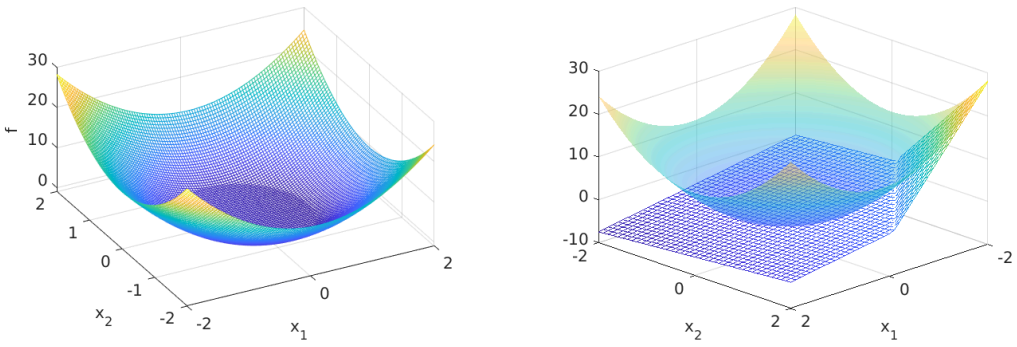


Figure 1: Abs-smooth function (3) from Example 1.5 and its piecewise linear model (5).

While abs-smooth functions may lack gradients, they are guaranteed to possess directional derivatives; furthermore, within a neighborhood around the development point $\bar{x}$, their directional derivatives coincide with the piecewise linear model based on abs-linearization.

Proposition 1.6. *Let $f \in \mathcal{C}^d_{\text{abs}}(\mathbb{R}^n)$ and $\bar{x} \in \mathbb{R}^n$. Then there exists a constant $\rho > 0$ such that, for every $d \in \mathbb{R}^n$, the directional (Bouligand) derivative $f'(\bar{x}; d)$ exists and if $\|d\| \leq \rho$ then*

$$\Delta f(\bar{x}; d) = f'(\bar{x}; d) \, . \tag{6}$$

As a result, for every $\alpha \in [0, 1]$, $\|d\| \leq \rho$ implies positive homogeneity $\Delta f(\bar{x}; \alpha d) = \alpha \Delta f(\bar{x}; d)$.

Proof. See [16, Section 3.2] and [20, pp. 390]. The final claim follows from positive homogeneity of $f'(\bar{x}; \cdot)$. □

Whether or not $\bar{x}$ resides on a kink, ρ basically describes the distance to the next-closest kink of $\Delta f(\bar{x}; \cdot)$ along the direction d. Hence, even though $\rho > 0$ is guaranteed, ρ can tend towards 0 as $\bar{x}$ approaches a point where f is nondifferentiable. Proposition 1.6 illuminates the connection between the abs-linearization and a *first-order minimal* point $\bar{x}$ [24, Theorem 3.8] which satisfies, for all $v \in C$,

$$f'(\bar{x}; v - \bar{x}) \geq 0 \, . \tag{7}$$

2 A Frank-Wolfe algorithm for abs-smooth functions

We begin in Section 2.1 by presenting our Abs-Smooth Frank-Wolfe algorithm (ASFW) and describing the guiding principles in its design. Sections 2.2 and 2.3 establish its theoretical footing by, respectively, providing optimality conditions and convergence guarantees. When the objective function f is smooth, the original Frank-Wolfe algorithm can be viewed as a special case of our algorithm; in this setting, the optimality and convergence guarantees for the nonconvex Frank-Wolfe algorithm would arise as corollaries to our results.

2.1 Motivating the algorithm

A vanilla Frank-Wolfe algorithm for the smooth setting is shown in Algorithm 1 (see also [7]). The hallmark of the Frank-Wolfe algorithm is in Line 3 – the so-called LMO step which, at iteration $t \in \mathbb{N}$, solves

$$\max_{v \in C} \; \langle -\nabla f(x_t) \, , \, v - x_t \rangle \, . \tag{8}$$

The optimal value of (8) is called the *Frank-Wolfe gap*. An advantage of Frank-Wolfe algorithms is that the subproblem (8) is oftentimes computationally cheaper than evaluating the projection onto C. However, since gradients are not available in the nonsmooth setting, this step of Frank-Wolfe algorithms must be changed. As far as we are aware, there is no generic replacement of (8) for the fully nonsmooth setting of Frank-Wolfe algorithms. For instance, Nesterov pointed out in [30, Example 1] that the simple approach of

replacing the gradient $\nabla f(x_t)$ in (8) with a subgradient – similar to subgradient descent methods – fails in general. However, it was recently shown that a subgradient-based approach does work for some nonsmooth functions [33].

A key component in deriving a comprehensive convergence theory for Algorithm 1 in the smooth nonconvex setting is the *smoothness inequality*, which provides a quadratic upper-bound on the difference between a function and its first-order Taylor approximation. Even though we do not have this inequality, the abs-linearization provides a piecewise linear model which also has second order accuracy (Theorem 1.4); in fact, this model *is* the Taylor series approximation in the smooth case. Hence, it is reasonable to consider a generalization of Algorithm 1, Step 3 which relies on the abs-linearization.

Algorithm 1 Frank-Wolfe algorithm

Require: Point $x_0 \in C$, smooth function f

1: **for** $t = 0$ **to** $\ldots$ **do**
2: Choose step size $\alpha_t \in (0, 1]$
3: Compute $v_t \in \arg\min_{v \in C} \langle \nabla f(x_t)\,,\, v \rangle$
4: $x_{t+1} = (1 - \alpha_t)x_t + \alpha_t v_t$
5: **end for**

Our approach in Algorithm 2 is to (a) generalize the Frank-Wolfe gap for abs-smooth functions, and (b) solve its corresponding piecewise linear optimization problem as a subroutine in our algorithm. As we further illustrate in Section 2.2, we propose the following *generalized Frank-Wolfe gap* for abs-smooth functions

$$\max_{v \in C} \frac{-\Delta f(x_t; \alpha_t(v - x_t))}{\alpha_t}\,, \tag{9}$$

whose corresponding optimization problem is formally equivalent to Line 3 in Algorithm 2.

As we will see in Section 2.2, just as in the smooth setting, first-order minimality is achieved when the generalized Frank-Wolfe gap vanishes. Furthermore, Section 2.3 demonstrates that the approximation properties in Theorem 1.4 can be leveraged to acquire identical convergence rates to the smooth setting.

Algorithm 2 Abs-Smooth Frank-Wolfe (ASFW) algorithm

Require: Point $x_0 \in C$, abs-smooth function f

1: **for** $t = 0$ **to** $\ldots$ **do**
2: Choose step size $\alpha_t \in (0, 1]$
3: Compute $v_t \in \arg\min_{v \in C} \Delta f(x_t, \alpha_t(v - x_t))$
4: $x_{t+1} = (1 - \alpha_t)x_t + \alpha_t v_t$
5: **end for**

2.2 Optimality criterion: Generalizing the Frank-Wolfe gap

In this section, we establish that, just as in the smooth setting, first-order optimality is acquired when the generalized Frank-Wolfe gap (9) vanishes.

We note that (9) is consistent with the original Frank-Wolfe gap (8). Indeed, if f is smooth, then for every $\bar{x} \in \mathbb{R}^n$, $\Delta f(\bar{x}; \cdot) = \langle \nabla f(\bar{x})\,,\, \cdot \rangle$ [16]. Hence, under the assumption of smoothness, at every iteration $t \in \mathbb{N}$ we have

$$\frac{-\Delta f(x_t; \alpha_t(v_t - x_t))}{\alpha_t} = \frac{-\langle \nabla f(x_t)\,,\, \alpha_t(v_t - x_t) \rangle}{\alpha_t} = \langle -\nabla f(x_t)\,,\, v_t - x_t \rangle\,. \tag{10}$$

In other words, our generalization captures the traditional Frank-Wolfe gap (8) as a special case. This generalization from a linear gap to a nonlinear gap also mirrors the form of the perspective function from nonlinear analysis [9].

In the smooth setting, if the Frank-Wolfe gap vanishes this implies first-order optimality. Hence, our goal is to show the same for our generalization. To-date, it was only possible to infer that f is Clarke-stationary

at $\bar{x}$, if $\Delta f(\bar{x};\cdot)$ was Clarke-stationary at 0 [11, Lemma 1]. However, since there are no guarantees that the solution to our problem will lie on a vertex of C, we do not have a guarantee that $v_t - x_t \to 0$. In fact, there are concrete examples where this never occurs – even for the smooth setting of Algorithm 1[7, Figure 2.4]. We therefore provide a new result below.

