

---

# Review of Frank Wolfe and its variants

---

William Saint-Arnaud  
Elyes Lamouchi  
Frederic Boileau

WILLIAM.ST-ARNAUD@UMONTREAL.CA  
ELYESLAMOUCHI@GMAIL.COM  
FREDERIC.BOILEAU@UMONTREAL.CA

## Abstract

Due to the combinatorial nature of multilabel outputs, predicting structured data typically comes with an exponentially large number of constraints, which makes the problem inefficient or intractable in practice. There has been a lot of research focused on providing a solution to that issue. In the structured SVM setting, conditional gradient a.k.a Frank-Wolfe type algorithms have become a method of choice.

This paper's aim is to synthesize the recent advances, starting from the classical F-W to the more sophisticated variants, while motivating this with the problems each variant addresses. Finally, we will discuss the pitfalls of some variants and their intrinsic trade-offs. We will then evaluate the performance of the methods proposed on synthetic data to see how reasonable the assumptions (providing theoretical guarantees) are, and to get an idea whether each variant's trade-off is worth it.

## 1. From classical Frank-Wolfe to more sophisticated variants

### 1.1. Classical Frank-Wolfe

Consider the problem of minimizing a convex objective function  $f$  over the convex hull of its domain  $\mathcal{M} = \text{conv}(\mathcal{A})$ .

By minimizing the first-order approximation of the objective function, the Frank-Wolfe algorithm takes a convex combination of the immediate iterate with the previous one. Consider a linear minimization oracle,

$$LMO_{\mathcal{A}}(\nabla f(x_t)) \in \text{argmin}_{s \in \mathcal{A}} \langle s, \nabla f(x_t) \rangle$$

Starting with an active set consisting of only an initial feasible point  $S^0 = \{x_0\}$ , the Frank-Wolfe algorithm adds an

"atom"  $s_t = LMO_{\mathcal{A}}(\nabla f(x_t))$ , to the active set in a convex combination with its elements while maintaining this combination sparse.

#### 1.1.1. CONVERGENCE RESULTS

We define the duality gap

$$g(\alpha^k) = \max_{s \in \mathcal{M}} \langle \alpha^k - s, \nabla f(\alpha^k) \rangle$$

By first order convexity of the objective, we have

$$\begin{aligned} f(s) &\geq f(\alpha^k) + \langle \alpha^k - s, \nabla f(\alpha^k) \rangle \\ \implies g(\alpha^k) &= -\min_{s \in \mathcal{M}} \langle \alpha^k - s, \nabla f(\alpha^k) \rangle \geq f(\alpha^k) - f^* \end{aligned}$$

We can thus see that the duality gap gives us a computable optimality guarantee.

**Definition.** The curvature constant  $C_f$  is given by the maximum relative deviation of the objective function  $f$  from its linear approximations, over the domain  $\mathcal{M}$ ,

$$C_f = \sup_{\substack{x, s \in \mathcal{M} \\ \gamma \in [0, 1], y = x + \gamma(s - x)}} \frac{2}{\gamma^2} \left( f(y) - f(x) - \langle y - x, \nabla f(x) \rangle \right)$$

Intuitively, the curvature constant can be seen as a measure of how flat the objective function is. For example, if the objective is linear, say  $f(x) = ax + b$  and  $x \in [e, f]$  then  $\nabla f(x) = a$  and the curvature constant is zero:

$$C_f = \frac{2}{\gamma^2} \left( ay + b - ax - b + (-ay + ax) \right) = 0$$

Moreover  $s = \text{argmin}_{s \in [e, f]} \langle s, a \rangle = \frac{e}{a}$ . Hence we reach the minimum in one F-W step.

Thus, we can observe that for flatter functions, that is with smaller curvature constants, Frank-Wolfe should converge faster.

**Theorem.** The duality gap obtained in the  $t^{\text{th}}$  iteration of the Frank-Wolfe algorithm satisfies

$$g(x_t) \leq 2\beta \frac{C_f}{t+2} (1 + \delta)$$

Where  $\beta = \frac{27}{8}$  and  $\delta$  is the approximation error tolerated in the *LMO*.

