Frank-Wolfe and friends: a journey into projection-free first-order optimization methods

Immanuel. M. Bomze · Francesco Rinaldi · Damiano Zeffiro

Received: date / Accepted: date

Abstract Invented some 65 years ago in a seminal paper by Marguerite Straus-Frank and Philip Wolfe, the Frank-Wolfe method recently enjoys a remarkable revival, fuelled by the need of fast and reliable first-order optimization methods in Data Science and other relevant application areas. This review tries to explain the success of this approach by illustrating versatility and applicability in a wide range of contexts, combined with an account on recent progress in variants, both improving on the speed and efficiency of this surprisingly simple principle of first-order optimization.

Keywords First-order methods · Projection-free methods · Structured optimization · Conditional gradient · Sparse optimization

Immanuel M. Bomze
ISOR, VCOR & ds:UniVie, Universität Wien, Austria
E-mail: immanuel.bomze@univie.ac.at

Francesco Rinaldi
Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Italy
E-mail: rinaldi@math.unipd.it

Damiano Zeffiro
Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Italy
E-mail: damiano.zeffiro@math.unipd.it

1 Introduction

In their seminal work [36], Marguerite Straus-Frank and Philip Wolfe introduced a first-order algorithm for the minimization of convex quadratic objectives over polytopes, now known as Frank-Wolfe (FW) method. The main idea of the method is simple: to generate a sequence of feasible iterates by moving at every step towards a minimizer of a linearized objective, the so-called FW vertex. Subsequent works, partly motivated by applications in optimal control theory (see [33] for references), generalized the method to smooth (possibly non-convex) optimization over closed subsets of Banach spaces admitting a linear minimization oracle (see [30, 34]).

Furthermore, while the $\mathcal{O}(1/k)$ rate in the original article was proved to be optimal when the solution lies on the boundary of the feasible set [19], improved rates were given in a variety of different settings. In [67] and [30], a linear convergence rate was proved over strongly convex domains assuming a lower bound on the gradient norm, a result then extended in [33] under more general gradient inequalities. In [46], linear convergence of the method was proved for strongly convex objectives with the minimum obtained in the relative interior of the feasible set.

The slow convergence behaviour for objectives with solution on the boundary motivated the introduction of several variants, the most popular being Wolfe's away step [88]. Wolfe's idea was to move away from bad vertices, in case a step of the FW method moving towards good vertices did not lead to sufficient improvement on the objective. This idea was successfully applied in several network equilibrium problems, where linear minimization can be achieved by solving a min-cost flow problem (see [40] and references therein). In [46], some ideas already sketched by Wolfe were formalized to prove linear convergence of the Wolfe's away step method and identification of the face containing the solution in finite time, under some suitable strict complementarity assumptions.

In recent years, the FW method has regained popularity thanks to its ability to handle the structured constraints appearing in machine learning and data science applications efficiently. Examples include LASSO, SVM training, matrix completion, minimum enclosing ball, density mixture estimation, cluster detection, to name just a few (see Section 3 for further details).

One of the main features of the FW algorithm is its ability to naturally identify sparse and structured (approximate) solutions. For instance, if the optimization domain is the simplex, then after k steps the cardinality of the support of the last iterate generated by the method is at most $k+1$. Most importantly, in this setting every vertex added to the support at every iteration must be the best possible in some sense, a property that connects the method with many greedy optimization schemes [26]. This makes the FW method pretty efficient on the abovementioned problem class. Indeed, the combination of structured solutions with often noisy data makes the sparse approximations found by the method possibly more desirable than high precision solutions generated by a faster converging approach. In some cases, like in cluster detection (see, e.g., [10]), finding the support of the solution is actually enough to solve the problem independently from the precision achieved.

Another important feature is that the linear minimization used in the method is often cheaper than the projections required by projected-gradient methods. It is important to notice that, even when these two operations have the same complexity, constants defining the related bounds can differ significantly (see [28] for some examples and tests). When dealing with large scale problems, the FW method hence has a much smaller per-iteration cost with respect to projected-gradient methods. For this reason, FW methods fall into the category of *projection-free methods*. Furthermore, the method can be used to approximately solve quadratic subproblems in accelerated schemes, an approach usually referred to as conditional gradient sliding (see, e.g., [20, 65]).

1.1 Organisation of the paper

The present review is not intended to provide an exhaustive literature survey, but rather as an advanced tutorial demonstrating versatility and power of this approach. The article is structured as follows: in Section 2, we introduce the classic FW method, together with a general scheme for all the methods we consider. In Section 3, we present applications from classic optimization to more recent machine learning problems. In Section 4, we review some important stepsizes for first order methods. In Section 5, we discuss the main theoretical results about the FW method and the most popular variants, including the $\mathcal{O}(1/k)$ convergence rate for convex objectives, affine invariance, the sparse approximation property, and support identification. In Section 6 we illustrate some recent improvements on the $\mathcal{O}(1/k)$ convergence rate. Finally, in Section 7 we present recent FW variants fitting different optimization frameworks, in particular block coordinate, distributed, accelerated, and trace norm optimization.

1.2 Notation

For any integers a and b, denote by $[a:b] = \{x \text{ integer} : a \leq x \leq b\}$ the integer range between them. For a set V, the power set 2^V denotes the system all subsets of V, whereas for any positive integer $s \in \mathbb{N}$ we set $\binom{V}{s} := \{S \in 2^V : |S| = s\}$, with $|S|$ denoting the number of elements in S. Matrices are denoted by capital sans-serif letters (e.g., the zero matrix O, or the $n \times n$ identity matrix I_n with columns e_i the length of which should be clear from the context). The all-ones vector is $\mathsf{e} := \sum_i \mathsf{e}_i \in \mathbb{R}^n$. Generally, vectors are always denoted by boldface sans-serif letters x, and their transpose by $\mathsf{x}^\intercal$. The Euclidean norm of x is then $\|\mathsf{x}\| := \sqrt{\mathsf{x}^\intercal \mathsf{x}}$ whereas the general p-norm is denoted by $\|\mathsf{x}\|_p$ for any $p \geq 1$ (so $\|\mathsf{x}\|_2 = \|\mathsf{x}\|$). By contrast, the so-called zero-norm simply counts the number of nonzero entries:

$$\|\mathsf{x}\|_0 := |\{i \in [1:n] : x_i \neq 0\}| .$$

For a vector d we denote as $\widehat{\mathsf{d}} := \frac{1}{\|\mathsf{d}\|}\,\mathsf{d}$ its normalization, with the convention $\widehat{\mathsf{d}} = \mathsf{o}$ if $\mathsf{d} = \mathsf{o}$. Here o denotes the zero vector. In context of symmetric matrices, “psd” abbreviates “positive-semidefinite”.

2 Problem and general scheme

We consider the following problem:

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

where C is a convex and compact (i.e. bounded and closed) subset of $\mathbb{R}^n$ and, unless specified otherwise, f is a differentiable function having Lipschitz continuous gradient with constant $L > 0$:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \text{for all } \{x, y\} \subset C\,.$$

Throughout the article, we denote by x^* a (global) solution to (1) and use the symbol $f^* := f(x^*)$ as a shorthand for the corresponding optimal value.

The general scheme of the first-order methods we consider for problem (1), reported in Algorithm 1, is based upon a set $F(x, g)$ of directions feasible at x using first-order local information on f around x, in the smooth case $g = \nabla f(x)$. From this set, a particular $d \in F(x, g)$ is selected, with the maximal stepsize $\alpha^{\max}$ possibly dependent from auxiliary information available to the method (at iteration k, we thus write $\alpha_k^{\max}$), and not always equal to the maximal feasible stepsize.

Algorithm 1 `First-order method`

```
1 Choose a point $x_0 \in \Omega$
2 For $k = 0, \dots$
3     If $x_k$ satisfies some specific condition, then STOP
4     Choose $d_k \in F(x_k, \nabla f(x_k))$
5     Set $x_{k+1} = x_k + \alpha_k d_k$, with $\alpha_k \in (0, \alpha_k^{\max}]$ a suitably chosen stepsize
6 End for
```

2.1 The classical Frank-Wolfe method

The classical FW method for minimization of a smooth objective f generates a sequence of feasible points $\{x_k\}$ following the scheme of Algorithm 2. At the iteration k it moves toward a vertex i.e., an extreme point, of the feasible set minimizing the scalar product with the current gradient $\nabla f(x_k)$. It therefore makes use of a linear minimization oracle (LMO) for the feasible set C

$$\text{LMO}_C(g) \in \underset{z \in C}{\arg\min}\, g^\intercal z\,, \tag{2}$$

defining the descent direction as

$$d_k = d_k^{FW} := s_k - x_k, \;\; s_k \in \text{LMO}_C(\nabla f(x_k))\,. \tag{3}$$

In particular, the update at step 6 can be written as

$$x_{k+1} = x_k + \alpha_k(s_k - x_k) = \alpha_k s_k + (1 - \alpha_k)x_k \tag{4}$$

Since $\alpha_k \in [0, 1]$, by induction x_{k+1} can be written as a convex combination of elements in the set $S_{k+1} := \{x_0\} \cup \{s_i\}_{0 \leq i \leq k}$. When $C = \text{conv}(A)$ for a set A of points with some common property, usually called "elementary atoms", if $x_0 \in A$ then x_k can be written as a convex combination of $k+1$ elements in A. Note that due to Caratheodory's theorem, we can even limit the number of occurring atoms to $\min\{k, n\} + 1$. In the rest of the paper the primal gap at iteration k is defined as $h_k = f(\mathsf{x}_k) - f^*$.

Algorithm 2 `Frank-Wolfe method`

```
Choose a point $x_0 \in \Omega$
For $k = 0, \dots$
    If $\mathsf{x}_k$ satisfies some specific condition, then STOP
    Compute $\mathsf{s}_k \in \mathrm{LMO}_C(\nabla f(x_k))$
    Set $\mathsf{d}_k^{FW} = \mathsf{s}_k - \mathsf{x}_k$
    Set $\mathsf{x}_{k+1} = \mathsf{x}_k + \alpha_k \mathsf{d}_k^{FW}$, with $\alpha_k \in (0, 1]$ a suitably chosen stepsize
End for
```

3 Examples

FW methods and variants are a natural choice for constrained optimization on convex sets admitting a linear minimization oracle significantly faster than computing a projection. We present here in particular the traffic assignment problem, submodular optimization, LASSO problem, matrix completion, adversarial attacks, minimum enclosing ball, SVM training, maximal clique search in graphs.