Theorem 2.1 (First-order minimality)**.** *Let $C \subset \mathbb{R}^n$ be nonempty and convex, let $\bar{x} \in C$, let $\alpha \in (0,1]$, and let $f \in \mathcal{C}^d_{\mathrm{abs}}(\mathbb{R}^n)$. Suppose that*

$$\max_{v \in C} \frac{-\Delta f(\bar{x}; \alpha(v - \bar{x}))}{\alpha} = 0 \,. \tag{11}$$

Then f is first-order minimal at $\bar{x}$, i.e., for every $v \in C$, $f'(\bar{x}; v - \bar{x}) \geq 0$. In particular, $\min_{v \in C} \Delta f(\bar{x}; v - \bar{x}) = 0$ implies first-order minimality at $\bar{x}$.

Proof. For the sake of contradiction, suppose that there exists a point $v \in C$ yielding a descent direction $f'(\bar{x}; v - \bar{x}) < 0$. Recall that Proposition 1.6 guarantees $\Delta f(\bar{x};\cdot) = f'(\bar{x};\cdot)$ for arguments with norm bounded by $\rho > 0$. So, for $\tau \in (0,1)$ satisfying $\alpha\tau\|v - \bar{x}\| \leq \rho$ we have $\alpha\tau v + (1 - \alpha\tau)\bar{x} \in C$ and hence

$$\frac{-\Delta f(\bar{x}; (\alpha\tau v + (1-\alpha\tau)\bar{x}) - \bar{x})}{\alpha} = \frac{-\Delta f(\bar{x}; \alpha\tau(v - \bar{x}))}{\alpha} = \frac{-f'(\bar{x}; \alpha\tau(v - \bar{x}))}{\alpha} = -\tau f'(\bar{x}; v - \bar{x}) > 0 \,,$$

which is absurd since it contradicts (11). □

Note that first-order minimality is sometimes also called d-stationarity, see, e.g., [33]. Hence, if (11) holds for $\bar{x}$, then $\bar{x}$ is d-stationary.

Corollary 2.2 (Convex optimality)**.** *Let $C \subset \mathbb{R}^n$ be a nonempty compact convex set, let $\bar{x} \in C$, and let $f \in \mathcal{C}^d_{\mathrm{abs}}(\mathbb{R}^n)$ be convex. Suppose that $\min_{v \in C} \Delta f(\bar{x}; v - \bar{x}) = 0$. Then $f(\bar{x}) = \min_{x \in C} f(x)$.*

Proof. From Theorem 2.1, we can conclude that

$$f'(\bar{x}; v - \bar{x}) \geq 0 \qquad \forall v \in C \,,$$

which is a sufficient optimality condition for convex functions, see, e.g., [24, Theorem 3.8]. □

2.3 Convergence results

In this section, we provide convergence proofs for Algorithm 2 under various settings. As we will see in this section, our $\mathcal{O}(1/\sqrt{t})$ convergence results achieve the same optimal rate as for the nonconvex *smooth* setting [34], even though our functions are nonsmooth. For this purpose, we show some important basic properties.

Corollary 2.3 (Sign of $\Delta f(x; \alpha(v_* - x))$)**.** *Let $x \in C$, let $\alpha > 0$, and let $f \in \mathcal{C}^d_{\mathrm{abs}}(\mathbb{R}^n)$. Suppose that $v_* \in C$ satisfies one of the following statements*

(i) $\Delta f(x, \alpha(v_* - x)) \leq \Delta f(x, 0)$

(ii) $v_* \in \arg\min_{v \in C} \Delta f(x; \alpha(v - x))$.

Then $\Delta f(x; \alpha(v_* - x)) \leq 0$.

Proof. Theorem 1.4 implies that $\Delta f(x; 0) = 0$ proving (i). To show (ii), since $x \in C$, we know

$$\Delta f(x; \alpha(v_* - x)) \leq \Delta f(x; \alpha(x - x)) = \Delta f(x; 0) \,,$$

so using (i) we obtain the required estimate. □

Lemma 2.4. *Let C be a nonempty compact convex set with diameter $D \in \mathbb{R}_{\geq 0}$. Assume that $f \in \mathcal{C}^d_{\mathrm{abs}}(\mathbb{R}^n)$. Then, for every $t \in \mathbb{N}$, the iterates generated by Algorithm 2 satisfy*

$$0 \leq \frac{-\Delta f(x_t; \alpha_t(v_t - x_t))}{\alpha_t} \leq \frac{f(x_t) - f(x_{t+1})}{\alpha_t} + \alpha_t \gamma D^2 \tag{12}$$

$$t \min_{0 \leq k \leq t-1} \frac{-\Delta f(x_k; \alpha_k(v_k - x_k))}{\alpha_k} \leq \sum_{k=0}^{t-1} \frac{-\Delta f(x_k; \alpha_k(v_k - x_k))}{\alpha_k} \leq \sum_{k=0}^{t-1} \frac{f(x_k) - f(x_{k+1})}{\alpha_k} + \gamma D^2 \sum_{k=0}^{t-1} \alpha_k \,. \tag{13}$$

Proof. It follows from Corollary 2.3 and Theorem 1.4 that

$$0 \leq -\Delta f(x_t; \alpha_t(v_t - x_t)) \leq f(x_t) - f(x_{t+1}) + \gamma \|x_{t+1} - x_t\|^2 = f(x_t) - f(x_{t+1}) + \alpha_t^2 \gamma \|v_t - x_t\|^2 \,.$$

Division by α_t yields (12), and summing over all iterations from 0 to $t-1$ yields (13). □

These properties already suffice to show the first nonconvex convergence result, where no additional assumptions on the piecewise linear model are required.

Theorem 2.5 (Open-loop convergence)**.** *Let C be a nonempty compact convex set with diameter $D \in \mathbb{R}_{\geq 0}$. Assume that $f \in \mathcal{C}^d_{\text{abs}}(\mathbb{R}^n)$. Then, for every $t \in \mathbb{N}$, the iterates generated by Algorithm 2 with $\alpha_t = 1/\sqrt{1+t}$ satisfy*

$$0 \leq \min_{0 \leq k \leq t-1} \frac{-\Delta f\left(x_k; \alpha_k(v_k - x_k)\right)}{\alpha_k} \leq \frac{1}{t} \sum_{k=0}^{t-1} \frac{-\Delta f(x_k; \alpha_k(v_k - x_k))}{\alpha_k} \leq \mathcal{O}\left(\frac{1}{\sqrt{t}}\right) \,. \tag{14}$$

Proof. Let x_* be a minimizer of f over C, and for every $t \in \mathbb{N}$, let $g_t = f(x_t) - f(x_*)$. Corollary 2.3 and Lemma 2.4 yield

$$\begin{aligned} 0 &\leq \min_{0 \leq k \leq t-1} \frac{-\Delta f(x_k; \alpha_k(v_k - x_k))}{\alpha_k} \leq \frac{1}{t} \sum_{k=0}^{t-1} \frac{-\Delta f(x_k; \alpha_k(v_k - x_k))}{\alpha_k} \\ &\leq \frac{1}{t} \left(\sum_{k=0}^{t-1} \frac{g_k - g_{k+1}}{\alpha_k} + \gamma D^2 \sum_{k=0}^{t-1} \alpha_k \right) \,. \end{aligned} \tag{15}$$

Since f is continuous and C is compact, f has a Lipschitz constant $\beta > 0$ over C. Hence, since $\sum_{k=0}^{t-1} \alpha_k \leq \sqrt{t}$,

$$\begin{aligned} \sum_{k=0}^{t-1} \frac{g_k - g_{k+1}}{\alpha_k} + \gamma D^2 \sum_{k=0}^{t-1} \alpha_k &= \frac{g_0}{\alpha_0} - \frac{g_t}{\alpha_{t-1}} + \sum_{k=1}^{t-1} \left(\frac{1}{\alpha_k} - \frac{1}{\alpha_{k-1}} \right) g_k + \gamma D^2 \sum_{k=0}^{t-1} \alpha_t \\ &\leq \beta D \left(\frac{1}{\alpha_0} + \sum_{k=1}^{t-1} \left(\frac{1}{\alpha_k} - \frac{1}{\alpha_{k-1}} \right) \right) + \gamma D^2 \sqrt{t} \\ &= (\beta D + \gamma D^2) \sqrt{t} \,, \end{aligned} \tag{16}$$

and substituting (16) into (15) yields the result. □

Remark 2.6. *Slight modifications in the proof of Theorem 2.5 also yield convergence results for the abs-smooth version of existing variants of the smooth Frank-Wolfe algorithm.*

(i) *The* fixed-horizon Frank-Wolfe algorithm *(see [7]), where a horizon $T \in \mathbb{N}$ is chosen and, for all iterations, the fixed step size $\alpha_t \equiv 1/\sqrt{T}$ is used. This simplifies (16) and guarantees that the gap is bounded by $\mathcal{O}(1/\sqrt{T})$ after T iterations – consistent with the results which require differentiability [7].*