**Definition.** A function  $f$  has Lipschitz continuous gradient if:

$$\forall x, y \in \text{dom}(f), \exists L > 0 \text{ such that} \\ \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$$

**Theorem.** If a convex function  $f$  on  $C$  has Lipschitz gradient, i.e  $\|\nabla f(x) - \nabla f(y)\|_p \leq L_q\|x - y\|_p$ ,  $\forall x, y \in C$ , then

$$C_f \leq L_q \cdot \text{diam}_p^2(C)$$

**Proof.**  $f$  has Lipschitz gradient therefore by the fundamental descent lemma we have,

$$\begin{aligned} f(y) - f(x) - \langle y - x, \nabla f(x) \rangle &\leq \frac{L_q}{2} \|y - x\|_p^2 \\ C_f &\leq \max_{\substack{y=(1-\gamma)x+\gamma s \\ x, s \in C}} \frac{2}{\gamma^2} \frac{L_q}{2} \underbrace{\|y - x\|_p^2}_{=\gamma^2\|x-s\|_p^2} \\ C_f &\leq L_q \underbrace{\max_{x, s \in C} \|x - s\|_p^2}_{\triangleq \text{diam}_p^2(C)} \quad \square. \end{aligned}$$

Therefore, assuming  $\delta = 0$ , we get the following optimality certificate

$$f(\alpha^k) - f^* \leq g(\alpha^k) \leq 2\beta \frac{C_f}{t+2} \leq 2\beta \frac{L_q \cdot \text{diam}_p^2(C)}{t+2}$$

Thus, we see that the Frank-Wolfe algorithm has a sublinear convergence rate.

### 1.1.2. OPTIMALITY IN TERMS OF SPARSITY OF THE ITERATES

**Lemma.** For  $f(x) = \|x\|_2^2$  and  $1 \leq k \leq n$ , it holds that

$$\begin{aligned} \min_{\substack{x \in \mathcal{X} \\ \text{card}(x) \leq k}} f(x) &= \frac{1}{k}, \quad \text{and} \\ g(x) &\geq \frac{2}{k} \quad \forall x \in \Delta_n \text{ s.t. } \text{card}(x) \leq k. \end{aligned}$$

By the first equality we have, for any vector  $x$  s.t.  $\text{card}(x) = k$ , we get  $g(x) \leq \frac{1}{k} - \frac{1}{n}$ . Thus, combining the upper and lower bound, we have that the sparsity (number of used atoms) by the Frank-Wolfe algorithm is worst case optimal.

## 1.2. Away steps Frank-Wolfe

When the minimizer of the objective function lies at the boundary of the domain, after a number of iterations, the duality gap starts to stagnate.

### Algorithm 1 Frank-Wolfe

---

```

Let  $\alpha \in \mathcal{M}$ 
for  $k = 0$  to  $K$  do
    Compute  $s = \text{argmin}_{s \in \mathcal{M}} \langle s, \nabla f(\alpha^k) \rangle$ 
    Let  $\gamma = \frac{2}{k+2}$ , or optimize  $\gamma$  by line search
    Update  $\alpha^{k+1} = (1 - \gamma)\alpha^k + \gamma s$ 
end for
    
```

---

As a result of the strong dependency of the immediate iterate on previously accumulated atoms in the active set, as it approaches to boundary, the F-W algorithm starts to zig-zag around the descent direction.

To address this issue an improved variant of F-W named **Away-steps Frank-Wolfe** adds the possibility of moving away (by removing a fraction of) a maximizer of the  $LMO_{S_\square}$  in the active set. While this slows down each iteration it should be noted that the added step is easier than  $LMO_{\mathcal{A}}$  given that we maximize over a subset of  $\mathcal{A}$ . Furthermore, given that this variant converges linearly, the algorithm progresses in a fewer number of iterations in the descent direction, making it much faster than the original F-W.

## 1.3. Block Coordinate Frank Wolfe

### 1.3.1. STRUCTURED SVM CONTEXT

Given a training set  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$  where  $y \in \mathcal{Y}$  is a multi-label output, and a feature map  $\phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ , which encodes a similarity measure between  $\mathcal{X}$  and  $\mathcal{Y}$ , such that if  $y_i$  is the ground truth (target) for an input  $x_i$ , then

$$\forall y \in \mathcal{Y} \setminus \{y_i\} \text{ we have } \psi_i(y) = \phi(x_i, y_i) - \phi(x_i, y) > 0$$

The aim is to construct an accurate linear classifier,  $h_w(x) = \text{argmax}_{y \in \mathcal{Y}(x)} \langle w, \phi(x, y) \rangle$ .

