Berlin Mathematics Research Center MATH+

On a Frank-Wolfe Approach for Abs-Smooth Optimization

Sri Harshitha Tadinada (HUB), Sebastian Pokutta (TUB/ZIB), Andrea Walther (HUB) and Zev Woodstock (ZIB)

Problem setting

Given an abs-smooth function $f : \mathbb{R}^n \to \mathbb{R}$ and a compact convex constraint $C \subset \mathbb{R}^n$, we seek to

$$\text{minimize} \;\; f(x) \;\; \text{subject to} \;\; x \in C. \tag{1}$$

Setting:

- f is non-smooth, possibly non-convex (e.g., neural network loss with ReLU activation, Hinge loss, ...)
- C improves, e.g., sparsity, robustness, or interpretability
- Target application: medium to large-scale problems, so our algorithm must scale well

Abs-smooth functions

A function $f : \mathbb{R}^n \to \mathbb{R}$, $y = f(x)$ defined by an *abs-smooth form*

$$z = F(x, z, |z|) \quad \text{and} \quad y = \varphi(x, z)$$

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 *abs-smooth*.
Alternatively, [4] defines an *abs-smooth* function f procedurally:

$$\begin{aligned} v_0 &= x \\ &\vdots \\ (\forall\, j, k < i)\;\; v_i &= \begin{cases} v_j \circ v_k & \text{where } \circ \in \{+, -, \cdot\} \\ \phi_i(v_j) & \text{where } \phi_i \text{ smooth} \\ |v_j| \end{cases} \\ &\vdots \\ f(x) &= v_l \,. \end{aligned}$$

Examples: compositions of smooth functions, max, min, $|\cdot|$, their linear combinations. E.g., the *Hinge loss* is given by

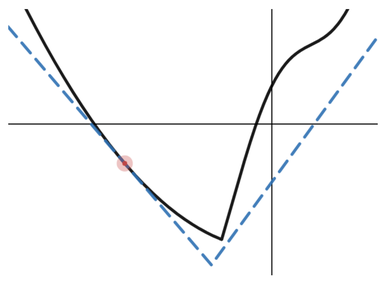
$$\max\{0, x\} = \frac{x + |x|}{2}, \; (v_1 = |v_0|, v_2 = v_0 + v_1, v_3 = v_2/2) \,.$$

The *abs-linearization* is given by an extended form of algorithmic differentiation [8, 11]

$$\begin{aligned} \Delta v_0 &= \Delta x \\ &\vdots \\ \Delta v_i &= \begin{cases} \Delta v_j \pm \Delta v_k & \text{if } v_i = v_j \pm v_k \\ v_j \Delta v_k + \Delta v_j v_k & \text{if } v_i = v_j \cdot v_k \\ \phi'(v_j)\Delta v_j & \text{if } v_i = \phi(v_j) \\ |\boldsymbol{v_j + \Delta v_j}| - \boldsymbol{v_i} & \text{if } \boldsymbol{v_i = |v_j|} \end{cases} \\ &\vdots \\ \Delta v_l &= \Delta f(x_0; \Delta x) \,. \end{aligned}$$

Yields a piecewise-linear approximation [4]

$$\|f(x_0) + \Delta f(x_0; \Delta x) - f(x_0 + \Delta x)\| \le \mathcal{O}(\|\Delta x\|^2)$$



According to [4, 5, 7, 9], we can minimize piecewise-linear functions with piecewise-linear constraints and general abs-smooth functions without constraints. First results with respect to the non-smooth Frank-Wolfe approach were addressed in [10], but there are a lot of improvements to be made.

Frank Wolfe algorithms

The Frank-Wolfe algorithm for *smooth* functions [3] is given by

"Vanilla" Frank-Wolfe algorithm

Require: Point $x_0 \in C$, smooth function f
1: **for** $t = 0$ **to** ... **do**
2: Compute $v_t \in \operatorname{argmin}_{v \in C} \langle \nabla f(x_t) \mid v \rangle$
3: Choose step size $\alpha_t \in (0, 1]$
4: $x_{t+1} = (1 - \alpha_t)x_t + \alpha_t v_t$
5: **end for**

where step 2 is denoted as *linear minimization oracle* (LMO) call.