3.1 Traffic assignment

Finding a traffic pattern satisfying the equilibrium conditions in a transportation network is a classic problem in optimization that dates back to Wardrop's paper [86]. Let $\mathcal{G}$ be a network with set of nodes $[1:n]$. Let $\{D(i,j)\}_{i \neq j}$ be demand coefficients, modeling the amount of goods with destination j and origin i. For any i, j with $i \neq j$ let furthermore $f_{ij} : \mathbb{R} \to \mathbb{R}$ be non-linear cost functions, and x_{ij}^s be the flow on link (i, j) with destination s. The traffic assignment problem can be modeled as the following non-linear *multicommodity network* problem [40]:

$$\min \left\{ \sum_{i,j} f_{ij} \left(\sum_s x_{ij}^s \right) : \sum_i x_{i\ell}^s - \sum_j x_{\ell j}^s = D(\ell, s)\,,\ \text{all } \ell \neq s,\ x_{ij}^s \geq 0 \right\}. \quad (5)$$

Then the linearized optimization subproblem necessary to compute the FW vertex can be split in n shortest paths subproblems, each of the form

$$\min \left\{ \sum_{i,j} c_{ij} x_{ij}^s : \sum_i x_{i\ell}^s - \sum_j x_{\ell j}^s = D(\ell, s),\, \ell \neq s \right\} \quad (6)$$

for a fixed $s \in [1:n]$. A number of FW variants were proposed in the literature for efficiently handling this kind of problems (see, e.g., [9, 40, 66, 87] and references therein for further details). In the more recent work [55] a FW variant also solving a shortest path subproblem at each iteration was applied to image and video co-localization.

3.2 Submodular optimization

Given a finite set V, a function $r : 2^V \to \mathbb{R}$ is said to be submodular if for every $A, B \subset V$

$$r(A) + r(B) \geq r(A \cup B) + r(A \cap B)\,. \tag{7}$$

As is common practice in the optimization literature (see e.g. [4, Section 2.1]), here we always assume $s(\emptyset) = 0$. A number of machine learning problems, including image segmentation and sensor placement, can be cast as minimization of a submodular function (see, e.g., [4,22] and references therein for further details):

$$\min_{A \subseteq V} r(A)\,. \tag{8}$$

Submodular optimization can also be seen as a more general way to relate combinatorial problems to convexity, for example for structured sparsity [4,53]. By a theorem from [39], problem (8) can be in turn reduced to an minimum norm point problem over the base polytope

$$B(G) = \{\mathsf{s} \in \mathbb{R}^V : \sum_{a \in A} s_a \leq r(A) \text{ for all } A \subseteq V\,,\ \sum_{a \in V} s_a = r(V)\}\,. \tag{9}$$

For this polytope, linear optimization can be achieved with a simple greedy algorithm. More precisely, consider the LP

$$\max_{\mathsf{s} \in B(F)} \mathsf{w}^\intercal \mathsf{s}\,.$$

Then if the objective vecor w has a negative component, the problem is clearly unbounded. Otherwise, a solution to the LP can be obtained by ordering w in decreasing manner as $w_{j_1} \geq w_{j_2} \geq ... \geq w_{j_n}$, and setting

$$s_{j_k} := r(\{j_1, ..., j_k\}) - r(\{j_1, ..., j_{k-1}\})\,, \tag{10}$$

for $k \in [1\!:\!n]$. We thus have a LMO with a $\mathcal{O}(n \log n)$ cost. This is the reason why FW variants are widely used in the context of submodular optimization; further details can be found in, e.g., [4,53].

3.3 LASSO problem

The LASSO, proposed by Tibshirani in 1996 [81], is a popular tool for sparse linear regression. Given the training set

$$T = \{(\mathsf{r}_i, b_i) \in \mathbb{R}^n \times \mathbb{R} : i \in [1\!:\!m]\}\,,$$

where $\mathsf{r}_i^\intercal$ are the rows of an $m \times n$ matrix A, the goal is finding a sparse linear model (i.e., a model with a small number of non-zero parameters) describing the data. This problem is strictly connected with the Basis Pursuit Denoising (BPD) problem in signal analysis (see, e.g., [25]). In this case, given a discrete-time input signal b, and a *dictionary*

$$\{\mathsf{a}_j \in \mathbb{R}^m \ :\ j \in [1\!:\!n]\}$$

of elementary discrete-time signals, usually called atoms (here a_j are the columns of a matrix A), the goal is finding a sparse linear combination of the atoms that *approximate* the real signal. From a purely formal point of view, LASSO and BPD problems are equivalent, and both can be formulated as follows:

$$\begin{aligned} &\min_{\mathsf{x}\in\mathbb{R}^n} \; f(\mathsf{x}) := \|\mathsf{A}\mathsf{x}-\mathsf{b}\|_2^2 \\ &s.t. \quad \|\mathsf{x}\|_1 \le \tau\,, \end{aligned} \tag{11}$$

where the parameter τ controls the amount of shrinkage that is applied to the model (related to sparsity, i.e., the number of nonzero components in x). The feasible set is

$$C = \{\mathsf{x}\in\mathbb{R}^n : \|\mathsf{x}\|_1 \le \tau\} = \mathrm{conv}\{\pm\tau\mathsf{e}_i : \; i\in[1\!:\!n]\}\,.$$

Thus we have the following LMO in this case:

$$\mathrm{LMO}_C(\nabla f(\mathsf{x}_k)) = \mathrm{sign}(-\nabla_{i_k} f(\mathsf{x}_k))\cdot\tau\mathsf{e}_{i_k}\,,$$

with $i_k \in \arg\max_i |\nabla_i f(\mathsf{x}_k)|$. It is easy to see that the FW per-iteration cost is then $\mathcal{O}(n)$. The peculiar structure of the problem makes FW variants well-suited for its solution. This is the reason why LASSO/BPD problems were considered in a number of FW-related papers (see, e.g., [52,53,62,68]).

3.4 Matrix completion

Matrix completion is a widely studied problem that comes up in many areas of science and engineering, including collaborative filtering, machine learning, control, remote sensing, and computer vision (just to name a few; see also [18] and references therein). The goal is to retrieve a low rank matrix $\mathsf{X}\in\mathbb{R}^{n_1\times n_2}$ from a sparse set of observed matrix entries $\{U_{ij}\}_{(i,j)\in J}$ with $J\subset[1\!:\!n_1]\times[1\!:\!n_2]$. Thus the problem can be formulated as follows [38]:

$$\begin{aligned} &\min_{\mathsf{X}\in\mathbb{R}^{n_1\times n_2}} \; f(\mathsf{X}) := \sum_{(i,j)\in J} (X_{ij}-U_{ij})^2 \\ &s.t. \qquad \mathrm{rank}(\mathsf{X}) \le \delta, \end{aligned} \tag{12}$$

where the function f is given by the squared loss over the observed entries of the matrix and $\delta > 0$ is a parameter representing the assumed belief about the rank of the reconstructed matrix we want to get in the end. In practice, the low rank constraint is relaxed with a nuclear norm ball constraint, where we recall that the nuclear norm $\|\mathsf{X}\|_*$ of a matrix X is equal the sum of its singular values. Thus we get the following convex optimization problem:

$$\begin{aligned} &\min_{\mathsf{X}\in\mathbb{R}^{n_1\times n_2}} \; \sum_{(i,j)\in J} (X_{ij}-U_{ij})^2 \\ &s.t. \qquad \|\mathsf{X}\|_* \le \delta\,. \end{aligned} \tag{13}$$

The feasible set is the convex hull of rank-one matrices:

$$\begin{aligned} C &= \{\mathsf{X}\in\mathbb{R}^{n_1\times n_2} : \|\mathsf{X}\|_* \le \delta\} \\ &= \mathrm{conv}\{\delta\mathsf{u}\mathsf{v}^\mathsf{T} : \mathsf{u}\in\mathbb{R}^{n_1}, \mathsf{v}\in\mathbb{R}^{n_2}, \; \|\mathsf{u}\| = \|\mathsf{v}\| = 1\}\,. \end{aligned}$$

If we indicate with A_J the matrix that coincides with A on the indices J and is zero otherwise, then we can write $\nabla f(\mathsf{X}) = 2\,(\mathsf{X}-\mathsf{U})_J$. Thus we have the following LMO in this case:

$$\mathrm{LMO}_C(\nabla f(\mathsf{X}_k)) \in \arg\min\{\mathrm{tr}(\nabla f(\mathsf{X}_k)^\intercal \mathsf{X}) : \|\mathsf{X}\|_* \leq \delta\}\,, \tag{14}$$

which boils down to computing the gradient, and the rank-one matrix $\delta \mathsf{u}_1 \mathsf{v}_1^\intercal$, with $\mathsf{u}_1, \mathsf{v}_1$ right and left singular vectors corresponding to the top singular value of $-\nabla f(\mathsf{X}_k)$. Consequently, the FW method at a given iteration approximately reconstructs the target matrix as a sparse combination of rank-1 matrices. Furthermore, as the gradient matrix is sparse (it only has $|J|$ non-zero entries) storage and approximate singular vector computations can be performed much more efficiently than for dense matrices[1]. A number of FW variants has hence been proposed in the literature for solving this problem (see, e.g., [38,52,53]).