(ii) *The* monotone Frank-Wolfe algorithm [8]*, wherein we keep the open-loop step sizes $\alpha_t = 1/\sqrt{1+t}$ and modify the iterate to update $x_{t+1} = \alpha_t v_t + (1-\alpha_t)x_t$ if $f(\alpha_t v_t + (1-\alpha_t)x_t) < f(x_t)$ and otherwise set $x_{t+1} = x_t$. Simple case analysis reveals that, in both update scenarios, (15) holds, and the proof proceeds identically as before.*

(iii) *For the setting when $\Delta f(x_t; \cdot)$ is convex, one can derive an abs-smooth version of the* short-step Frank-Wolfe algorithm *(see [7]). In the smooth setting, the step size is adaptively selected in order to minimize the smoothness inequality over $[0, 1]$; for the abs-smooth setting, the step size $\alpha_t = \min\{1, \Delta f(x_t; v_t - x_t)/2\gamma\|v_t - x_t\|^2\}$ adaptively minimizes the upper bound arising from Theorem 1.4 over the same set. Just as in the smooth nonconvex case [34], the function values $(f(x_t))_{t \in \mathbb{N}}$ are guaranteed to monotonically decrease, and $\mathcal{O}(1/\sqrt{t})$ convergence is acquired. While only local convexity of $\Delta f(x_t; \cdot)$ is guaranteed in general (cf. Proposition 1.6), convexity of $\Delta f(x_t; \cdot)$ is guaranteed in a variety of applications as detailed in the beginning of the next section.*

If one uses the open-loop step size strategy $\alpha_t = 2/(t+2)$, we have also observed $\mathcal{O}(1/t)$ convergence in experiments on both convex and nonconvex objectives. This can be proven to hold under the convexity-type inequality $\Delta f(x_t; x_* - x_t) \leq f(x_*) - f(x_t)$, which is not guaranteed to hold in the general abs-smooth setting.

3 The piecewise linear subproblem

This section concerns the central generalization from the linear subproblem in Line 3 of Algorithm 1 to the piecewise linear subproblem in Line 3 of Algorithm 2. In particular, for $\bar{x} \in \mathbb{R}^n$, and $\alpha > 0$ we must solve

$$\min_{v \in C} \ \Delta f(\bar{x}; \alpha(v - \bar{x})) \ . \tag{17}$$

While our theoretical analysis demonstrates that one can achieve the same per-iteration rate of convergence as in the smooth setting of Frank-Wolfe, this nonetheless requires solving the more challenging subproblem (17). As can be seen, instead of a constrained linear problem, one now has to solve a constrained piecewise linear problem. This can be a challenging optimization problem on its own, and there is no off-the-shelf algorithm for its solution; see [2] for a recent overview of nonsmooth optimization approaches and [25] for the case of piecewise linear constraints.

In Section 3.1, we discuss our approach for numerically solving (17) in the case when C is polyhedral. Our methodology is guaranteed to find a local minimizer, hence for the case when $\Delta f(\bar{x}; \cdot)$ is convex, we solve (17) exactly. As will be shown in Section 4, all constrained LASSO problems (in the sense of [13]) have convex piecewise linearizations. This also holds, e.g., for the Mifflin II problem in Examples 1.3 and 1.5 and the counterexample by Nesterov [30, Example 1]. In practice, we observe convergence to a first-order minimal solution even without the assumption of convexity. So, we conjecture that a relaxation of (17) may be sufficient to yield convergence, and we discuss this gap between theory and observations further in Section 3.2.

3.1 Solving the piecewise linear subproblem

In this subsection, we present an approach to minimize piecewise linear functions on a polyhedral feasible set. To better explain our methodology, we introduce the abs-linearization in more detail. Due to the smoothness in Definition 1.1, the following matrices and vectors are well-defined

$$\begin{aligned}
Z &= \frac{\partial}{\partial x} F(x,z,w) \in \mathbb{R}^{s \times n} \ , \\
M &= \frac{\partial}{\partial z} F(x,z,w) \in \mathbb{R}^{s \times s} \qquad \text{strictly lower triangular} \ , \\
L &= \frac{\partial}{\partial w} F(x,z,w) \in \mathbb{R}^{s \times s} \qquad \text{strictly lower triangular} \ , \\
a &= \frac{\partial}{\partial x} \varphi(x,z) \in \mathbb{R}^n \ , \qquad\qquad b = \frac{\partial}{\partial z} \varphi(x,z) \in \mathbb{R}^s \ .
\end{aligned}$$

For $f \in \mathcal{C}^d_{\text{abs}}(\mathbb{R}^n)$, these matrices define the *abs-linear form* of its piecewise linear model f_{PL} localized at $\bar{x} \in \mathbb{R}^n$ that is given by

$$\begin{aligned}
z &= c + Z\Delta x + Mz + L|z| \ , \\
f_{PL}(\bar{x}; \Delta x) &= d + a^\top \Delta x + b^\top z \ ,
\end{aligned} \tag{18}$$

for every $\Delta x \in \mathbb{R}^n$, where the constants $c \in \mathbb{R}^s$ and $d \in \mathbb{R}$ are chosen appropriately [16]. Such an abs-linear form can be generated using appropriate variants of AD [3, 5, 21]. The switching variables z in (18), which technically depend on $\Delta x \in \mathbb{R}^n$, are used to define the *signature matrix* of $f_{PL}(\bar{x}; \cdot)$, given by

$$\Sigma(\Delta x) = \operatorname{diag}(\sigma(\Delta x)) \ , \quad \text{where} \quad \sigma(\Delta x) = \operatorname{sign} z(\Delta x) \ .$$

For a fixed signature $\sigma \in \{-1, 0, 1\}^s$ and $\Sigma = \operatorname{diag}(\sigma)$, the inverse image

$$P_\sigma \ = \ \{\Delta x \in \mathbb{R}^n : \operatorname{sgn}(z(\Delta x)) = \sigma\}$$

is called a *signature domain*. These regions $(P_\sigma)_{\sigma\in\{-1,0,1\}^s}$ are relatively open polyhedra that form a partition of $\mathbb{R}^n$. It follows from [37, Proposition 2.2.2] that each piecewise linear function can be written in an abs-linear form with appropriately-sized vectors and real lower-triangular matrices. Since every abs-linear form has a switching variable z, it also gives rise to signature domains.

Note that the minimization of the piecewise linear function $f_{PL}(\bar{x};\cdot)$ is equivalent to the minimization of $\Delta f(\bar{x};\cdot-\bar{x})$, and their abs-linear forms only differ in the constants c and d. Since both functions are affine when restricted to a particular signature domain P_σ, constrained minimization over P_σ is achieved via the solution of one linear program (provided a solution exists).

Our approach is to adapt the Active Signature Method (ASM) of [17], which computes an unconstrained local minimizer of a given piecewise linear function ψ (for our applications, we will set $\psi = \Delta f(\bar{x};\alpha(\cdot-\bar{x}))$). The key idea of the ASM as proposed in [17] is to perform successive linear minimization over the signature domains of ψ. Initializing on the signature domain σ which contains $\bar{x}$, ASM then computes a minimizer of a regularized strongly convex problem $\psi(\cdot)+\|\cdot\|_Q^2$ (where $Q\geq 0$) subject to the constraint $\overline{P_\sigma}$. Note that (a) the quadratic penalty term ensures the existence of a minimizer, and (b) the solution may have a different signature than $\bar{x}$ if it resides on the boundary of a signature domain. A major feature of the ASM is that, in polynomial time, it can determine if the constrained minimizer over P_σ is a local minimizer of ψ over $\mathbb{R}^n$, see [19]. If local optimality is detected, ASM terminates. Otherwise, ASM identifies an adjacent polyhedron P_{σ^+} which ensures descent of the target function ψ, see [26], in the sense that

$$\min_{v\in P_{\sigma^+}} \psi(v) < \min_{v\in P_\sigma} \psi(v)\ ,$$

and repeats an iteration. Since there is a finite number of signature domains, ASM is guaranteed to terminate.

Algorithm 3 Adapted Active Signature Method (AASM)

Require: Point $\bar{x}\in\mathbb{R}^n$, piecewise linear function ψ

1: Initialize $P \leftarrow P_{\sigma(\bar{x})}$
2: **for** $t=0$ **to** $\ldots$ **do**
3: Compute $v_*\in\arg\min_{v\in\overline{P}\cap C}\psi(v)$
4: **if** v_* is a local minimizer of ψ over C **then**
5: Return v_*.
6: **else**
7: $P\leftarrow P_{\sigma^+}$ which guarantees $P_{\sigma^+}\cap C\neq\varnothing$ and descent of ψ
8: **end if**
9: **end for**

For our setting, Algorithm 3 performs successive minimization of ψ over the signature domains intersected with C. Since our feasible domain C is described by linear equalities and inequalities, they can be added as additional constraints to the description of the signature domain P_σ, effectively encoding the constraint $\overline{P}_\sigma\cap C$ as a single polyhedron. Since C is compact, ψ is guaranteed to possess a minimizer on every subdomain $\overline{P}_\sigma\cap C$ (provided it is nonempty). Therefore, we can remove the quadratic regularization term from ASM which was needed to guarantee existence of a solution. Since ψ is affine on every subdomain, computing a minimizer in Line 3 of Algorithm 3 is performed by a single LP call. Next, Algorithm 3 proceeds as in the ASM by checking the optimality conditions of [17] with $Q=0$. Algorithm 3 terminates if optimality is detected, and otherwise it proceeds to a new polyhedron which guarantees descent of ψ, see [19].