To learn  $w$ , consider the task loss  $L : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ , where  $L(y, y') = 0 \iff y = y'$ .

The  $n$ -slack formulation of the problem would be,

$$\begin{aligned} \max_{w, \xi} \quad & \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^n \varepsilon_i \\ \text{s.t.} \quad & \langle w, \psi_i(y) \rangle \geq L(y_i, y) - \varepsilon_i, \quad \forall i, \forall y \in \mathcal{Y}(x) = \mathcal{Y}_i \end{aligned}$$

**Problems:** (1) The zero-one loss is not differentiable and (2) we have an exponential number of constraints.

**Solutions:** (1) Minimizing an upper bound to the task loss gives us a worst case guarantee.

Consider the **max oracle**,  $\tilde{H} = \max_{y \in \mathcal{Y}_i} \underbrace{L_i(y) - \langle w, \psi_i(y) \rangle}_{=H_i(y, w)} \text{ the hinge loss}$ .

(2) The exponential number of constraints are replaced by

$n$  piecewise linear ones.

**Proposition.** The max oracle is a convex upper bound to the task loss.

*Proof.* The maximum of two convex (linear) functions is convex, and

$$\begin{aligned} L(y_i, h_w(x_i)) &\leq L(y_i, h_w(x_i)) + \underbrace{\langle w, \psi_i(y) \rangle}_{\geq 0 \text{ by definition}} \\ &\leq \max_{y \in \mathcal{Y}_i} L_i(y) - \langle w, \psi_i(y) \rangle \end{aligned}$$

Thus learning  $w$  amounts to the unconstrained problem,

$$\max_w \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^n \tilde{H}_i(w)$$

### 1.3.2. BCFW VARIANT IN THE STRUCTURED SVM SETTING

Due to the exponential number of dual variables in the structured SVM setting, classical algorithms, like projected gradient are intractable.

Stochastic subgradient methods, on the other hand, achieve a sublinear convergence rate while only requiring a single call to the maximization oracle every step. They are nonetheless very sensitive to the sequence of stepsizes and it is unclear when to terminate the iterations.

Frank-Wolfe methods address these problems by giving an adaptive stepsize  $\gamma = \frac{2}{k+2}$  and a computable duality gap while still retaining a sublinear convergence rate. Moreover, despite the exponential number of constraints, the algorithm has sparse iterates alleviating the memory issues which come with the exponential number of dual variables.

**Note.** The main idea here, is that the linear subproblem in Frank-Wolfe and the loss augmented decoding of the structured SVM are equivalent.

*Proof of the equivalence.* The objective function being differentiable and convex, if we are at a point  $\alpha$  such that  $f(\alpha)$  is minimized along each coordinate axis, then  $\alpha$  is a global minimizer. Therefore,

$$\min_{s \in \mathcal{M}} \langle s, \nabla f(\alpha) \rangle = \sum_i \min_{s_i \in \Delta_{|\mathcal{Y}_i|}} \langle s_i, \nabla_i f(\alpha) \rangle$$

Moreover, with

$$\begin{aligned} w &= A\alpha, A = \left[ \frac{1}{n\lambda} \psi_1(y) \dots \frac{1}{n\lambda} \psi_{\sum_i |\mathcal{Y}_i|}(y) \right] \\ \text{and } b &= \left( \frac{1}{n} L_i(y) \right)_{i \in [n], y \in \mathcal{Y}_i} \end{aligned}$$

---

### Algorithm 2 Batch Primal-Dual Frank-Wolfe

---

```

Let  $\alpha \in \mathcal{M}$ 
Let  $w^0 = 0, l^0 = 0$ 
for  $k = 0, \dots, K$  do
  for  $i = 1, \dots, n$  do
    Solve  $y_i^* = \max_{y_i \in \mathcal{Y}_i} H_i(y, w^k) //$ 
    Let  $w_s = \sum_{i=1}^n \frac{1}{n\lambda} \psi_i(y_i^*)$ , and  $l_s = \frac{1}{n} \sum_{i=1}^n L_i(y_i^*)$ 
    Let  $\gamma = \frac{\lambda(w^k - w_s)^T w^k - l^k + l_s}{\lambda \|w^k - w_s\|^2}$ , and clip to  $[0, 1]$ 
    Update  $w^{k+1} = (1 - \gamma)w^k + \gamma w_s$ , and  $l^{k+1} = (1 - \gamma)l^k + \gamma l_s$ 
  end for
end for
    
```