3.5 Adversarial attacks in machine learning

Adversarial examples are maliciously perturbed inputs designed to mislead a properly trained learning machine at test time. An *adversarial attack* hence consists in taking a correctly classified data point x_0 and slightly modifying it to create a new data point that leads the considered model to misclassification (see, e.g., [21, 24,45] for further details). A possible formulation of the problem (see, e.g., [23, 45]) is given by the so called *maximum allowable ℓ_p-norm attack* that is,

$$\begin{aligned} &\min_{\mathsf{x}\in\mathbb{R}^n} \ f(\mathsf{x}_0 + \mathsf{x}) \\ &s.t. \quad \|\mathsf{x}\|_p \leq \varepsilon\,, \end{aligned} \tag{15}$$

where f is a suitably chosen attack loss function, x_0 is a correctly classified data point, x represents the additive noise/perturbation, $\varepsilon > 0$ denotes the magnitude of the attack, and $p \geq 1$. It is easy to see that the LMO has a cost $\mathcal{O}(n)$. If x_0 is a feature vector of a dog image correctly classified by our learning machine, our adversarial attack hence suitably perturbs the feature vector (using the noise vector x), thus getting a new feature vector $\mathsf{x}_0 + \mathsf{x}$ classified, e.g., as a cat. In case a target adversarial class is specified by the attacker, we have a *targeted attack*. In some scenarios, the goal may not be to push x_0 to a specific target class, but rather push it away from its original class. In this case we have a so called *untargeted attack*. The attack function f will hence be chosen depending on the kind of attack we aim to perform over the considered model. Due to its specific structure, problem (15) can be nicely handled by means of tailored FW variants. Some FW frameworks for adversarial attacks were recently described in, e.g., [23, 56,78].

3.6 Minimum enclosing ball

Given a set of points $P = \{\mathsf{p}_1, \ldots, \mathsf{p}_n\} \subset \mathbb{R}^d$, the minimum enclosing ball problem (MEB, see, e.g., [26,92]) consists in finding the smallest ball containing P. Such a

[1] Details related to the LMO cost can be found in, e.g., [53].

problem models numerous important applications in clustering, nearest neighbor search, data classification, machine learning, facility location, collision detection, and computer graphics, to name just a few. We refer the reader to [60] and the references therein for further details. Denoting by $\mathsf{c} \in \mathbb{R}^d$ the center and by $\sqrt{\gamma}$ (with $\gamma \geq 0$) the radius of the ball, a convex quadratic formulation for this problem is

$$\min_{(\mathsf{c},\gamma)\in\mathbb{R}^d\times\mathbb{R}} \gamma \tag{16}$$

$$s.t. \qquad \|\mathsf{p}_i - \mathsf{c}\|^2 \leq \gamma\,, \quad \text{all } i \in [1\!:\!n]\,. \tag{17}$$

This problem can be formulated via Lagrangian duality as a convex *Standard Quadratic Optimization Problem* (StQP, see, e.g. [12])

$$\min\{\mathsf{x}^\intercal \mathsf{A}^\intercal \mathsf{A}\mathsf{x} - \mathsf{b}^\intercal \mathsf{x} : \mathsf{x} \in \Delta_{n-1}\} \tag{18}$$

with $\mathsf{A} = [\mathsf{p}_1, ..., \mathsf{p}_n]$ and $\mathsf{b}^\intercal = [\mathsf{p}_1^\intercal \mathsf{p}_1, \dots, \mathsf{p}_n^\intercal \mathsf{p}_n]$. The feasible set is the standard simplex

$$\Delta_{n-1} := \{\mathsf{x} \in \mathbb{R}^n_+ : \mathsf{e}^\intercal \mathsf{x} = 1\} = \text{conv}\{\mathsf{e}_i : i \in [1\!:\!n]\}\,,$$

and the LMO is defined as follows:

$$\text{LMO}_{\Delta_{n-1}}(\nabla f(\mathsf{x}_k)) = \mathsf{e}_{i_k}\,,$$

with $i_k \in \arg\min_i \nabla_i f(\mathsf{x}_k)$. It is easy to see that cost per iteration is $\mathcal{O}(n)$. When applied to (18), the FW method can find an ε-cluster in $\mathcal{O}(\frac{1}{\varepsilon})$, where an ε-cluster is a subset P' of P such that the MEB of P' dilated by $1+\varepsilon$ contains P [26]. The set P' is given by the atoms in P selected by the LMO in the first $\mathcal{O}(\frac{1}{\varepsilon})$ iterations. Further details related to the connections between FW methods and MEB problems can be found in, e.g., [2,26,1] and references therein.

3.7 Training linear Support Vector Machines

Support Vector Machines (SVMs) represent a very important class of machine learning tools (see, e.g., [82] for further details). Given a labeled set of data points, usually called *training set*:

$$TS = \{(\mathsf{p}_i, y_i),\ \mathsf{p}_i \in \mathbb{R}^d,\ y_i \in \{-1, 1\},\ i = 1, \dots, n\},$$

the linear SVM training problem consists in finding a linear classifier $\mathsf{w} \in \mathbb{R}^d$ such that the label y_i can be deduced with the "highest possible confidence" from $\mathsf{w}^\intercal \mathsf{p}_i$. A convex quadratic formulation for this problem is the following [26]:

$$\begin{aligned} \min_{\mathsf{w}\in\mathbb{R}^d,\rho\in\mathbb{R}} &\quad \rho + \tfrac{\|\mathsf{w}\|^2}{2} \\ s.t. &\quad \rho + y_i\,\mathsf{w}^\intercal \mathsf{p}_i \geq 0\,, \quad \text{all } i \in [1\!:\!n]\,, \end{aligned} \tag{19}$$

where the slack variable ρ stands for the negative margin and we can have $\rho < 0$ if and only if there exists an exact linear classifier, i.e. w such that $\mathsf{w}^\intercal \mathsf{p}_i = \text{sign}(y_i)$. The dual of (19) is again an StQP:

$$\min\{\mathsf{x}^\intercal \mathsf{A}^\intercal \mathsf{A}\mathsf{x} : \mathsf{x} \in \Delta_{n-1}\} \tag{20}$$

with $\mathsf{A} = [y_1\mathsf{p}_1, ..., y_n\mathsf{p}_n]$. Notice that problem (20) is equivalent to an MNP problem on $\text{conv}\{y_i\mathsf{p}_i : i \in [1:n]\}$, see Section 7.2 below. Further details on FW methods for SVM training problems can be found in, e.g., [26,52].

3.8 Finding maximal cliques in graphs

In the context of network analysis the clique model, dating back at least to the work of Luce and Perry [69] about social networks, refers to subsets with every two elements in a direct relationship. The problem of finding maximal cliques has numerous applications in domains including telecommunication networks, biochemistry, financial networks, and scheduling (see, e.g., [11,90]). Let $G = (V, E)$ be a simple undirected graph with V and E set of vertices and edges, respectively. A clique in G is a subset $C \subseteq V$ such that $(i, j) \in E$ for each $(i, j) \in C$, with $i \neq j$. The goal in finding a clique C such that $|C|$ is maximal (i.e., it is not contained in any strictly larger clique). This corresponds to find a local minimum for the following equivalent (this time non-convex) StQP (see, e.g., [10,11,51] for further details):

$$\max \left\{ \mathsf{x}^\intercal \mathsf{A}_G \mathsf{x} + \frac{1}{2}\|\mathsf{x}\|^2 : \mathsf{x} \in \Delta_{n-1} \right\} \tag{21}$$

where A_G is the adjacency matrix of G. Due to the peculiar structure of the problem, FW methods can be fruitfully used to find maximal cliques (see, e.g., [51]).

4 Stepsizes

Popular rules for determining the stepsize are:

- *diminishing stepsize*:
$$\alpha_k = \frac{2}{k+2}\,, \tag{22}$$
mainly used for the classic FW (see, e.g., [37,53]);
- *exact line search*:
$$\alpha_k = \min \mathop{\arg\min}_{\alpha \in [0, \alpha_k^{\max}]} \varphi(\alpha) \quad \text{with } \varphi(\alpha) := f(\mathsf{x}_k + \alpha\, \mathsf{d}_k)\,, \tag{23}$$
where we pick the smallest minimizer of the function φ for the sake of being well-defined even in rare cases of ties (see, e.g., [14,62]);
- *Armijo line search:* the method iteratively shrinks the step size in order to guarantee a sufficient reduction of the objective function. It represents a good way to replace exact line search in cases when it gets too costly. In practice, we fix parameters $\delta \in (0, 1)$ and $\gamma \in (0, \frac{1}{2})$, then try steps $\alpha = \delta^m \alpha_k^{\max}$ with $m \in \{0, 1, 2, \dots\}$ until the sufficient decrease inequality
$$f(\mathsf{x}_k + \alpha\, \mathsf{d}_k) \le f(\mathsf{x}_k) + \gamma\alpha\, \nabla f(\mathsf{x}_k)^\intercal \mathsf{d}_k \tag{24}$$
holds, and set $\alpha_k = \alpha$ (see, e.g., [13] and references therein).
- *Lipschitz constant dependent step size:*
$$\alpha_k = \alpha_k(L) := \min \left\{ -\frac{\nabla f(\mathsf{x}_k)^\intercal \mathsf{d}_k}{L\|\mathsf{d}_k\|^2}, \alpha_k^{max} \right\}\,, \tag{25}$$
with L the Lipschitz constant of ∇f (see, e.g., [14,74]).

The Lipschitz constant dependent step size can be seen as the minimizer of the quadratic model $m_k(\cdot; L)$ overestimating f along the line $\mathsf{x}_k + \alpha\,\mathsf{d}_k$:

$$m_k(\alpha; L) = f(\mathsf{x}_k) + \alpha\,\nabla f(\mathsf{x}_k)^\intercal \mathsf{d}_k + \frac{L\alpha^2}{2}\,\|\mathsf{d}_k\|^2 \geq f(\mathsf{x}_k + \alpha\,\mathsf{d}_k)\,, \tag{26}$$

where the inequality follows by the standard Descent Lemma [9, Proposition 6.1.2].

In case L is unknown, it is even possible to approximate L using a backtracking line search (see, e.g., [58, 74]).

We now report a lower bound for the improvement on the objective obtained with the stepsize (25), often used in the convergence analysis.