Under the assumption that $\Delta f(x_t;\cdot)$ is a convex function, local minima coincide with global minima so we solve (17) in Algorithm 2 by applying Algorithm 3 with $\psi(\cdot)=\Delta f(x_t;\cdot-x_t)$ and $\bar{x}=x_t$. We have also observed good algorithmic performance and convergence in settings where $\Delta f(x_t;\cdot)$ is not guaranteed to be convex. The theoretical analysis of this behavior is the subject of future work.

3.2 Alternative subproblems

Most of the computational effort in Algorithm 2 comes from computing an abs-linearization (performed once per iteration), and solving the sequence of linear programs in the inner solver Algorithm 3. Our experiments indicate that Algorithm 3 often terminates after a low number of iterations, and hence it uses a low number

of LP calls in-practice. Nonetheless, for some specific problems, solving (17) may require Algorithm 3 to visit all signature domains in a Klee-Minty exhaustive search [17], costing one LP call per signature domain. Since, in worst case settings, the number of signature domains is exponential in the number of switching variables s, the theoretical upper-bound on the number of LP calls per iteration of Algorithm 2 is quite high. Since this does not reflect what we see is required in-practice, this begs the question, *does Algorithm 2 produce a first-order minimal solution of (1) when, in Line 3, v_t is only a partial solution to (17)?* In the smooth setting, a similar question was answered in the affirmative via "lazified" variants of the Frank-Wolfe Algorithm 1 [6].

We provide a partial negative answer to this question by establishing that one iteration of the inner solver Algorithm 3 is insufficient to find a solution. This is demonstrated by considering an abs-linear objective function f: in this setting, at every iteration of Algorithm 2, $\Delta f(\bar{x};\cdot)$ is equal to $f(\cdot)$ (modulo translation), and hence the signature domains of the abs-linear model remain unchanged. Note that, if we only use one inner iteration of Algorithm 3, every vertex $(v_t)_{t\in\mathbb{N}}$ of Algorithm 2 resides in the closure of the same signature domain, i.e., the initial signature domain. Therefore, the iterates $(x_t)_{t\in\mathbb{N}}$ will also remain in the same closed convex set. So, unless Algorithm 2 is initialized on a signature domain whose closure contains a first-order minimal solution of (1), it cannot yield a solution.

Interestingly, even though replacing Line 3 in Algorithm 2 with one iteration of Algorithm 3 will not yield a solution in general, this algorithm *is* sufficient to guarantee that $\Delta f(x_t;\alpha_t(v_t-x_t))/\alpha_t$ converges to zero! Indeed, the proof of Theorem 2.5 only relies on Theorem 1.4 and negativity of our generalized gap, which we point out below.

Lemma 3.1. *Let $\alpha > 0$, let $\bar{x} \in \mathbb{R}^n$, and let $f \in \mathcal{C}^d_{\text{abs}}(\mathbb{R}^n)$. After one iteration of Algorithm 3 with $\psi = \Delta f(\bar{x};\alpha(\cdot-\bar{x}))$ and $x_0 = \bar{x}$, we have*

$$\Delta f(\bar{x};\alpha(v_1-\bar{x})) \leq 0\ .$$

Proof. Since v_1 is the result of minimizing over $\overline{P_\sigma} \ni \bar{x}$, we know $\Delta f(\bar{x};\alpha(v_*-\bar{x})) \leq \Delta f(\bar{x};\alpha(\bar{x}-\bar{x})) = 0$. □

In view of Lemma 3.1, one could use the same proof technique in Theorem 2.5 to show that $\Delta f(x_t;\alpha_t(v_t-x_t))/\alpha_t$ converges to zero. However, since v_t is no longer a solution to (11), $\Delta f(x_t;\alpha_t(v_t-x_t))/\alpha_t$ is not our generalized Frank-Wolfe gap, so we can not infer first-order minimality via Theorem 2.1.

In all of our experiments, we have observed convergence to a solution even when $\Delta f(\bar{x};\cdot)$ was not guaranteed to be convex. Therefore, since Algorithm 3 terminates when it detects local optimality, we conjecture that Algorithm 2 will still yield a first-order minimal solution of (1) when provided a locally minimal solution to (17).

4 Numerical examples

To verify our theoretical findings, we implemented the Frank-Wolfe approach for abs-smooth functions as stated in Algorithm 2 in C++. In Section 4.1, we benchmark our subproblem solver Algorithm 3. In Section 4.2, we test our full algorithm on a suite of scalable problems from nonsmooth optimization, and particular attention is given to constrained LASSO problems in Section 4.3. For the generation of the local piecewise linear model with abs-linearization we used ADOL-C [3]. For Linear Programming, we employed the solver HiGHS [22]. All computations were performed on a ThinkPad X1 laptop running Ubuntu 20.04.4 with an Intel Core i5-8265U CPU 1.60GHz x 8 processor.

4.1 Algorithm 3 (AASM): Rosenbrock-Nesterov II

In this section, we analyze the performance of our inner solver AASM on the Rosenbrock-Nesterov II function. According to [28], Nesterov suggested the Rosenbrock-like test function defined as

$$\psi : \mathbb{R}^n \to \mathbb{R}, \quad \psi(x) = \tfrac{1}{4}\,|x_1 - 1| \; + \sum_{i=1}^{n-1} |x_{i+1} - 2|x_i| + 1|\ , \tag{19}$$

which is piecewise linear and nonconvex. The analysis presented in [28] shows that this test function has (a) a unique global minimizer $x^* = (1, 1, \ldots, 1) \in \mathbb{R}^n$ and (b) $2^{n-1} - 1$ other stationary points which are not local minima, at which nonsmooth optimization algorithms may get stuck. Numerical tests showed that with the initial point

$$x_{0,1} = -1, \qquad x_{0,i} = 1 \quad 2 \le i \le n \,,$$

it is very likely to encounter all of the stationary points; see again [28]. Comparisons of three different solvers, i.e., a bundle method, an adapted quasi-Newton method, and ASM were presented for $1 \le n \le 10$ in [17]. It was found that the bundle method and the adapted quasi-Newton method got trapped at one of the stationary points for $3 \le n \le 10$ after a high number of iterations. On the other hand, for all values of n considered, ASM reached the minimizer exactly after 2^n iterations. Larger values of n were not considered due to numerical difficulties for these higher dimensions.

To validate the proposed Frank-Wolfe algorithm for abs-smooth functions, we introduced artificial bounds on the variables such that we have a compact convex feasible set. That is, we consider

$$C = \{x \in \mathbb{R}^n \mid -20 \le x_i \le 20,\ 1 \le i \le n\} \,.$$

Note that the constraint C excludes neither the suboptimal stationary points nor the global minimizer.

We coded the function evaluation exactly as stated in (19). That is, the required abs-linear form (18) was generated by ADOL-C. When applying Algorithm 3 to minimize ψ as defined in (19), we obtain the behavior shown in Table 1, where we also state the number of existing polyhedra for $n \le 10$ but skip this number for $n > 10$ due to the very large value. As compared to ASM which took 2^n iterations [17], the iteration numbers reduced significantly to 2^{n-1} corresponding exactly to the number of stationary points. Hence, due to the LP solve, AASM actually visits a very small fraction of all existing polyhedra (cf. Table 1). Furthermore, the dimension n could be increased considerably more than in [17]. It is very interesting to note that the presolve of the linear solver HiGHS reduces the linear optimization problem on each polyhedron to an empty one such that a solution could be computed without performing a simplex step at all – as seen in Table 1. This is a tremendous advantage in comparison to ASM which can be seen as an adapted QP solver. We observe this behavior of the linear solver also for other test problems. Due to this fact, even the largest instance which required more than 500,000 iterations could be solved in less than three minutes.

n	1	2	3	4	5	6	7	8	9	10
# polyhedra	1	8	32	128	512	2048	8192	32768	131072	524288
# iter (AASM)	1	2	4	8	16	32	64	128	256	512
# iter (simplex)	0	0	0	0	0	0	0	0	0	0
n	11	12	13	14	15	16	17	18	19	20
# iter (AASM)	1024	2048	4096	8192	16384	32768	65536	131072	262144	524288
# iter (simplex)	0	0	0	0	0	0	0	0	0	0

Table 1: Number of signature domains and iteration counts for Rosenbrock-Nesterov II example and Algorithm 3, i.e., the adapted active signature method (AASM).