---

The gradient of the dual would be,

$$\begin{aligned} \nabla f(\alpha) &= \nabla \left[ \frac{\lambda}{2} \|A\alpha\|^2 - b^T \alpha \right] = \lambda A^T A \alpha - b \\ &= \lambda A^T w - b = \frac{1}{n} H_i(y, w) \\ \max_{y_i \in \mathcal{Y}_i} \tilde{H}_i &= - \min_{y_i \in \mathcal{Y}_i} \tilde{H}_i = \min_{y_i \in \mathcal{Y}_i} L_i - \langle w, \psi_i \rangle \\ &= \min_{s_i \in \Delta_{|\mathcal{Y}_i|}} \langle s_i, \nabla_i f(\alpha) \rangle \end{aligned}$$

Thus we can see that, if  $n$  = size of the training data, one Frank-Wolfe step is equivalent to  $n$  calls to the maximization oracle. Unlike stochastic subgradient and stochastic methods in general, classical Frank-Wolfe requires one call for each training example at each iteration. For large datasets, this can get unpractical.

Hence the stochastic variant of Frank Wolfe, **Block Coordinate Frank Wolfe (BCFW)**.

**Theorem.** Given a convex, differentiable objective  $f : \mathcal{M}^1 \times \dots \times \mathcal{M}^n \rightarrow \mathbb{R}$ , where  $\forall i \in \{1..n\}$ , each factor  $\mathcal{M}^i \subseteq \mathbb{R}^n$  is convex and compact, if we are at a point  $x$  such that  $f(x)$  is minimized along each coordinate axis, then  $x$  is a global minimum.

As in coordinate descent, we minimize the objective function one coordinate (block) at a time. At each iteration, BCFW picks the  $i^{th}$  block (from  $n$ ) uniformly at random and updates the  $i^{th}$  coordinate of the corresponding weight, by calling the maximization oracle on the chosen block.

**Algorithm 3** Block-Coordinate Frank-Wolfe

---

Let  $w^0 = w_i^0 = \bar{w}^0 = 0, l^0 = l_i^0 = 0$   
**for**  $k = 0 \dots K$  **do**  
   Pick  $i$  at random in  $\{1, \dots, n\}$   
   Solve  $y_i^* = \max_{y_i \in \mathcal{Y}_i} H_i(y, w^k)$   
   Let  $w_s = \frac{1}{n\lambda} \psi_i(y_i^*)$ , and  $l_s = \frac{1}{n} L_i(y_i^*)$   
   Let  $\gamma = \frac{\lambda(w_i^k - w_s)^T w^k - l_i^k + l_s}{\lambda \|w_i^k - w_s\|^2}$ , and clip to  $[0, 1]$   
   Update  $w_i^{k+1} = (1 - \gamma)w_i^k + \gamma w_s$ , and  $l_i^{k+1} = (1 - \gamma)l_i^k + \gamma l_s$   
   Update  $w^{k+1} = w^k + w_i^{k+1} - w_i^k$ , and  $l_i^{k+1} = (1 - \gamma)l_i^k + \gamma l_s$   
**end for**

---

## 1.3.3. CONVERGENCE RESULTS

**Definition.** Over each coordinate block  $\mathcal{M}^i$ , let the curvature be given by,

$$C_f^{(i)} = \sup_{\substack{x \in \mathcal{M}, s_i \in \mathcal{M}^i \\ y = x + \gamma(s_i - x) \\ \gamma \in [0, 1]}} \frac{2}{\gamma^2} \left( f(y) - f(x) - \langle y_i - x_i, \nabla_i f(x) \rangle \right)$$

Where  $x_{[i]}$  refers to the zero-padding of  $i^{th}$  coordinate of  $x$ . And let the global product curvature constant be,

$$C_f^\otimes = \sum_{i=1}^n C_f^{(i)}$$

**Theorem.** For the dual structural SVM objective function over the domain  $\mathcal{M} = \Delta_{|\mathcal{Y}_1|} \times \dots \times \Delta_{|\mathcal{Y}_n|}$ , the total curvature constant  $C_f^\otimes$ , on the product domain  $\mathcal{M}$ , is upper bounded by,

$$C_f^\otimes \leq \frac{4R^2}{\lambda n} \quad \text{where} \quad R = \max_{i \in [n], y \in \mathcal{Y}_i} \|\psi_i(y)\|_2$$