Lemma 1 *If α_k is given by* (25) *and $\alpha_k < \alpha_k^{\max}$ then*

$$f(\mathsf{x}_{k+1}) \leq f(\mathsf{x}_k) - \frac{1}{2L}(\nabla f(\mathsf{x}_k)^\intercal \widehat{\mathsf{d}}_k)^2\,. \tag{27}$$

Proof We have

$$\begin{aligned} f(\mathsf{x}_k + \alpha_k\,\mathsf{d}_k) &\leq f(\mathsf{x}_k) + \alpha_k \nabla f(\mathsf{x}_k)^\intercal \mathsf{d}_k + \tfrac{L\alpha_k^2}{2}\,\|\mathsf{d}_k\|^2 \\ &= f(\mathsf{x}_k) - \tfrac{(\nabla f(\mathsf{x}_k)^\intercal \mathsf{d}_k)^2}{2L\|\mathsf{d}_k\|^2} = f(\mathsf{x}_k) - \tfrac{1}{2L}(\nabla f(\mathsf{x}_k)^\intercal \widehat{\mathsf{d}_k})^2\,, \end{aligned} \tag{28}$$

where we used the standard Descent Lemma in the inequality. □

5 Properties of the FW method and its variants

5.1 The FW gap

A key parameter often used as a measure of convergence is the FW gap

$$G(\mathsf{x}) = \max_{\mathsf{s} \in C} -\nabla f(\mathsf{x})^\intercal(\mathsf{s} - \mathsf{x})\,, \tag{29}$$

which is always nonnegative and equal to 0 only in first order stationary points. This gap is, by definition, readily available during the algorithm. If f is convex, using that $\nabla f(\mathsf{x})$ is a subgradient we obtain

$$G(\mathsf{x}) \geq -\nabla f(\mathsf{x})^\intercal(\mathsf{x}^* - \mathsf{x}) \geq f(\mathsf{x}) - f^*\,, \tag{30}$$

so that $G(\mathsf{x})$ is an upper bound on the optimality gap at x. Furthermore, $G(\mathsf{x})$ is a special case of the Fenchel duality gap [63].

If $C = \Delta_{n-1}$ is the simplex, then G is related to the Wolfe dual as defined in [26]. Indeed, this variant of Wolfe's dual reads

$$\begin{aligned} &\max\; f(\mathsf{x}) + \lambda(\mathsf{e}^\intercal \mathsf{x} - 1) - \mathsf{u}^\intercal \mathsf{x} \\ &\text{s.t.}\;\; \nabla_i f(\mathsf{x}) - u_i + \lambda = 0\,, \quad i \in [1\!:\!n]\,, \\ &\qquad (\mathsf{x}, \mathsf{u}, \lambda) \in \mathbb{R}^n \times \mathbb{R}^n_+ \times \mathbb{R} \end{aligned} \tag{31}$$

and for a fixed $\mathsf{x} \in \mathbb{R}^n$, the optimal values of (u, λ) are

$$\lambda_\mathsf{x} = -\min_j \nabla_j f(\mathsf{x})\,, \quad u_i(\mathsf{x}) := \nabla_i f(\mathsf{x}) - \min_j \nabla_j f(\mathsf{x}) \geq 0\,.$$

Performing maximization in problem (31) iteratively, first for (u, λ) and then for x, this implies that (31) is equivalent to

$$\begin{aligned} &\max_{\mathsf{x}\in\mathbb{R}^n} \left[f(\mathsf{x}) + \lambda_{\mathsf{x}}(\mathsf{e}^\intercal \mathsf{x} - 1) - \mathsf{u}(\mathsf{x})^\intercal \mathsf{x}\right] \\ = &\max_{\mathsf{x}\in\mathbb{R}^n} \left[f(\mathsf{x}) - \max_j (\mathsf{e}_j - \mathsf{x})^\intercal \nabla f(\mathsf{x})\right] = \max_{\mathsf{x}\in\mathbb{R}^n} \left[f(\mathsf{x}) - G(\mathsf{x})\right] . \end{aligned} \tag{32}$$

Furthermore, since Slater's condition is satisfied, strong duality holds by Slater's theorem [15], resulting in $G(\mathsf{x}^*) = 0$ for every solution x^* of the primal problem.

The FW gap is related to several other measures of convergence (see e.g. [64, Section 7.5.1]). First, consider the projected gradient

$$\widetilde{\mathsf{g}}_k := \pi_C(\mathsf{x}_k - \nabla f(\mathsf{x}_k)) - \mathsf{x}_k\,. \tag{33}$$

with π_B the projection on a convex and closed subset $B \subseteq \mathbb{R}^n$. We have $\|\widetilde{\mathsf{g}}_k\| = 0$ if and only if x_k is stationary, with

$$\begin{aligned} \|\widetilde{\mathsf{g}}_k\|^2 &= \widetilde{\mathsf{g}}_k^\intercal \widetilde{\mathsf{g}}_k \;\le\; \widetilde{\mathsf{g}}_k^\intercal[(\mathsf{x}_k - \nabla f(\mathsf{x}_k)) - \pi_C(\mathsf{x}_k - \nabla f(\mathsf{x}_k))] + \widetilde{\mathsf{g}}_k^\intercal \widetilde{\mathsf{g}}_k \\ &= -\widetilde{\mathsf{g}}_k^\intercal \nabla f(\mathsf{x}_k) \;=\; -(\pi_C(\mathsf{x}_k - \nabla f(\mathsf{x}_k)) - \mathsf{x}_k)^\intercal \nabla f(\mathsf{x}_k) \\ &\le \max_{\mathsf{y}\in C} -(\mathsf{y} - \mathsf{x}_k)^\intercal \nabla f(\mathsf{x}_k) \;=\; G(\mathsf{x}_k)\,, \end{aligned} \tag{34}$$

where we used $[\mathsf{y} - \pi_C(\mathsf{x})]^\intercal[\mathsf{x} - \pi_C(\mathsf{x})] \le 0$ in the first inequality, with $\mathsf{x} = \mathsf{x}_k - \nabla f(\mathsf{x}_k)$ and $\mathsf{y} = \mathsf{x}_k$.
Let now $N_C(x)$ denote the normal cone to C at a point $\mathsf{x} \in C$:

$$N_C(\mathsf{x}) := \{\mathsf{r} \in \mathbb{R}^n : \mathsf{r}^\intercal(\mathsf{y} - \mathsf{x}) \le 0 \;\text{ for all } \mathsf{y} \in C\}\,. \tag{35}$$

First-order stationarity conditions are equivalent to $-\nabla f(\mathsf{x}) \in N_C(\mathsf{x})$, or

$$\operatorname{dist}(N_C(\mathsf{x}), -\nabla f(\mathsf{x})) = \| - \nabla f(\mathsf{x}) - \pi_{N_C(\mathsf{x})}(-\nabla f(\mathsf{x}))\| = 0\,.$$

The FW gap provides a lower bound on the distance from the normal cone $\operatorname{dist}(N_C(\mathsf{x}), -\nabla f(\mathsf{x}))$, inflated by the diameter $D > 0$ of C, as follows:

$$\begin{aligned} G(\mathsf{x}_k) &= -(\mathsf{s}_k - \mathsf{x}_k)^\intercal \nabla f(\mathsf{x}_k) \\ &= (\mathsf{s}_k - \mathsf{x}_k)^\intercal[\pi_{N_C(\mathsf{x}_k)}(-\nabla f(\mathsf{x}_k)) - (\pi_{N_C(\mathsf{x}_k)}(-\nabla f(\mathsf{x}_k)) + \nabla f(\mathsf{x}_k))] \\ &\le \|\mathsf{s}_k - \mathsf{x}_k\|\, \|\pi_{N_C(\mathsf{x}_k)}(-\nabla f(\mathsf{x}_k)) + \nabla f(\mathsf{x}_k)\| \\ &\le D \operatorname{dist}(N_C(\mathsf{x}_k), -\nabla f(\mathsf{x}_k))\,, \end{aligned} \tag{36}$$

where in the first inequality we used $(\mathsf{s}_k - \mathsf{x}_k)^\intercal[\pi_{N_C(\mathsf{x}_k)}(-\nabla f(\mathsf{x}_k))] \le 0$ together with the Cauchy-Schwarz inequality, and $\|\mathsf{s}_k - \mathsf{x}_k\| \le D$ in the second.

5.2 $\mathcal{O}(1/k)$ rate for convex objectives

If f is non-convex, it is possible to prove a $\mathcal{O}(1/\sqrt{k})$ rate for $\min_{i\in[1:k]} G(x_i)$ (see, e.g., [61]). On the other hand, if f is convex, we have an $\mathcal{O}(1/k)$ rate on the optimality gap (see, e.g., [36,67]) for all the stepsizes discussed in Section 4. Here we include a proof for the Lipschitz constant dependent stepsize α_k given by (25).

Theorem 1 *If f is convex and the stepsize is given by* (25)*, then for every $k \geq 1$*

$$f(\mathsf{x}_k) - f^* \leq \frac{2LD^2}{k+2}\,. \tag{37}$$

Before proving the theorem we prove a lemma concerning the decrease of the objective in the case of a full FW step, that is a step with $\mathsf{d}_k = \mathsf{d}_k^{FW}$ and with α_k equal to 1, the maximal feasible stepsize.