FW can be faster than projection methods: e.g., for a nuclear norm ball or spectrahedron C, projection requires full eigen decomposition, while the LMO (step 2) a dominant eigenpair.

For smooth f, a vanilla Frank-Wolfe algorithm achieves [2]

	f Convex	f Non-convex
$\min_{k \in \{0,\dots,t-1\}} \langle -\nabla f(x_k) \mid v_k - x_k \rangle \to 0$	$\mathcal{O}\left(\frac{1}{t}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$
$f(x_t) - f(x_*) \to 0$	$\mathcal{O}\left(\frac{1}{t}\right)$	

where x^* is a solution to (1).
Note: first-order stationarity is closely related to vanishing of the *FW gap* $\langle -\nabla f(x_t) \mid v_t - x_t \rangle$.

Algorithm and convergence results

Motivation: in the non-smooth setting, an LMO solve is identical to minimizing the abs-linearization $\Delta f(x_t; \cdot - x_t)$. More details can be found in [10].

Frank-Wolfe algorithm for abs-smooth functions (ASFW)

Require: Point $x_0 \in C$, abs-smooth function f
1: **for** $t = 0$ **to** ... **do**
2: Choose step size $\alpha_t \in (0, 1]$
3: Compute $v_t \in \operatorname{argmin}_{v \in C} \Delta f(x_t; \alpha_t(v - x_t))$
4: $x_{t+1} = (1 - \alpha_t)x_t + \alpha_t v_t$
5: **end for**

Same order of convergence of FW gap as in smooth setting [10]:

Theorem (Open-loop convergence)

Let C be a compact convex set with diameter D. Assume that f is an abs-smooth function. Then, for every $t \in \mathbb{N}$, the iterates generated by ASFW algorithm with $\alpha_t \equiv 1/\sqrt{t+1}$ satisfy

$$0 \le - \min_{0 \le k \le T-1} \frac{\Delta f(x_k; \alpha_k(v_k - x_k))}{\alpha_k} \le \mathcal{O}(1/\sqrt{t}) \,.$$

Under the assumption that the given f is convex, we get improved results. We can not say anything about the convergence of the iterates x_t. Given U_t and L_t, the uniform upper and lower bounds of $f(x_{t+1})$ and $f(x^*)$ respectively, we have:

Theorem (Primal-Dual convergence)

Let $f \in \mathcal{C}^d_{\text{abs}}(\mathbb{R}^n)$ be a convex function over a compact convex set $C \subset \mathbb{R}^n$ with diameter D. Let step-sizes be given by $\alpha_t = 2/(t+2)$. For each $t \ge 1$, the iterates x_t of ASFW algorithm satisfy

$$G_t \le \frac{4(\gamma + c)D^2}{t + 2} \,,$$

where $x^* \in C$ is an optimal solution to problem 1 and G_t given by $f(x_{t+1}) - f(x^*) \le U_t - L_t = G_t$.

With given agnostic step-sizes α_t for $t \ge 1$, the order of convergence in the non-smooth setting are:

	f Convex	f Non-convex
$\min_{k \in \{0,\dots,t-1\}} \frac{-\Delta f(x_k; \alpha_k(v_k - x_k))}{\alpha_k} \to 0$	$\mathcal{O}\left(\frac{1}{t}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$
$f(x_t) - f(x^*) \to 0$	$\mathcal{O}\left(\frac{1}{t}\right)$	

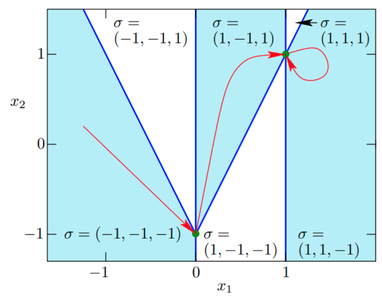
where x^* is a solution to (1).
It is possible that $\Delta f(x; \cdot)$ is convex even if the underlying f is non-convex [4]. Hence similar rates of convergence were also observed for some well known non-convex functions.