4.2 Algorithm 2: Standard nonsmooth problems

In this subsection, we test our abs-smooth Frank-Wolfe (ASFW) Algorithm 2 on several standardized test cases (for details, see [1]). For that purpose, we implemented the two convex examples MAXQ and Chained LQ as well as the four nonconvex examples: Number of active faces, Chained Mifflin 2, Chained Crescent 1, and Chained Crescent 2. Furthermore, we tested the first nonsmooth and nonconvex Rosenbrock-Nesterov function analyzed in [28]. For all benchmark objectives except for MAXQ, we add bounds

$$C = \{x \in \mathbb{R}^n \mid -5 \le x_i \le 5, 1 \le i \le n\} \,,$$

which do not interfere with the optimal solution, allowing us to properly test our implementation. Section 4.2.1 reports on the results for MAXQ and considers different feasible sets in which constraints are active at the solution. All example functions are scalable such that we could examine various dimensions n. For all test cases, we observe a very similar convergence behavior for the step size $\alpha_t = 1/\sqrt{1+t}$, namely the convergence rate $\mathcal{O}(1/\sqrt{t})$.

Since the results are very similar for all test cases, we illustrate the convergence behavior just for two convex examples (Sections 4.2.1 and 4.2.2) and two nonconvex examples (Sections 4.2.3 and 4.2.4).

4.2.1 MAXQ problem

The MAXQ problem is given by

$$f : \mathbb{R}^n \mapsto \mathbb{R}, f(x) = \max_{1\le i\le n} x_i^2 \tag{20}$$
$$(\forall i \in \{1,2,\ldots,n\}) \quad (x_0)_i = \begin{cases} i & \text{if } i \in \{1,\ldots,\lfloor n/2\rfloor\}\,, \\ -i & \text{if } i \in \{\lfloor n/2\rfloor+1,\ldots,n\}\,. \end{cases}$$

For this academic test case, we also consider the following feasible sets:

$$\begin{aligned} C_1 &= \{x \in \mathbb{R}^n \mid -5 \le x_i \le 2i-2 \text{ for } i \in \{1,\ldots,\lfloor n/2\rfloor\}; \quad -2i+2 \le x_i \le 5 \text{ for } i \in \{\lfloor n/2\rfloor+1,...,n\}\}\,, \\ C_2 &= \{x \in \mathbb{R}^n \mid 0 \le x_i \le 2i-2 \text{ for } i \in \{1,\ldots,\lfloor n/2\rfloor\}; \quad -2i+2 \le x_i \le 0 \text{ for } i \in \{\lfloor n/2\rfloor+1,\ldots,n\}\}\,, \\ C_3 &= \{x \in \mathbb{R}^n \mid 1 \le x_i \le 2i-1 \text{ for } i \in \{1,\ldots,\lfloor n/2\rfloor\}; \quad -2i+1 \le x_i \le -1 \text{ for } i \in \{\lfloor n/2\rfloor+1,\ldots,n\}\}\,. \end{aligned}$$

The constraints in C_1 are inactive at the global solution $x_* = 0 \in \mathbb{R}^n$, while constraints in C_2 are active precisely at x_*. For C_3 we obtain the new optimal solution whose ith component is given by

$$(x_*)_i = \begin{cases} 1 & \text{for } i \in \{1,\ldots,\lfloor n/2\rfloor\}\,, \\ -1 & \text{for } i \in \{\lfloor n/2\rfloor+1,\ldots,n\}\,. \end{cases}$$

Once more, we coded the function evaluation exactly as stated in (20). That is, the abs-smooth form was used only internally in ASFW to generate the numerical results. For the step size $\alpha_t = 1/\sqrt{1+t}$, the observed convergence history is shown in Figure 2. Algorithm 2 terminated regularly with a norm of the generalized Frank-Wolf gap being smaller than 10^{-10}. For all combinations considered here, the function values also converged to the optimal value.

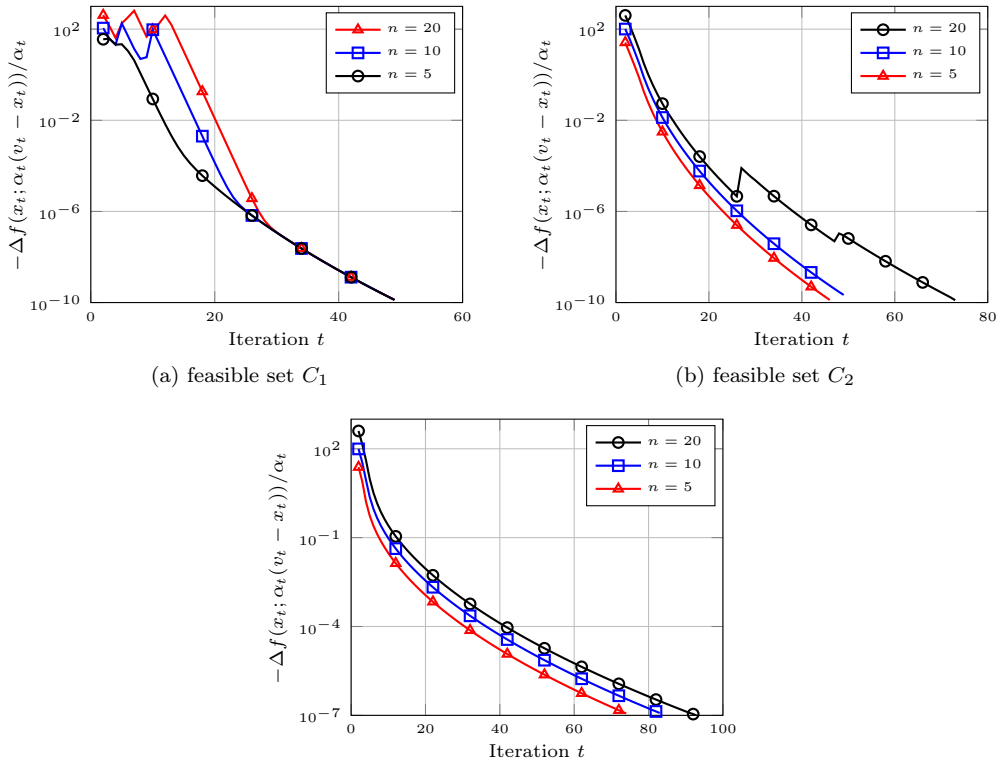
4.2.2 Chained LQ problem

The convex Chained LQ objective problem is given by

$$f(x) = \sum_{i=1}^{n-1} \max\left\{-x_i - x_{i+1},\ -x_i - x_{i+1} + x_i^2 + x_{i+1}^2 - 1\right\} \tag{21}$$
$$= \sum_{i=1}^{n-1} \tfrac{1}{2}\left(-2x_i - 2x_{i+1} + x_i^2 + x_{i+1}^2 - 1 + \left|x_i^2 + x_{i+1}^2 - 1\right|\right)\,, \tag{22}$$
$$x_{0,i} = -0.5 \quad \text{for all } \ i = 1,...,n\,,$$

where (21) is the version usually stated for this test problem and also used here for the implementation, whereas (22) can be used to derive a corresponding abs-smooth form. For a fixed $\bar{x}$, the abs-linearization generated in an automated fashion by ADOL-C is given by

$$\begin{aligned} \Delta f(\bar{x}, \Delta x) = \sum_{i=1}^{n-1} \Big(&- \Delta x_i - \Delta x_{i+1} + \bar{x}_i \Delta x_i + \bar{x}_{i+1} \Delta x_{i+1} \\ &+ \tfrac{1}{2}\left|\bar{x}_i^2 + \bar{x}_{i+1}^2 - 1 + 2\bar{x}_i \Delta x_i + 2\bar{x}_{i+1}\Delta x_{i+1}\right| - \tfrac{1}{2}\left|\bar{x}_i^2 + \bar{x}_{i+1}^2 - 1\right| \Big)\,. \end{aligned}$$



(a) feasible set C_1

(b) feasible set C_2

(c) feasible set C_3

Figure 2: Generalized Frank-Wolfe gap $-\Delta f(x_t;\alpha_t(v_t - x_t))/\alpha_t$ versus iteration count t displaying the convergence behavior of Algorithm 2 with $\alpha_t = 1/\sqrt{1+t}$ on the MAXQ problem (Section 4.2.1) for various values of n.

As can be seen, this function is convex in Δx. Hence, Algorithm 3 indeed solves (17) globally such that our convergence theory holds. For different values of n, the convergence history for the first 500 iterates is illustrated in Figure 3. One can clearly observe the convergence rate $\mathcal{O}(1/\sqrt{t})$, where the rate coefficients vary with n, as can be seen from the proof of Theorem 2.5. Furthermore, we also observe convergence of the function value to the optimal quantity $-(n-1)\sqrt{2}$ for all considered dimensions n.