Lemma 2 *If $\alpha_k = 1$ and $\mathsf{d}_k = \mathsf{d}_k^{FW}$ then*

$$f(\mathsf{x}_{k+1}) - f^* \leq \frac{1}{2} \min\left\{L\|\mathsf{d}_k\|^2, f(\mathsf{x}_k) - f^*\right\}\,. \tag{38}$$

Proof If $\alpha_k = 1 = \alpha_k^{\max}$ then by Definitions (3) and (29)

$$G(\mathsf{x}_k) = -\nabla f(\mathsf{x}_k)^\intercal \mathsf{d}_k \geq L\|\mathsf{d}_k\|^2\,, \tag{39}$$

the last inequality following by Definition (25) and the assumption that $\alpha_k = 1$. By the standard Descent Lemma it also follows

$$f(\mathsf{x}_{k+1}) - f^* = f(\mathsf{x}_k + \mathsf{d}_k) - f^* \leq f(\mathsf{x}_k) - f^* + \nabla f(\mathsf{x}_k)^\intercal \mathsf{d}_k + \frac{L}{2}\|\mathsf{d}_k\|^2\,. \tag{40}$$

Considering the definition of d_k and convexity of f, we get

$$f(\mathsf{x}_k) - f^* + \nabla f(\mathsf{x}_k)^\intercal \mathsf{d}_k \leq f(\mathsf{x}_k) - f^* + \nabla f(\mathsf{x}_k)^\intercal(\mathsf{x}^* - \mathsf{x}_k) \leq 0\,,$$

so that (40) entails $f(\mathsf{x}_{k+1}) - f^* \leq \frac{L}{2}\|\mathsf{d}_k\|^2$. To conclude, it suffices to apply to the RHS of (40) the inequality

$$f(\mathsf{x}_k) - f^* + \nabla f(\mathsf{x}_k)^\intercal \mathsf{d}_k + \tfrac{L}{2}\|\mathsf{d}_k\|^2 \leq f(\mathsf{x}_k) - f^* - \tfrac{1}{2}G(\mathsf{x}_k) \leq \tfrac{f(\mathsf{x}_k) - f^*}{2} \tag{41}$$

where we used (39) in the first inequality and $G(\mathsf{x}_k) \geq f(\mathsf{x}_k) - f^*$ in the second. □

We can now proceed with the proof of the main result.

Proof (Theorem 1) For $k = 0$ and $\alpha_0 = 1$ then by Lemma 2

$$f(\mathsf{x}_1) - f^* \leq \frac{L\|\mathsf{d}_0\|^2}{2} \leq \frac{LD^2}{2}\,. \tag{42}$$

If $\alpha_0 < 1$ then

$$f(\mathsf{x}_0) - f^* \leq G(\mathsf{x}_0) < L\|\mathsf{d}_0\|^2 \leq LD^2\,. \tag{43}$$

Therefore in both cases (27) holds for $k = 0$.
Reasoning by induction, if (37) holds for k with $\alpha_k = 1$, then the claim is clear by (38). On the other hand, if $\alpha_k < \alpha_k^{\max} = 1$ then by Lemma 1, we have

$$\begin{aligned} f(\mathsf{x}_{k+1}) - f^* &\leq f(\mathsf{x}_k) - f^* - \tfrac{1}{2L}(\nabla f(x_k)^\intercal \widehat{\mathsf{d}}_k)^2 \\ &\leq f(\mathsf{x}_k) - f^* - \tfrac{(\nabla f(\mathsf{x}_k)^\intercal \mathsf{d}_k)^2}{2LD^2} \\ &\leq f(\mathsf{x}_k) - f^* - \tfrac{(f(\mathsf{x}_k) - f^*)^2}{2LD^2} \\ &= (f(\mathsf{x}_k) - f^*)(1 - \tfrac{f(\mathsf{x}_k) - f^*}{2LD^2}) \leq \tfrac{2LD^2}{k+3}\,, \end{aligned} \tag{44}$$

where we used $\|\mathsf{d}_k\| \leq D$ in the second inequality, $\nabla f(\mathsf{x}_k)^\intercal \mathsf{d}_k = G(\mathsf{x}_k) \geq f(\mathsf{x}_k) - f^*$ in the third inequality; and the last inequality follows by induction hypothesis. □

As can be easily seen from above argument, the convergence rate of $\mathcal{O}(1/k)$ is true also in more abstract normed spaces than $\mathbb{R}^n$, e.g. when C is a convex and weakly compact subset of a Banach space (see, e.g., [30,34]). A generalization for some unbounded sets is given in [35]. The bound is tight due to a zigzagging behaviour of the method near solutions on the boundary, leading to a rate of $\Omega(1/k^{1+\delta})$ for every $\delta > 0$ (see [19] for further details), when the objective is a strictly convex quadratic function and the domain is a polytope.
Also the minimum FW gap $\min_{i \in [0:k]} G(\mathsf{x}_i)$ converges at a rate of $\mathcal{O}(1/k)$ (see [53,37]). In [37], a broad class of stepsizes is examined, including $\alpha_k = \frac{1}{k+1}$ and $\alpha_k = \bar{\alpha}$ constant. For these stepsizes a convergence rate of $\mathcal{O}\left(\frac{\ln(k)}{k}\right)$ is proved.

5.3 Variants

Active set FW variants mostly aim to improve over the $\mathcal{O}(1/k)$ rate and also ensure support identification in finite time. They generate a sequence of active sets $\{A_k\}$, such that $\mathsf{x}_k \in \mathrm{conv}(A_k)$, and define alternative directions making use of these active sets.

For the *pairwise FW (PFW)* and the *away step FW (AFW)* (see [26,62]) we have that A_k must always be a subset of S_k, with x_k a convex combination of the elements in A_k. The away vertex v_k is then defined by

$$\mathsf{v}_k \in \underset{\mathsf{y} \in A_k}{\arg\max}\, \nabla f(\mathsf{x}_k)^\intercal \mathsf{y}\,. \tag{45}$$

The AFW direction, introduced in [88], is hence given by

$$\begin{aligned} \mathsf{d}_k^{AS} &= \mathsf{x}_k - \mathsf{v}_k \\ \mathsf{d}_k &\in \arg\max\{-\nabla f(\mathsf{x}_k)^\intercal \mathsf{d} : \mathsf{d} \in \{\mathsf{d}_k^{AS}, \mathsf{d}_k^{FW}\}\}\,, \end{aligned} \tag{46}$$

while the PFW direction, as defined in [62] and inspired by the early work [70], is

$$\mathsf{d}_k^{PFW} = \mathsf{d}_k^{FW} + \mathsf{d}_k^{AS} = \mathsf{s}_k - \mathsf{v}_k\,, \tag{47}$$

with s_k defined in (3).

The *FW method with in-face directions (FDFW)* (see [38,46]), also known as Decomposition invariant Conditional Gradient (DiCG) when applied to polytopes [5], is defined exactly as the AFW, but with the minimal face $\mathcal{F}(\mathsf{x}_k)$ of C containing x_k as the active set. The *extended FW (EFW)* was introduced in [50] and is also known as simplicial decomposition [83]. At every iteration the method minimizes the objective in the current active set A_{k+1}

$$\mathsf{x}_{k+1} \in \underset{\mathsf{y} \in \mathrm{conv}(A_{k+1})}{\arg\min} f(\mathsf{y})\,, \tag{48}$$

where $A_{k+1} \subseteq A_k \cup \{s_k\}$ (see, e.g., [26], Algorithm 4.2). A more general version of the EFW, only approximately minimizing on the current active set, was introduced in [62] under the name of fully corrective FW. In Table 1, we report the main features of the classic FW and of the variants under analysis.

Variant	Direction	Active set
FW	$d_k = d_k^{FW} = s_k - x_k, \quad s_k \in \arg\max\{\nabla f(x_k)^\intercal x : x \in C\}$	-
AFW	$d_k \in \arg\max\{-\nabla f(x_k)^\intercal d : d \in \{x_k - v_k, d_k^{FW}\},\ v_k \in A_k\}$	$A_{k+1} \subseteq A_k \cup \{s_k\}$
PFW	$d_k = s_k - v_k, \quad v_k \in \arg\max\{\nabla f(x_k)^\intercal v_k : v_k \in A_k\}$	$A_{k+1} \subseteq A_k \cup \{s_k\}$
EFW	$d_k = y_k - x_k, \quad y_k \in \arg\min\{f(y) : y \in \mathrm{conv}(A_k)\}$	$A_{k+1} \subseteq A_k \cup \{s_k\}$
FDFW	$d_k \in \arg\max\{-\nabla f(x_k)^\intercal d : d \in \{x_k - v_k, d_k^{FW}\},\ v_k \in A_k\}$	$A_k = \mathcal{F}(x_k)$

Table 1: FW method and variants covered in this review.

5.4 Sparse approximation properties

As discussed in the previous section, for the classic FW method and the AFW, PFW, EFW variants x_k can always be written as a convex combination of elements in $A_k \subset S_k$, with $|A_k| \leq k+1$. Even for the FDFW we still have the weaker property that x_k must be an affine combination of elements in $A_k \subset A$ with $|A_k| \leq k+1$. It turns out that the convergence rate of methods with this property is $\Omega(\frac{1}{k})$ in high dimension. More precisely, if $C = \mathrm{conv}(A)$ with A compact, the $\mathcal{O}(1/k)$ rate of the classic FW method is worst case optimal given the sparsity constraint

$$x_k \in \mathrm{aff}(A_k) \text{ with } A_k \subset A,\ |A_k| \leq k+1\,. \tag{49}$$

An example where the $\mathcal{O}(1/k)$ rate is tight was presented in [53]. Let $C = \Delta_{n-1}$ and $f(x) = \|x - \frac{1}{n} e\|^2$. Clearly, $f^* = 0$ with $x^* = \frac{1}{n} e$. Then it is easy to see that $\min\{f(x) - f^* : \|x\|_0 \leq k+1\} \geq \frac{1}{k+1} - \frac{1}{n}$ for every $k \in \mathbb{N}$, so that in particular under (49) with $A_k = \{e_i : i \in [1\!:\!n]\}$, the rate of any FW variant must be $\Omega(\frac{1}{k})$.

5.5 Affine invariance

The FW method and the AFW, PFW, EFW are affine invariant [53]. More precisely, let P be a linear transformation, $\hat{f}$ be such that $\hat{f}(Px) = f(x)$ and $\hat{C} = P(C)$. Then for every sequence $\{x_k\}$ generated by the methods applied to (f, C), the sequence $\{y_k\} := \{Px_k\}$ can be generated by the FW method with the same stepsizes applied to $(\hat{f}, \hat{C})$. As a corollary, considering the special case where P is the matrix collecting the elements of A as columns, one can prove results on $C = \Delta_{|A|-1}$ and generalize them to $\hat{C} := \mathrm{conv}(A)$ by affine invariance.