**Proof.** By the second order convexity condition on  $f$  at  $y$ , we have

$$\begin{aligned}
 f(y) &\leq f(x) + \langle y_i - x_i, \nabla_i f(x) \rangle \\
 &\quad + (y - x)^T \nabla^2 f(x) (y - x) \\
 f(y) - f(x) - \langle y_i - x_i, \nabla_i f(x) \rangle \\
 &\leq (y - x)^T \nabla^2 f(x) (y - x)
 \end{aligned}$$

$$\begin{aligned}
 C_f^{(i)} &\leq \sup_{\substack{x \in \mathcal{M}, s_i \in \mathcal{M}^i \\ y = x + \gamma(s_i - x) \\ \gamma \in [0, 1]}} \left( f(y) - f(x) - \langle y_i - x_i, \nabla_i f(x) \rangle \right) \\
 &\leq \sup_{\substack{x, y \in \mathcal{M}, (y-x) \in \mathcal{M}^{[i]} \\ z \in [x, y] \subseteq \mathcal{M}}} (y - x)^T \nabla^2 f(z) (y - x)
 \end{aligned}$$

$$\begin{aligned}
 \text{Moreover} \quad &\sup_{\substack{x, y \in \mathcal{M}, (y-x) \in \mathcal{M}^{[i]} \\ z \in [x, y] \subseteq \mathcal{M}}} (y - x)^T \nabla^2 f(z) (y - x) \\
 &= \lambda \sup_{x, y \in \mathcal{M}, (y-x) \in \mathcal{M}^{[i]}} (A(y - x))^T \nabla^2 f(z) (A(y - x))
 \end{aligned}$$

$$C_f^{(i)} \leq \lambda \sup_{v, w \in A\mathcal{M}^{(i)}} \|v - w\|_2^2 \leq \lambda \sup_{v \in A\mathcal{M}^{(i)}} \|2v\|_2^2$$

Where  $\forall v \in A\mathcal{M}^{(i)}$ ,  $v$  is a convex combination of the feature vectors corresponding to the possible labelings for the  $i^{th}$  example of the training data, such that  $\|v\|_2 \leq$  the longest column of  $A = \frac{1}{n\lambda} R$ . Therefore,

$$C_f^\otimes = \sum_{i=1}^n C_f^{(i)} \leq 4\lambda \sum_{i=1}^n \left( \frac{1}{n\lambda} R \right)^2 = \frac{4}{n\lambda} R^2 \quad \square$$

First, we observe that the curvature constant for BCFW is  $n$  times smaller than that of batch Frank Wolfe which is  $\leq \frac{4}{\lambda} R^2$ . Hence the  $n$  times faster convergence rate of BCFW.

## 1.3.4. TIGHTENING THE BOUND

**Definition.** Let  $\|\cdot\|$  be a norm on  $\mathbb{R}^n$ . The associated dual norm, denoted  $\|\cdot\|_*$  is defined as,

$$\|z\|_* = \sup_z \{z^T x \mid \|x\| \leq 1\}$$

We denote the dual norm of  $l_p$  by  $l_q$ . For  $p = 2$  we have  $q = 2$  and for  $p = 1, q = \infty$ . *Problem.* For  $p = q = 2$  we get  $\text{diam}_2^2(C) = 2n$ , and the Lipschitz constant  $L_q$  is the largest eigenvalue of the hessian.

$$\lambda A^T A = \frac{1}{n^2 \lambda} \left( \langle \psi_i(y) - \psi_j(y') \rangle \right)_{(i,y),(j,y')}$$

And say  $\langle \psi_i(y) - \psi_j(y') \rangle \approx 1$  for a lot of outputs, we get:

$$1^T 1 \approx 1 \underbrace{\text{diam}_2(C)}_{=\sqrt{2n}}$$

Hence the largest eigenvalue the hessian, and therefore the Lipschitz constant, can scale with the dimension of  $A^T A$ , i.e exponentially with the size of the training data, rendering the bound above very loose, and thus of little practical use.