4.2.3 Nonconvex Rosenbrock-Nesterov problem

Next, we present the results for the nonconvex Rosenbrock-Nesterov function

$$f(x) = \tfrac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} \left| x_{i+1} - 2x_i^2 + 1 \right| ,$$

$$(x_0)_i = \begin{cases} -0.5 & \text{if} \quad \text{mod } (i,2) = 1 \\ 0.5 & \text{if} \quad \text{mod } (i,2) = 0 \end{cases} \qquad \text{for} \quad i \in \{1,\dots,n\} .$$

For different values of n, the convergence history for the first 500 iterates is illustrated in Figure 4. Once more, the convergence rate $\mathcal{O}(1/\sqrt{t})$ is clearly visible and the heights of the lines vary with n, which is consistent with the prefactor's dependence on the set diameter D in the proof of Theorem 2.5. We again observe convergence of the function value to the optimal value 0 for all considered dimensions n.

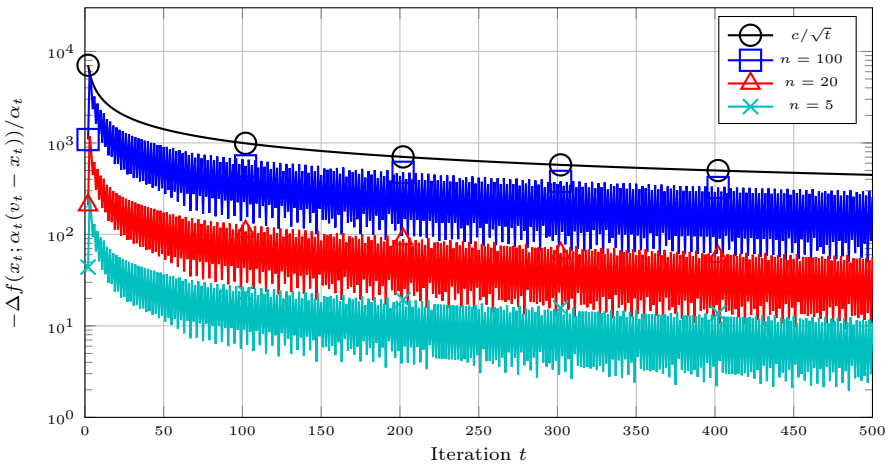


Figure 3: Convergence behavior of Algorithm 2 with $\alpha_t = 1/\sqrt{1+t}$ on the Chained LQ problem (Section 4.2.2) in various dimensions n.

4.2.4 Nonconvex Chained Crescent 1 problem

The nonconvex Chained Crescent 1 problem is given by

$$f(x) = \max\{f_1(x), f_2(x)\},$$
$$\text{with} \quad f_1(x) = \sum_{i=1}^{n-1} \left(x_i^2 + (x_{i+1}-1)^2 + x_{i+1} - 1\right), \quad f_2(x) = \sum_{i=1}^{n-1} \left(-x_i^2 - (x_{i+1}-1)^2 + x_{i+1} + 1\right),$$
$$(x_0)_i = \begin{cases} -1.5 & \text{if} \quad \text{mod}\ (i,2) = 1 \\ 2.0 & \text{if} \quad \text{mod}\ (i,2) = 0 \end{cases} \qquad \text{for} \quad i \in \{1,\dots,n\}.$$

For this problem, we tested the step size $\alpha_t = 2/(t+2)$ since, according to the theory developed for Frank-Wolfe methods, one may expect to observe a convergence rate of $\mathcal{O}(1/t)$. We can verify this convergence behavior numerically; see Figure 5 for the convergence history of the first 500 iterates. However, it is currently unknown if one can always expect this convergence rate.

4.3 Constrained LASSO problems

Finally, we consider the constrained LASSO problem

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \ \tfrac{1}{2}\|Ax - y\|_2^2 + \rho\|x\|_1 \\ &\text{s.t.} \quad Bx = b \quad \text{and} \quad Mx \le d, \end{aligned} \tag{23}$$

where $y \in \mathbb{R}^p$ is the response vector, $A \in \mathbb{R}^{p\times n}$ is the design matrix, $x \in \mathbb{R}^n$ is the vector of unknown regression coefficients, and $\rho \ge 0$ is a regularization parameter. The matrices $B \in \mathbb{R}^{r\times n}$, $M \in \mathbb{R}^{q\times n}$ and the vectors $b \in \mathbb{R}^r$, $d \in \mathbb{R}^q$ describe additional equality and inequality constraints. As its name suggests, the constrained LASSO augments the standard LASSO [35] with additional constraints that allow to take prior knowledge into account. This could be, for example, an ordering of the regression coefficients leading to an ordered LASSO problem or the requirement of positive regression coefficients yielding the positive LASSO.

The abs-linearization of the objective function (23) given by

$$\Delta f(\bar{x}, \Delta x) = (\bar{x}^\top A^\top - y)A\Delta x + \rho\left(\|\bar{x} + \Delta x\|_1 - \|\bar{x}\|_1\right)$$

is convex in Δx. While this fact is not exploited in the convergence analysis of Theorem 2.5, this does mean that our subproblem solver Algorithm 3 will yield a global minimizer.

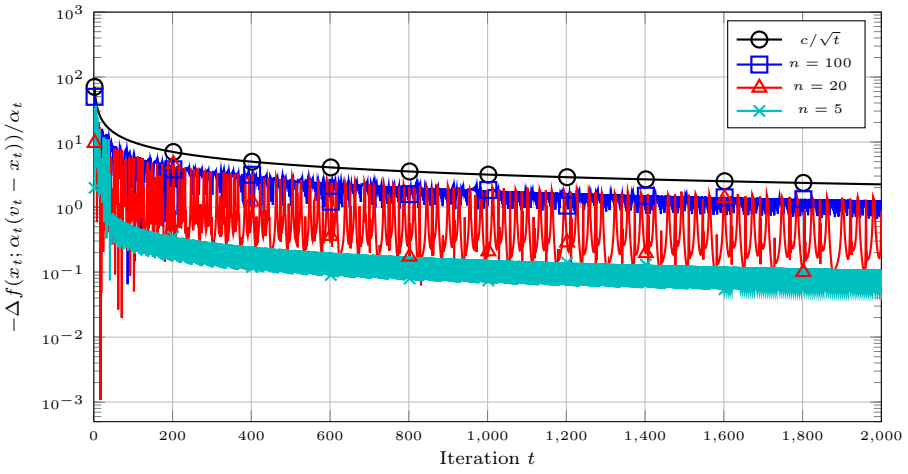


Figure 4: Convergence behavior of Algorithm 2 with $\alpha_t = 1/\sqrt{1+t}$ on the first Rosenbrock-Nesterov problem (Section 4.2.3) in various dimensions n.

To test the performance of our algorithm, we use random data so that we can scale the dimension arbitrarily. The entries of the design matrix A and the response are generated as independent and identical standard normal variables. We examine two variants of (23). First, we add again the bound constraints

$$C = \{x \in \mathbb{R}^n \mid -5 \leq x_i \leq 5,\ 1 \leq i \leq n\} .$$

Hence, for n and p large enough one expects $0 \in \mathbb{R}^n$ as the optimal solution. As initial point, we use a randomly generated vector with entries which are identically standard normal distributed.

For the step size $\alpha_t = 1/\sqrt{1+t}$, the results using various values of n and p are shown in Figure 6. All optimizations terminated at the optimal solution with the criterion $\Delta f(x_t, \alpha_t(v_t - x_t)) = 0$, i.e., when a first-order minimal point was found (cf. Theorem 2.1). For several combinations of n and p, Algorithm 2 reached a first-order minimal point early and therefore the corresponding curve ends. Once more, the convergence rate of $\mathcal{O}(1/\sqrt{t})$ proven in Theorem 2.5 is clearly visible in all scenarios.

Furthermore, we also tested the step size $\alpha_t = 2/(t+2)$. The optimization history for various values of n and p are shown in Figure 7. Once more, all optimizations terminated at the optimal solution with the stopping criterion $\Delta f(x_t, \alpha_t(v_t - x_t)) = 0$. The observed convergence rate is $\mathcal{O}(1/t)$, motivating further research in this direction.

For the second setting, we studied a LASSO problem where the feasible set is given by

$$\tilde{C} = \{x \in \mathbb{R}^n \mid -5 \leq x_0 \leq x_1 \leq \cdots \leq x_n \leq 5\} .$$

This setting corresponds to the requirement that the parameters grow monotonically, which is required for example in some climate models [13]. As initial point, we use

$$(x_0)_i = -1 + \frac{2(i-1)}{n-1} \qquad 1 \leq i \leq n ,$$

i.e., a feasible vector with equality distributed values from -1 to 1.