Solving the piecewise-linear subproblem

Step 3 in the ASFW algorithm requires

$$\operatorname{argmin}_{v \in C} \Delta f(x; \alpha(v - x)), \tag{2}$$

which can be easily solved in the setting when C is polyhedral via an augmented version of the *Active Signature Method (ASM)* [5].



Domain decomposition of Rosenbrock-Nestorov II with different signature vectors σ [6]

Piecewise-linear approximation induces a polyhedral partition of the domain. Yields an LP on each subdomain. We can solve (2) by "hopping" from subdomain to subdomain (*a la* [5, 9]), requiring a (usually short) sequence of LP solves.

Rosenbrock-Nesterov II

$\min f(x) = \frac{1}{4}|x_1 - 1| + \sum_{i=1}^{n-1} |x_{i+1} - 2|x_i| + 1|$
s.t. $x \in C = [-20, 20]^n$

- f is non-smooth and non-convex, $2^{n-1} - 1$ stationary points
- [5] compared bundle methods, ASM, and quasi-Newton, on the unconstrained problem; solved for $n \le 10$.
- ASFW (with adapted ASM) can solve this for $n \le 20$.

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. Adapted ASM computes the solution without a simplex step.

Julia version of code and numerics

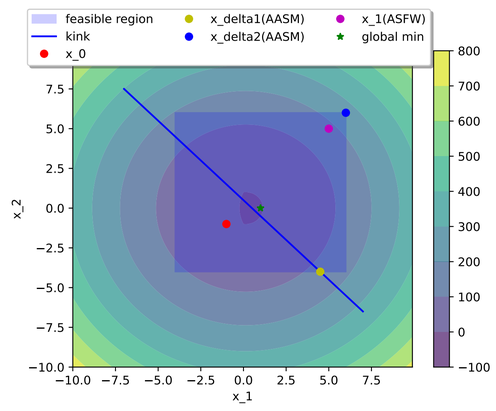
As part of the Frank-Wolfe algorithms - *FrankWolfe.jl*, the ASFW algorithm is implemented in Julia and is available as the Julia package - *AbsSmoothFrankWolfe.jl*. The code was tested against various benchmark examples from [1].

Problem (n)	Type	True $f(x^*)$	Computed $f(x^*)$	# iter	Time (s)	# simplex steps
CB3 (2)	Convex	2	f_{ASFW} = 2.000000e+00	8	2.61	36
DEM (2)	Convex	-3	f_{ASFW} = -3.000000e+00	2	2.57	7
LQ (2)	Convex	$-\sqrt{2}$	f_{ASFW} = -1.414203e+00	1738	5.78	6933
Rosen-Suzuki (4)	Convex	-44	f_{ASFW} = -4.399943e+01	4443	11.94	41004
Shor (5)	Convex	22.6	f_{ASFW} = 2.260020e+01	6078	18.09	122273
Wong 1 (7)	Convex	680.63	f_{ASFW} = 6.806305e+02	5511	17.96	84904
Wong 2 (10)	Convex	24.3	f_{ASFW} = 2.432384e+01	340	3.7	8131
Wong 3 (20)	Convex	133.72	f_{ASFW} = 1.337293e+02	9383	109.63	622982
MAXQ (20)	Convex	0	f_{ASFW} = 3.385600e-06	15281	51.10	789004
Crescent (2)	Non-convex	0	f_{ASFW} = 3.422500e-06	5518	12.82	17692
Mifflin-2 (2)	Non-convex	-1	f_{ASFW} = -9.999869e-01	5458	12.40	17725
Rosenbrock (2)	Non-convex	0	f_{ASFW} = 2.220446e-14	2	2.70	9
SPIRAL (2)	Non-convex	0	f_{ASFW} = 6.855629e-07	14213	36.4	54440

Table 2. Results for some standard small test problems.