An affine invariant convergence rate bound for convex objectives can be given using the curvature constant

$$\kappa_{f,C} := \sup\left\{2\frac{f(\alpha y+(1-\alpha)x)-f(x)-\alpha\nabla f(x)^\intercal(y-x)}{\alpha^2} : \{x, y\} \subset C,\ \alpha \in (0,1]\right\}\,. \tag{50}$$

It is easy to prove that $\kappa_{f,C} \leq LD^2$ if D is the diameter of C. In the special case where $C = \Delta_{n-1}$ and $f(x) = x^\intercal \tilde{A}^\intercal \tilde{A} x + b^\intercal x$, then $\kappa_{f,C} \leq \mathrm{diam}(A\Delta_{n-1})^2$ for $A^\intercal = [\tilde{A}^\intercal, b]$; see [26].

When the method uses the stepsize sequence (22), it is possible to give the following affine invariant convergence rate bounds (see [37]):

$$\begin{aligned} f(\mathsf{x}_k) - f^* &\leq \frac{2\kappa_{f,C}}{k+4}\,, \\ \min_{i\in]0:k]} G(\mathsf{x}_i) &\leq \frac{9\kappa_{f,C}}{2k}\,, \end{aligned} \tag{51}$$

thus in particular slightly improving the rate we gave in Theorem 1 since we have that $\kappa_{f,C} \leq LD^2$.

5.6 Support identification for the AFW

It is a classic result that the AFW under some strict complementarity conditions and for strongly convex objectives identifies in finite time the face containing the solution [46]. Here we report some explicit bounds for this property proved in [14]. We first assume that $C = \Delta_{n-1}$, and introduce the multiplier functions

$$\lambda_i(\mathsf{x}) = \nabla f(\mathsf{x})^\intercal(\mathsf{e}_i - \mathsf{x}) \tag{52}$$

for $i \in [1\!:\!n]$. Let x^* be a stationary point for f, with the objective f not necessarily convex. It is easy to check that $\{\lambda_i(\mathsf{x}^*)\}_{i\in[1:n]}$ coincide with the Lagrangian multipliers. Furthermore, by complementarity conditions we have $x_i^*\lambda_i(\mathsf{x}^*) = 0$ for every $i \in [1\!:\!n]$. It follows that the set

$$I(\mathsf{x}^*) := \{i \in [1\!:\!n] : \lambda_i(\mathsf{x}^*) = 0\}$$

contains the support of x^*,

$$\mathrm{supp}(\mathsf{x}^*) := \{i \in [1\!:\!n] : x_i^* > 0\}\,.$$

The next lemma uses λ_i, and the Lipschitz constant L of ∇f, to give a lower bound of the so-called *active set radius* r_*, defining a neighborhood of x^*. Starting the algorithm in this neighbourhood, the active set (the minimal face of C containing x^*) is identified in a limited number of iterations.

Lemma 3 *Let x^* be a stationary point for f on the boundary of Δ_{n-1}, $\delta_{\min} = \min_{i:\lambda_i(\mathsf{x}^*)>0}\lambda_i(\mathsf{x}^*)$ and*

$$r_* = \frac{\delta_{\min}}{\delta_{\min} + 2L}\,. \tag{53}$$

Let. Assume that for every k for which $\mathsf{d}_k = \mathsf{d}_k^{\mathcal{A}}$ holds, the step size α_k is not smaller than the stepsize given by (25), $\alpha_k(L) \leq \alpha_k$.
If $\|\mathsf{x}_k - \mathsf{x}^*\|_1 < r_*$, then for some

$$j \leq \min\{n - |I(\mathsf{x}^*)|, |\,\mathrm{supp}(\mathsf{x}_k)| - 1\}$$

we have $\mathrm{supp}(\mathsf{x}_{k+j}) \subseteq I(\mathsf{x}^*)$ *and* $\|\mathsf{x}_{k+j} - \mathsf{x}^*\|_1 < r_*$.

Proof Follows from [14, Theorem 3.3], since under the assumptions the AFW sets one variable in $\mathrm{supp}(x_k)\backslash I(\mathsf{x}^*)$ to zero at every step without increasing the 1-norm distance from x^*. □

The above lemma does not require convexity and was applied in [14] to derive active set identification bounds in several convex and non-convex settings. Here we focus on the case where the domain $C = \mathrm{conv}(A)$ with $|A| < +\infty$ is a generic polytope, and where f is μ-strongly convex for some $\mu > 0$, i.e.

$$f(\mathsf{y}) \geq f(\mathsf{x}) + \nabla f(\mathsf{x})^{\intercal}(\mathsf{y}-\mathsf{x}) + \frac{\mu}{2}\|\mathsf{x}-\mathsf{y}\|^2 \quad \text{for all } \{\mathsf{x},\mathsf{y}\} \subset C\,. \tag{54}$$

Let $E_C(\mathsf{x}^*)$ be the face of C exposed by $\nabla f(x^*)$:

$$E_C(\mathsf{x}^*) := \underset{\mathsf{x}\in C}{\arg\min}\, \nabla f(\mathsf{x}^*)^{\intercal}\mathsf{x}\,, \tag{55}$$

Let then θ_A be the Hoffman constant (see [7]) related to $[\bar{\mathsf{A}}^{\intercal}, \mathsf{I}_n, \mathsf{e}, -\mathsf{e}]^{\intercal}$, with $\bar{\mathsf{A}}$ the matrix having as columns the elements in A. Finally, consider the function $f_A(\mathsf{y}) := f(\bar{\mathsf{A}}\mathsf{y})$ on $\Delta_{|A|-1}$, and let L_A be the Lipschitz constant of ∇f_A as well as

$$\delta_{\min} := \min_{\mathsf{a}\in A\setminus E_C(\mathsf{x}^*)} \nabla f(\mathsf{x}^*)^{\intercal}(\mathsf{a}-\mathsf{x}^*) \quad \text{and} \quad r_*(\mathsf{x}^*) := \frac{\delta_{\min}}{\delta_{\min}+2L_A}\,.$$

Using linearity of AFW convergence for strongly convex objectives (see Section 6.1), we have the following result:

Theorem 2 *The sequence $\{\mathsf{x}_k\}$ generated by the AFW with $\mathsf{x}_0 \in A$ enters $E_C(\mathsf{x}^*)$ for*

$$k \geq \max\left\{2\frac{\ln(h_0) - \ln(\mu_A r_*(\mathsf{x}^*)^2/2)}{\ln(1/q)}, 0\right\}\,, \tag{56}$$

where $\mu_A = \frac{\mu}{n\theta_A^2}$ and $q \in (0,1)$ is the constant related to the linear convergence rate of the AFW, i.e. $h_k \leq q^k h_0$ for all k.

Proof (sketch) We present an argument in the case $C = \Delta_{n-1}$, $A = \{e_i\}_{i\in[1:n]}$ which can be easily extended by affine invariance to the general case (see [14] for details). In this case $\theta_A \geq 1$ and we can define $\bar{\mu} := \mu/n \geq \mu_A$.
To start with, the number of steps needed to reach the condition

$$h_k \leq \frac{\mu}{2n} r_*(\mathsf{x}^*)^2 = \frac{\bar{\mu}}{2} r_*(\mathsf{x}^*)^2 \tag{57}$$

is at most

$$\bar{k} = \max\left\{\left\lceil \frac{\ln(h_0) - \ln(\bar{\mu} r_*(x^*)^2/2)}{\ln(1/q)} \right\rceil, 0\right\}\,.$$

Now we combine $n\|\cdot\| \geq \|\cdot\|_1$ with strong convexity and relation (57) to obtain $\|\mathsf{x}_k - \mathsf{x}^*\|_1 \leq r_*(\mathsf{x}^*)$, hence in particular $\|\mathsf{x}_k - \mathsf{x}^*\|_1 \leq r_*(\mathsf{x}^*)$ for every $k \geq \bar{k}$. Since x_0 is a vertex of the simplex, and at every step at most one coordinate is added to the support of the current iterate, $|\operatorname{supp}(\mathsf{x}_{\bar{k}})| \leq \bar{k}+1$. The claim follows by applying Lemma 3. □

Additional bounds under a quadratic growth condition weaker than strong convexity and strict complementarity are reported in [42].

Convergence and finite time identification for the PFW and the AFW are proved in [13] for a specific class of non-convex minimization problems over the standard simplex, under the additional assumption that the sequence generated has a finite set of limit points. In another line of work, active set identification strategies combined with FW variants have been proposed in [29] and [80].

5.7 Inexact linear oracle

In many real-world applications, linear subproblems can only be solved approximately. This is the reason why the convergence of FW variants is often analyzed under some error term for the linear minimization oracle (see, e.g., [16,17,37,53, 59]). A common assumption, relaxing the FW vertex exact minimization property, is to have access to a point (usually a vertex) $\tilde{\mathsf{s}}_k$ such that

$$\nabla f(\mathsf{x}_k)^\intercal(\tilde{\mathsf{s}}_k - \mathsf{x}_k) \leq \min_{\mathsf{s}\in C} \nabla f(\mathsf{x}_k)^\intercal(\mathsf{s} - \mathsf{x}_k) + \delta_k \, , \tag{58}$$

for a sequence $\{\delta_k\}$ of non negative approximation errors.
If the sequence $\{\delta_k\}$ is constant and equal to some $\delta > 0$, then trivially the lowest possible approximation error achieved by the FW method is δ. At the same time, [37, Theorem 5.1] implies a rate of $\mathcal{O}(\frac{1}{k} + \delta)$ if the stepsize $\alpha_k = \frac{2}{k+2}$ is used.

The $\mathcal{O}(1/k)$ rate can be instead retrieved by assuming that $\{\delta_k\}$ converges to 0 quickly enough, and in particular if

$$\delta_k = \frac{\delta \kappa_{f,C}}{k+2} \tag{59}$$

for a constant $\delta > 0$. Under (59), in [53] a convergence rate of

$$f(x_k) - f^* \leq \frac{2\kappa_{f,C}}{k+2}(1+\delta) \tag{60}$$

was proved for the FW method with α_k given by exact line search or equal to $\frac{2}{k+2}$, as well as for the EFW.