For the combinations $(n,p) \in \{125\} \times \{250, 375, 500\}$, $(n,p) \in \{250\} \times \{500, 750, 1000\}$, $(n,p) \in \{500\} \times \{1500, 2000\}$ and the step size $\alpha_t = 1/\sqrt{1+t}$, Algorithm 2 determined the optimal solution $x_* = 0 \in \mathbb{R}^n$ within two iterations, hence we do not plot the convergence for this experiment. Note that all constraints except for the lower and upper bound are active at the optimal solution found. Using the step size $\alpha_t = 2/(t+2)$, we also observed the same the convergence rate of $\mathcal{O}(1/t)$ as in Figure 7. Near the end of our experiments in both Figures 6 and 7, the algorithm terminates because the generalized Frank-Wolfe gap is zero. This sudden stopping is reminiscent of the smooth Frank-Wolfe literature [7] – in certain simple settings (e.g., for a linear or quadratic objective when the algorithm reaches the optimal face of C), one iteration of the Frank-Wolfe algorithm the algorithm exactly solves the optimization problem.

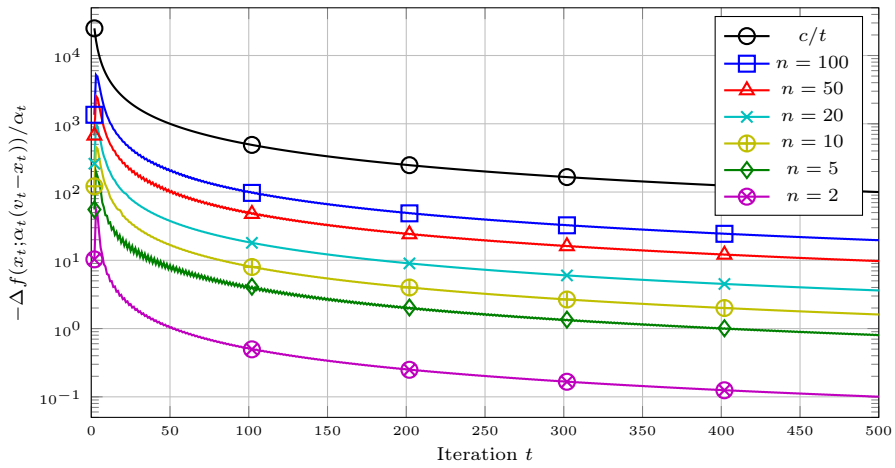


Figure 5: Convergence behavior of Algorithm 2 with step sizes $\alpha_t = 2/(t+2)$ on the Chained Crescent 1 problem (Section 4.2.4) in various dimensions n.

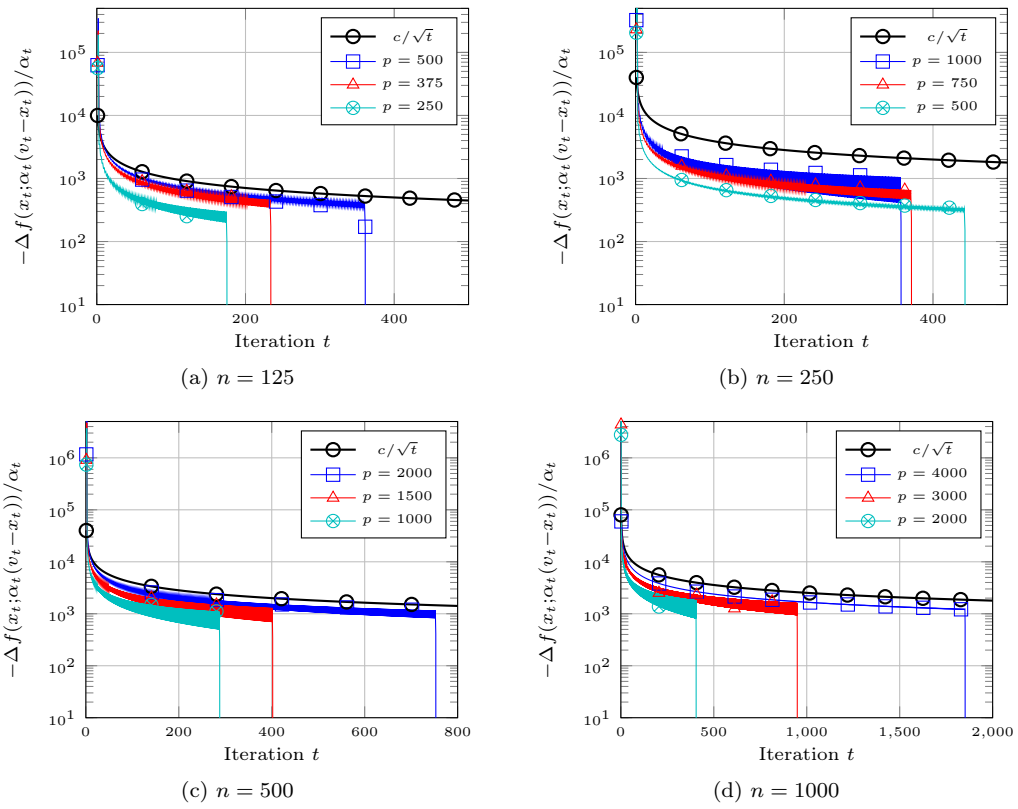
5 Summary and outlook

Even nowadays the solution of nonsmooth constrained optimization problems forms a challenge and correspondingly there is no off-the-self algorithm available. For a compact convex feasible set, we have shown that Algorithm 2, which appears to be the first connection between the fields of abs-smooth optimization and conditional gradient algorithms, can exhibit the same per-iteration rate of convergence as the Frank-Wolfe algorithm for smooth nonconvex objectives.

We have shown that, in generalizing from the smooth setting to the abs-smooth setting, the Frank-Wolfe gap becomes nonlinear and nonsmooth, and computing this gap necessitates the solution of a piecewise linear minimization problem (as opposed linear minimization). Our methodology stands in contrast to recent approaches which identify subclasses of nonsmooth functions for which, when one performs linear minimization against a subgradient, the Frank-Wolfe algorithm converges [4, 15, 33]. The approaches in [4, 15] only consider convex objectives, and [33] considers a class of functions which does not even contain the convex piecewise linear counterexample by Nesterov [30, Example 1]. Instead of finding a smaller class of functions for which subgradient-approaches work, we have generalized the Frank-Wolfe algorithm itself and shown that it will still converge on a broader class of functions.

The proposed algorithm can be implemented easily based on an AD tool that provides the required abs-linearization and an LP solver. The numerical illustrations in Sections 4.2 and 4.3 exhibit that the theory developed in this work is consistent with the rates observed in-practice. Furthermore, we observed experimentally that improved rates are possible with the step size strategy $\alpha_t = 2/(t+2)$. The experiments in Section 4.1 demonstrate that our subproblem solver Algorithm 3 scales far better than the previous algorithms for nonconvex nonsmooth optimization – capable of completing experiments at twice the dimension of the previous best contender.

As part of this line of work, we are left with several questions which we are eager to study in future work. Firstly, based on numerical experimentation and existing Frank-Wolfe literature, we believe that showing an $\mathcal{O}(1/t)$ convergence rate may be possible with a step size strategy of $\alpha_t = 2/(t+2)$. We also believe that the convergence analysis in Theorem 2.5 could be refined to directly incorporate the number of switching variables s. This would be particularly useful for improving convergence rates for functions with a low number of switching variables. Finally, our numerics indicate that Algorithm 2 works as long as one has a local minimizer to our ASFW-subproblem (17). If this is true in general, our approach of Algorithm 2 using Algorithm 3 as a subproblem solver would always converge to a solution, whether or not the objective function's piecewise linear model is convex.