*Solution.* Taking  $p = 1$  and therefore  $q = \infty$ , we get  $L \text{diam}^2(C) \approx \frac{4}{\lambda} R^2$ .

Combined with the the convergence results above, we get a sublinear convergence rate for BCFW. And although subgradient methods converge at the same rate, BCFW presents an adaptive stepsize and an indication as to when to terminate, making it a more practical alternative.

**2. Randomized Away-step Frank-Wolfe**

A crucial assumption in constructing the BCFW is whether the domain is block-separable. While this is true in the context of the structured SVM, this leaves out important cases such as  $l_1$  constrained optimization (e.g. lasso type problems).

Moreover, while being an improvement on the classical variant by being  $n = \text{size of the data}$  times cheaper per iteration, BCFW still converges at a sublinear rate unlike the Away-step FW.

**The Randomized Away-steps Frank-Wolfe (RAWF)** finds a compromise between the two variants. By subsampling a  $\eta \in (0, 1]$  portion of the domain  $\mathcal{A}$  in the *LMO* and adding an away step at each iteration, we get a linear convergence rate with cheaper oracle calls than that of the original F-W.

### 2.0.1. CONVERGENCE RESULTS

**Definition.** Let the *away curvature*  $C_f^A$  and the *geometric strong convexity* constants be, respectively

$$C_f^A = \sup_{\substack{x, s, v \in \mathcal{M} \\ y = x + \gamma(s - x) \\ \gamma \in [0, 1]}} \frac{2}{\gamma^2} \left( f(y) - f(x) - \gamma \langle \nabla f(x), s - v \rangle \right)$$

$$\mu_f^A = \inf_{x \in M} \inf_{\substack{x^* \in \mathcal{M} \\ \langle \nabla f(x), x^* - x \rangle < 0}} \frac{2}{\gamma^A(x, x^*)^2} B_f(x, x^*)$$

$$\text{where } \gamma^A(x, x^*) = \frac{\langle -\nabla f(x), x^* - x \rangle}{\langle -\nabla f(x), s_f(x) - v_f(x) \rangle}$$

And  $s_f, v_f(x)$  are the FW atom and away atom respectively, starting from  $x$ .

**Theorem.** Consider the set  $\mathcal{M} = \text{conv}(\mathcal{A})$ , with  $\mathcal{A}$  a finite set of extreme atoms, after  $T$  iterations of RAWF, we have the following convergence rate

$$E[f(x_{T+1})] - f^* \leq (f(x_0) - f^*) \cdot (1 - \eta^2 \rho_f)^{\max\{0, \lfloor \frac{T-s}{2} \rfloor\}}$$

With  $\rho_f = \frac{\mu_f^A}{4C_f^A}$ ,  $\eta_{|\mathcal{A}|}^p$  and  $s = |S_0|$ .

**Proof sketch.** First we upper-bound  $h_t = f(x_t) - f^*$  by the pairwise dual gap  $\tilde{g}_t = \langle \tilde{s}_t - v_t \rangle$ , then we lower bound the progress  $h_t - h_{t+1}$  by using the away curvature constant in similar way to the proof in (Lacoste-Julien Jaggi, 2015, Theorem 8).□

With the above theorem, we get

$$\lim_{t \rightarrow \infty} \frac{Ef(x_{t+1}) - f^*}{Ef(x_t) - f^*} \in (0, 1)$$

Thus proving a linear convergence rate for the Randomized Away-steps Frank-Wolfe. aa

---

#### Algorithm 4 Randomized Away-steps Frank-Wolfe

---

Let  $x_0 = \sum_{v \in \mathcal{A}} \alpha_v^{(0)}$  with  $s = |S_0|$ , a subsampling parameter  $1 \leq p \leq |\mathcal{A}|$ .

**for**  $t = 0 \dots T$  **do**

    Get  $\mathcal{A}_\square$  by sampling  $\min\{p, |\mathcal{A} \setminus S_t|\}$  elements uniformly from  $|\mathcal{A} \setminus S_t|$

    Compute  $s_t = \text{LMO}(\nabla f(x), S_t \cup \mathcal{A}_\square)$

**Randomized Frank-Wolfe step:** Let  $d_t^{FW} = s_t - x_t$

    Compute  $v_t = \text{LMO}(-\nabla f(x), S_t)$

**Away step:** Let  $d_t^A = x_t - v_t$

**end for**

---