A linearly convergent variant making use of an approximated linear oracle recycling previous solutions to the linear minimization subproblem is studied in [16]. In [37,49], the analysis of the classic FW method is extended to the case of inexact gradient information. In particular in [37], assuming the availability of the (δ, L) oracle introduced in [31], a convergence rate of $\mathcal{O}(1/k + \delta k)$ is proved.

6 Improved rates for strongly convex objectives

6.1 Linear convergence under an angle condition

In the rest of this section we assume that f is μ-strongly convex (54). We also assume that the stepsize is given by exact linesearch or by (25).

Under this assumption, an asymptotic linear convergence rate for the FDFW on polytopes was given in the early work [46]. Furthermore, in [44] a linearly convergent variant was proposed, making use however of an additional local linear minimization oracle.

Recent works obtain linear convergence rates by proving the angle condition

$$-\nabla f(\mathsf{x}_k)^\intercal \widehat{\mathsf{d}}_k \geq \frac{\tau}{\|\mathsf{x}_k - \mathsf{x}^*\|} \nabla f(\mathsf{x}_k)^\intercal(\mathsf{x}_k - \mathsf{x}^*) \tag{61}$$

for some $\tau > 0$ and some $\mathsf{x}^* \in \arg\min_{\mathsf{x}\in C} f(\mathsf{x})$. As we shall see in the next lemma, under (61) it is not difficult to prove linear convergence rates in the number of *good steps*. These are FW steps with $\alpha_k = 1$ and steps in any descent direction with $\alpha_k < 1$.

Method	Objective	Domain	Assumptions	Rate
FW	NC	Generic	-	$\mathcal{O}(1/\sqrt{k})$
FW	C	Generic	-	$\mathcal{O}(1/k)$
FW	SC	Generic	$\mathsf{x}^* \in \mathrm{ri}(\Omega)$	Linear
Variants	SC	Polytope	-	Linear
FW	SC	Strongly convex	-	$\mathcal{O}(1/k^2)$
FW	SC	Strongly convex	$\min \lVert\nabla f(\mathsf{x})\rVert > 0$	Linear

Table 2: Known convergence rates for the FW method and the variants covered in this review. NC, C and SC stand for non-convex, convex and strongly convex respectively.

Lemma 4 *If the step k is a good step and* (61) *holds, then*

$$h_{k+1} \leq \max\left\{\tfrac{1}{2}, 1 - \tfrac{\tau^2\mu}{L}\right\} h_k \,. \tag{62}$$

Proof If the step k is a full FW step then Lemma 2 entails $h_{k+1} \leq \frac{1}{2} h_k$. In the remaining case, first observe that by strong convexity

$$\begin{aligned} f^* &= f(\mathsf{x}^*) \geq f(\mathsf{x}_k) + \nabla f(\mathsf{x}_k)^\intercal(\mathsf{x}^* - \mathsf{x}_k) + \tfrac{\mu}{2}\lVert\mathsf{x}_k - \mathsf{x}^*\rVert^2 \\ &\geq \min_{\alpha\in\mathbb{R}} \left[f(\mathsf{x}_k) + \alpha\nabla f(\mathsf{x}_k)^\intercal(\mathsf{x}^* - \mathsf{x}_k) + \tfrac{\alpha^2\mu}{2}\lVert\mathsf{x}_k - \mathsf{x}^*\rVert^2\right] \\ &= f(\mathsf{x}_k) - \tfrac{1}{2\mu\lVert\mathsf{x}_k - \mathsf{x}^*\rVert^2}\left[\nabla f(\mathsf{x}_k)^\intercal(\mathsf{x}_k - \mathsf{x}^*)\right]^2 \,, \end{aligned} \tag{63}$$

which means

$$h_k \leq \frac{1}{2\mu\lVert\mathsf{x}_k - \mathsf{x}^*\rVert^2}\left[\nabla f(\mathsf{x}_k)^\intercal(\mathsf{x}_k - \mathsf{x}^*)\right]^2 \,. \tag{64}$$

We can then proceed using the bound (27) from Lemma 1 in the following way:

$$\begin{aligned} h_{k+1} &= f(\mathsf{x}_{k+1}) - f^* \leq f(\mathsf{x}_k) - f^* - \tfrac{1}{2L}\left[\nabla f(\mathsf{x}_k)^\intercal\widehat{\mathsf{d}}_k\right]^2 \\ &\leq h_k - \tfrac{\tau^2}{2L\lVert\mathsf{x}_k - \mathsf{x}^*\rVert^2}\left[\nabla f(\mathsf{x}_k)^\intercal(\mathsf{x}_k - \mathsf{x}^*)\right]^2 \\ &\leq h_k\left(1 - \tfrac{\tau^2\mu}{L}\right) \,, \end{aligned} \tag{65}$$

where we used (61) in the second inequality and (64) in the third one. □

As a corollary, under (61) we have the rate

$$f(\mathsf{x}_k) - f^* = h_k \leq \max\left\{\tfrac{1}{2}, 1 - \tfrac{\tau^2\mu}{L}\right\}^{\gamma(k)} h_0 \tag{66}$$

for any method with non increasing $\{f(\mathsf{x}_k)\}$ and following Algorithm 1, with $\gamma(k) \leq k$ an integer denoting the number of good steps until step k. It turns out that for all the variants we introduced in this review we have $\gamma(k) \geq Tk$ for some constant $T > 0$. When x^* is in the relative interior of C, the FW method satisfies (61) and we have the following result (see [46,62]):

Theorem 3 *If* $\mathsf{x}^* \in \mathrm{ri}(C)$*, then*

$$f(\mathsf{x}_k) - f^* \leq \left[1 - \frac{\mu}{L}\left(\frac{\mathrm{dist}(\mathsf{x}^*, \partial C)}{D}\right)^2\right]^k (f(\mathsf{x}_0) - f^*) \,. \tag{67}$$

Proof We can assume for simplicity $\mathrm{int}(C) \neq \emptyset$, since otherwise we can restrict ourselves to the affine hull of C. Let $\delta = \mathrm{dist}(\mathsf{x}^*, \partial C)$ and $\mathsf{g} = -\nabla f(\mathsf{x}_k)$. First, by assumption we have $\mathsf{x}^* + \delta\widehat{\mathsf{g}} \in C$. Therefore

$$\mathsf{g}^\intercal \mathsf{d}_k^{FW} \geq \mathsf{g}^\intercal((\mathsf{x}^* + \delta\widehat{\mathsf{g}}) - \mathsf{x}) = \delta\mathsf{g}^\intercal\widehat{\mathsf{g}} + \mathsf{g}^\intercal(\mathsf{x}^* - \mathsf{x}) \geq \delta\|\mathsf{g}\| + f(\mathsf{x}) - f^* \geq \delta\|\mathsf{g}\| \,, \tag{68}$$

where we used $\mathsf{x}^* + \delta\widehat{\mathsf{g}} \in C$ in the first inequality and convexity in the second. We can conclude

$$\mathsf{g}^\intercal \frac{\mathsf{d}_k^{FW}}{\|\mathsf{d}_k^{FW}\|} \geq \mathsf{g}^\intercal \frac{\mathsf{d}_k^{FW}}{D} \geq \frac{\delta}{D}\|\mathsf{g}\| \geq \frac{\delta}{D}\mathsf{g}^\intercal\left(\frac{\mathsf{x}_k - \mathsf{x}^*}{\|\mathsf{x}_k - \mathsf{x}^*\|}\right) \,. \tag{69}$$

The thesis follows by Lemma 4, noticing that for $\tau = \frac{\mathrm{dist}(\mathsf{x}^*, \partial C)}{D} \leq \frac{1}{2}$ we have $1 - \tau^2\frac{\mu}{L} > \frac{1}{2}$. □

In [62], the authors proved that directions generated by the AFW and the PFW on polytopes satisfy condition (61), with $\tau = \mathrm{PWidth}(A)/D$ and $\mathrm{PWidth}(A)$, pyramidal width of A. While $\mathrm{PWidth}(A)$ was originally defined with a rather complex minmax expression, in [73] it was then proved

$$\mathrm{PWidth}(A) = \min_{F \in \mathrm{faces}(C)} \mathrm{dist}(F, \mathrm{conv}(A \setminus F)) \,. \tag{70}$$

This quantity can be explicitly computed in a few special cases. For $A = \{0,1\}^n$ we have $\mathrm{PWidth}(A) = 1/\sqrt{n}$, while for $A = \{e_i\}_{i \in [1:n]}$ (so that C is the $n-1$ dimensional simplex)

$$\mathrm{PWidth}(A) = \begin{cases} \frac{2}{\sqrt{n}} & \text{if } n \text{ is even} \\ \frac{2}{\sqrt{n - 1/n}} & \text{if } n \text{ is odd.} \end{cases} \tag{71}$$

Angle conditions like (61) with τ dependent on the number of vertices used to represent x_k as a convex combination were given in [5] and [7] for the FDFW and the PFW respectively. In particular, in [7] a geometric constant Ω_C called vertex-facet distance was defined as

$$\Omega_C = \min\{\mathrm{dist}(\mathsf{v}, H) : \mathsf{v} \in V(C)\,, H \in \mathcal{H}(C), \, \mathsf{v} \notin H\} \,, \tag{72}$$

with $V(C)$ the set of vertices of C, and $\mathcal{H}(C)$ the set of supporting hyperplanes of C (containing a facet of C). Then condition (61) is satisfied for $\tau = \Omega_C/s$, with d_k the PFW direction and s the number of points used in the active set A_k.
In [5], a geometric constant H_s was defined depending on the minimum number s of vertices needed to represent the current point x_k, as well as on the proper[2] inequalities $\mathsf{q}_i^\intercal \mathsf{x} \leq b_i$, $i \in [1 : m]$, appearing in a description of C. For each of these inequalities the *second gap* g_i was defined as

$$g_i = \max_{\mathsf{v} \in V(C)} \mathsf{q}_i^\intercal \mathsf{v} - \operatorname*{secondmax}_{\mathsf{v} \in V(C)} \mathsf{q}_i^\intercal \mathsf{v} \,, \quad i \in [1 : m] \,, \tag{73}$$

[2] i.e., those inequalities strictly satisfied for some $\mathsf{x} \in C$.

with the secondmax function giving the second largest value achieved by the argument. Then H_s is defined as

$$H_s := \max \left\{ \sum_{j=1}^{n} \left(\sum_{i \in S} \frac{a_{ij}}{g_i} \right)^2 : S \in \binom{[1:m]}{s} \right\}. \tag{74}$$

The arguments used in the paper imply that (61) holds with $\tau = \frac{2}{D\sqrt{H_s}}$ if d_k is a FDFW direction and x_k the convex combination of at most s vertices. We refer the reader to [73] and [75] for additional results on these and related constants.