(a) $n = 125$ (b) $n = 250$

(c) $n = 500$ (d) $n = 1000$

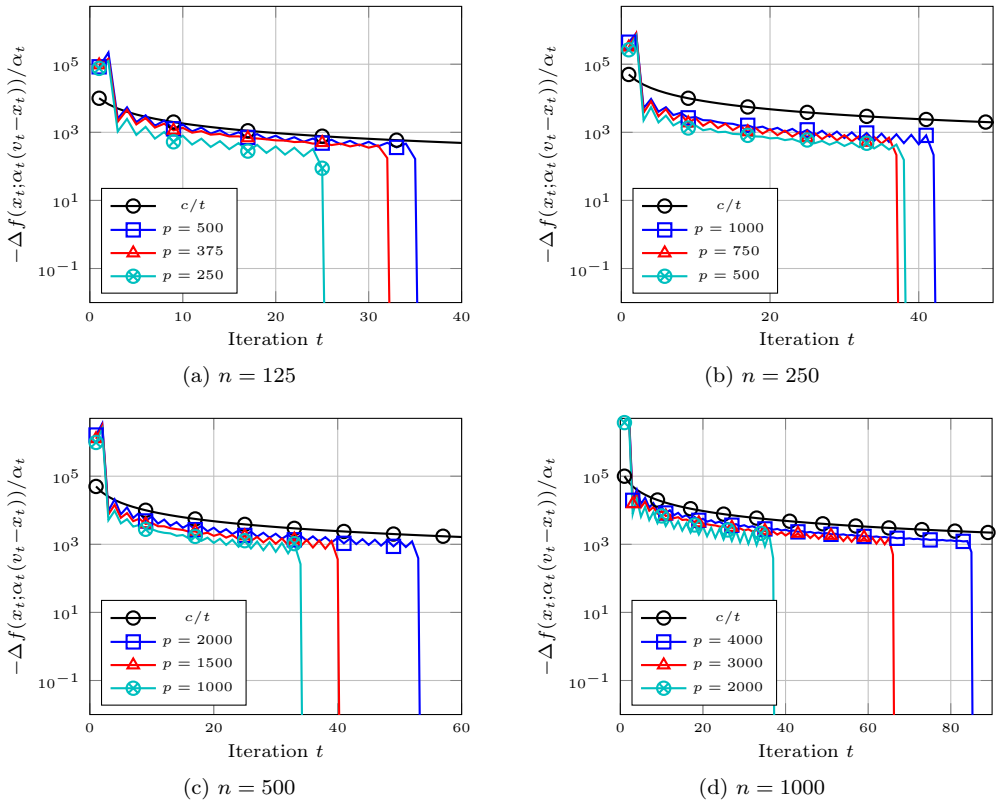
Figure 6: Convergence behavior of Algorithm 2 with $\alpha_t = 1/\sqrt{1+t}$ on the bound-constrained LASSO problem (Section 4.3) for various values of (n, p).

Acknowledgments

The authors thank the Deutsche Forschungsgemeinschaft for their support within Projects A05 and B10 in the Sonderforschungsbereich/Transregio 154 *Mathematical Modelling, Simulation and Optimization using the Example of Gas Networks* (project ID: 239904186).

The data that support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.



(a) $n = 125$ (b) $n = 250$

(c) $n = 500$ (d) $n = 1000$

Figure 7: Convergence behavior of Algorithm 2 with $\alpha_t = 2/(t+2)$ on the bound-constrained LASSO problem (Section 4.3) for various values of (n, p).

References

[1] A. Bagirov, N. Karmitsa, and M. Mäkelä. *Introduction to nonsmooth optimization. Theory, practice and software.* Springer, 2014.

[2] A. Bagirov et al., eds. *Numerical nonsmooth optimization. State of the art algorithms.* Cham: Springer, 2020.

[3] A. Walther and A. Griewank. "Combinatorial Scientific Computing". In: Chapman-Hall CRC Computational Science, 2012. Chap. Getting Started with ADOL-C, pp. 181–202.

[4] K. Asgari and M. J. Neely. *Projection-free non-smooth convex programming.* arXiv:2208.05127. 2022.

[5] B. Bell. *CppAD.* https://www.coin-or.org/CppAD/.

[6] G. Braun, S. Pokutta, and D. Zink. "Lazifying Conditional Gradient Algorithms". In: *Proceedings of the 34th International Conference on Machine Learning.* Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 566–575.

[7] G. Braun et al. *Conditional Gradient Methods.* arXiv:2211.14103. 2022.

[8] A. Carderera, M. Besançon, and S. Pokutta. "Simple steps are all you need: Frank-Wolfe and generalized self-concordant functions". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 5390–5401.

[9] P. L. Combettes. "Perspective functions: Properties, constructions, and examples". In: *Set-Valued and Variational Analysis* 26.2 (2018), pp. 247–264.

[10] J. Diakonikolas, A. Carderera, and S. Pokutta. "Locally accelerated conditional gradients". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 1737–1747.

[11] S. Fiege, A. Walther, and A. Griewank. "An Algorithm for Nonsmooth Optimization by Successive Piecewise Linearization". In: *Mathematical Programming* 177 (1-2 2018), pp. 343–370.

[12] M. Frank and P. Wolfe. "An algorithm for quadratic programming". In: *Naval Research Logistics Quarterly* 3.1-2 (1956), pp. 95–110.

[13] B. Gaines, J. Kim, and H. Zhou. "Algorithms for fitting the constrained Lasso". In: *Journal of Computational and Graphical Statistics* 27.4 (2018), pp. 861–871.

[14] D. Garber. "Faster projection-free convex optimization over the spectrahedron". In: *Advances in Neural Information Processing Systems* 29 (2016).

[15] D. Garber and E. Hazan. "A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization". In: *SIAM Journal on Optimization* 26.3 (2016), pp. 1493–1528.

[16] A. Griewank. "On stable piecewise linearization and generalized algorithmic differentiation". In: *Optimization Methods and Software* 28.6 (Apr. 2013), pp. 1139–1178.

[17] A. Griewank and A. Walther. "Finite convergence of an active signature method to local minima of piecewise linear functions". In: *Optimization Methods and Software* (Dec. 2018), pp. 1–21.

[18] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Second. USA: Society for Industrial and Applied Mathematics, 2008.

[19] A. Griewank and A. Walther. "Polyhedral DC decomposition and DCA optimization of piecewise linear functions". In: *Algorithms* 13.7 (2020), p. 166.

[20] A. Griewank et al. "On Lipschitz optimization based on gray-box piecewise linearization". In: *Mathematical Programming Series A* 158.1 (2016), pp. 383–415.

[21] L. Hascoët and V. Pascual. "The Tapenade Automatic Differentiation tool: Principles, Model, and Specification". In: *ACM Transactions on Mathematical Software* 39.3 (2013), 20:1–20:43.

[22] Q. Huangfu and J. A. J. Hall. "Parallelizing the dual revised simplex method". In: *Mathematical Programming Computation* 10.1 (2018), pp. 119–142.

[23] M. Jaggi. "Revisiting Frank-Wolfe: Projection-free sparse convex optimization". In: *International Conference on Machine Learning*. PMLR. 2013, pp. 427–435.

[24] J. Jahn. *Introduction to the theory of nonlinear optimization*. Springer Nature, 2020.

[25] T. Kreimeier, A. Walther, and A. Griewank. *An active signature method for constrained abs-linear minimization*. Preprint. TRR 154, 2021.

[26] T. Kreimeier. "Solving Constrained Piecewise Linear Optimization Problems by Exploiting the Abs-Linear Approach". PhD thesis. Humboldt-Universität zu Berlin, 2023.

[27] E. Levitin and B. Polyak. "Constrained minimization methods". In: *USSR Computational Mathematics and Mathematical Physics* 6.5 (1966), pp. 1–50.

[28] M. Gürbüzbalaban and M.L. Overton. "On Nesterov's nonsmooth Chebyshev-Rosenbrock functions." In: *Nonlinear Anal: Theory, Methods & Appl.* 75.3 (2012), pp. 1282–1289.

[29] J. Macdonald, M. E. Besançon, and S. Pokutta. "Interpretable Neural Networks with Frank-Wolfe: Sparse Relevance Maps and Relevance Orderings". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by K. Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 14699–14716.

[30] Y. Nesterov. "Complexity bounds for primal-dual methods minimizing the model of objective function". In: 171.1-2 (A) (2018), pp. 311–330.

[31] Y. Nesterov. "Smooth minimization of non-smooth functions". In: *Mathematical Programming* 103 (2005), pp. 127–152.

[32] G. Odor et al. “Frank-Wolfe works for non-Lipschitz continuous gradient objectives: Scalable poisson phase retrieval”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 6230–6234.

[33] W. de Oliveira. “Short Paper-A note on the Frank–Wolfe algorithm for a class of nonconvex and nonsmooth optimization problems”. In: *Open Journal of Mathematical Optimization* 4 (2023), pp. 1–10.

[34] F. Pedregosa et al. “Linearly convergent Frank-Wolfe with backtracking line-search”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 1–10.

[35] R. Tibshirani. “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society. Series B* 58.1 (1996), pp. 267–288.

[36] S. N. Ravi, M. D. Collins, and V. Singh. “A Deterministic Nonsmooth Frank Wolfe Algorithm with Coreset Guarantees”. In: *INFORMS Journal on Optimization* 1.2 (2018), pp. 120–142.

[37] S. Scholtes. *Introduction to Piecewise Differentiable Functions*. Springer, 2012.

[38] K. K. Tsuji, K. Tanaka, and S. Pokutta. “Pairwise Conditional Gradients without Swap Steps and Sparser Kernel Herding”. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by K. Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 21864–21883.

[39] A. Walther and A. Griewank. “Characterizing and Testing Subdifferential Regularity in Piecewise Smooth Optimization”. In: *SIAM Journal on Optimization* 29.2 (2019), pp. 1473–1501. eprint: https://doi.org/10.1137/17M115520X.