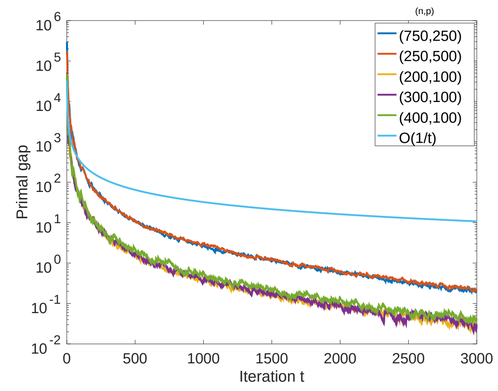
Visualization of one outer iteration of the ASFW Julia algorithm for 2d (non-convex) Chained Mifflin II function

$$f(x) = -x_1 + 2(x_1^2 + x_2^2 - 1) + 1.75\left|x_1^2 + x_2^2 - 1\right| \,.$$



Benchmark objective function [1] under polyhedral constraints: *(Convex) Constrained LASSO* ($b \in \mathbb{R}^p$)

$$f(x) = \|Ax - b\|^2 + \beta\|x\|_1 \,.$$



Outlook and future work

- ASFW algorithm with agnostic step size rule yields similar results as of the smooth Frank-Wolfe algorithms.
- *AbsSmoothFrankWolfe.jl* is a good alternative to the existing non-smooth optimization algorithms.
- Tech report on convergence results in preparation.
- We are looking into other variants of the abs-smooth Frank-Wolfe algorithms which help us in relaxation of few parameters and improve convergence rates.
- Current experiments are on standard "benchmark" problems from optimization [1]. Future work includes testing on machine learning applications and handling more than polyhedral constraints.

References

[1] A. Bagirov, N. Karmitsa, and M. Mäkelä. *Introduction to Nonsmooth Optimization. Theory, Practice and Software.* Springer, 2014.

[2] G. Braun, A. Carderera, C. Combettes, H. Hassani, A. Karbasi, A. Mokthari, and S. Pokutta. Conditional gradient methods. 2023. doi:10.48550/arXiv.2211.14103.

[3] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. doi:10.1002/nav.3800030109.

[4] A. Griewank. On stable piecewise linearization and generalized algorithmic differentiation. *Optimization Methods and Software*, 28(6):1139–1178. doi:10.1080/10556788.2013.796683.

[5] A. Griewank and A. Walther. Finite convergence of an active signature method to local minima of piecewise linear functions. *Optimization Methods and Software*, 34(5):1035–1055. doi:10.1080/10556788.2018.1546856.

[6] A. Griewank and A. Walther. First and second order optimality conditions for piecewise smooth objective functions. *Optimization Methods and Software*, 31(5):904–930. doi:10.1080/10556788.2016.1189549.

[7] A. Griewank, A. Walther, S. Fiege, and T. Bosse. On Lipschitz optimization based on gray-box piecewise linearization. *Mathematical Programming Series A*, 158(1):383–415, 2016. doi:10.1007/s10107-015-0934-x.

[8] L. Hascoët and V. Pascual. The Tapenade automatic differentiation tool: Principles, model, and specification. *ACM Transactions on Mathematical Software*, 39(3):20:1–20:43, 2013. doi:10.1145/2450153.2450158.

[9] T. Kreimeier, A. Walther, and A. Griewank. An active signature method for constrained abs-linear minimization. 2021. URL: http://www.optimization-online.org/DB_HTML/2021/12/8708.html.

[10] T. Kreimeier, A. Walther, S. Pokutta, and Z. Woodstock. On a Frank-Wolfe approach for abs-smooth functions, 2023. arXiv:2110.12650. doi:10.48550/ARXIV.2303.09881.

[11] A. Walther and A. Griewank. *Combinatorial Scientific Computing*, chapter Getting Started with ADOL-C, pages 181–202. Chapman-Hall CRC Computational Science, 2012.