The linear convergence results for strongly convex objectives are extended to compositions of strongly convex objectives with affine transformations in [7], [62], [73]. In [47], the linear convergence results for the AFW and the FW method with minimum in the interior are extended with respect to a generalized condition number $L_{f,C,D}/\mu_{f,C,D}$, with D a distance function on C.

For the AFW, the PFW and the FDFW, linear rates with no bad steps ($\gamma(k) = k$) are given in [76] for non-convex objectives satisfying a Kurdyka-Łojasiewicz inequality. In [77], condition (61) was proved for the FW direction and orthographic retractions on some convex sets with smooth boundary. The work [27] introduces a new FW variant using a subroutine to align the descent direction with the projection on the tangent cone of the negative gradient, thus implicitly maximizing τ in (61).

6.2 Strongly convex domains

When C is strongly convex we have a $\mathcal{O}(1/k^2)$ rate (see, e.g., [43,57]) for the classic FW method. Furthermore, when C is β_C-strongly convex and $\|\nabla f(\mathsf{x})\| \geq c > 0$, then we have the linear convergence rate (see [30,33,58,67])

$$h_{k+1} \leq \max \left\{ \frac{1}{2}, 1 - \frac{L}{2c\beta_C} \right\} h_k \,. \tag{75}$$

Finally, it is possible to interpolate between the $\mathcal{O}(1/k^2)$ rate of the strongly convex setting and the $\mathcal{O}(1/k)$ rate of the general convex one by relaxing strong convexity of the objective with Hölderian error bounds [91] and also by relaxing strong convexity of the domain with uniform convexity [57].

7 Extensions

7.1 Block coordinate Frank-Wolfe method

The block coordinate FW (BCFW) was introduced in [63] for block product domains of the form $C = C^{(1)} \times ... \times C^{(m)} \subseteq \mathbb{R}^{n_1 + \dots + n_m}$, and applied to structured SVM training. The algorithm operates by selecting a random block and performing a FW step in that block. Formally, for $\mathsf{s} \in \mathbb{R}^{m_i}$ let $\mathsf{s}^{(i)} \in \mathbb{R}^n$ be the vector with all blocks equal to o except for the i-th block equal to s. We can write the direction of the BCFW as

$$\begin{aligned} \mathsf{d}_k &= \mathsf{s}_k^{(i)} - \mathsf{x}_k \\ \mathsf{s}_k &\in \operatorname*{arg\,min}_{\mathsf{s} \in C^{(i)}} \nabla f(\mathsf{x}_k)^\intercal \mathsf{s}^{(i)} \end{aligned} \tag{76}$$

for a random index $i \in [1\!:\!n]$.
In [63], a convergence rate of

$$\mathbb{E}[f(x_k)] - f^* \leq \frac{2Km}{k+2m} \tag{77}$$

is proved, for $K = h_0 + \kappa_f^{\otimes}$, with $\kappa_f^{\otimes}$ the product domain curvature constant, defined as $\kappa_f^{\otimes} = \sum \kappa_f^{\otimes,i}$ where $\kappa_f^{\otimes,i}$ are the curvature constants of the objective fixing the blocks outside $C^{(i)}$:

$$\kappa_f^{\otimes,i} := \sup \left\{ 2\tfrac{f(\mathsf{x}+\alpha \mathsf{d}^{(i)}) - f(\mathsf{x}) - \alpha \nabla f(\mathsf{x})^\intercal \mathsf{d}^{(i)}}{\alpha^2} : \mathsf{d} \in C - \mathsf{x},\, \mathsf{x} \in C,\, \alpha \in (0,1] \right\} . \tag{78}$$

An asynchronous and parallel generalization for this method was given in [85]. This version assumes that a cloud oracle is available, modeling a set of worker nodes each sending information to a server at different times. This information consists of an index i and the following LMO on $C^{(i)}$:

$$\mathsf{s}_{(i)} \in \operatorname*{arg\,min}_{\mathsf{s} \in C^{(i)}} \nabla f(\mathsf{x}_{\widetilde{k}})^\intercal \mathsf{s}^{(i)} \, . \tag{79}$$

The algorithm is called asynchronous because $\widetilde{k}$ can be smaller than k, modeling a delay in the information sent by the node. Once the server has collected a minibatch S of τ distinct indexes (overwriting repetitions), the descent direction is defined as

$$\mathsf{d}_k = \sum_{i \in S} \mathsf{s}_{(i)}^{(i)} \, , \tag{80}$$

If the indices sent by the nodes are i.i.d., then under suitable assumptions on the delay, a convergence rate of

$$\mathbb{E}[f(\mathsf{x}_k)] - f^* \leq \frac{2mK_\tau}{\tau^2 k + 2m} \tag{81}$$

can be proved, where $K_\tau = m\kappa_{f,\tau}^{\otimes}(1+\delta) + h_0$ for δ depending on the delay error, with $\kappa_{f,\tau}^{\otimes}$ the average curvature constant in a minibatch keeping all the components not in the minibatch fixed.

In [72], several improvements are proposed for the BCFW, including an adaptive criterion to prioritize blocks based on their FW gap, and block coordinate versions of the AFW and the PFW variants.

In [79], a multi plane BCFW approach is proposed in the specific case of the structured SVM, based on caching supporting planes in the primal, corresponding to block linear minimizers in the dual. In [8], the duality for structured SVM between BCFW and stochastic subgradient descent is exploited to define a learning rate schedule for neural networks based only on one hyper parameter. The block coordinate approach is extended to the generalized FW in [6], with coordinates however picked in a cyclic order.

7.2 Variants for the min-norm point problem

Consider the min-norm point (MNP) problem

$$\min_{\mathsf{x}\in C} \|\mathsf{x}\|_* \,, \tag{82}$$

with C a closed convex subset of $\mathbb{R}^n$ and $\|\cdot\|_*$ a norm on $\mathbb{R}^n$. In [89], a FW variant is introduced to solve the problem when C is a polytope and $\|\cdot\|_*$ is the standard Euclidean norm $\|\cdot\|$. Similarly to the variants introduced in Section 5.3, it generates a sequence of active sets $\{A_k\}$ with $\mathsf{s}_k \in A_{k+1}$. At the step k the norm is minimized on the affine hull $\mathrm{aff}(A_k)$ of the current active set A_k, that is

$$\mathsf{v}_k = \operatorname*{arg\,min}_{\mathsf{y}\in \mathrm{aff}(A_k)} \|\mathsf{y}\| \,. \tag{83}$$

The descent direction d_k is then defined as

$$\mathsf{d}_k = \mathsf{v}_k - \mathsf{x}_k \,, \tag{84}$$

and the stepsize is given by a tailored linesearch that allows to remove some of the atoms in the set A_k (see, e.g. [62,89]). Whenever x_{k+1} is in the relative interior of $\mathrm{conv}(A_k)$, the FW vertex is added to the active set (that is, $\mathsf{s}_k \in A_{k+1}$). Otherwise, at least one of the vertices not appearing in a convex representation of x_k is removed. This scheme converges linearly when applied to generic smooth strongly convex objectives (see, e.g., [62]).

In [48], a FW variant is proposed for minimum norm problems of the form

$$\min\{\|\mathsf{x}\|_* : f(\mathsf{x}) \le 0,\, \mathsf{x} \in K\} \tag{85}$$

with K a convex cone, f convex with L-Lipschitz gradient. In particular, the optimization domain is $C = \{\mathsf{x} \in \mathbb{R}^n : f(\mathsf{x}) \le 0\} \cap K$. The technique proposed in the article applies the standard FW method to the problems

$$\min\{f(\mathsf{x}) : \|\mathsf{x}\|_* \le \delta_k,\, \mathsf{x} \in K\} \,,$$

with $\{\delta_k\}$ an increasing sequence convergent to the optimal value $\bar{\delta}$ of the problem (85). Let $C(\delta) = \{\mathsf{x} \in \mathbb{R}^n : \|\mathsf{x}\|_* \le \delta\} \cap K$ for $\delta \ge 0$, and let

$$\mathrm{LM}(\mathsf{r}) \in \operatorname*{arg\,min}_{\mathsf{x}\in C(1)} \mathsf{r}^\intercal \mathsf{x} \,,$$

so that by homogeneity for every k the linear minimization oracle on $C(\delta_k)$ is given by

$$\mathrm{LMO}_{C(\delta_k)}(\mathsf{r}) = \delta_k \mathrm{LM}(\mathsf{r}) \,. \tag{86}$$

For every k, applying the FW method with suitable stopping conditions an approximate minimizer x_k of $f(\mathsf{x})$ over $C(\delta_k)$ is generated, with an associated lower bound on the objective, an affine function in y:

$$f_k(\mathsf{y}) := f(\mathsf{x}_k) + \nabla f(\mathsf{x}_k)^\intercal (\mathsf{y} - \mathsf{x}_k) \,. \tag{87}$$

Then the function

$$\ell_k(\delta) := \min_{\mathsf{y}\in C(\delta)} f_k(\mathsf{y}) = f_k(\delta \mathrm{LM}(\mathsf{g}_k)) \quad \text{with } \mathsf{g}_k = \nabla f(\mathsf{x}_k) \tag{88}$$

is decreasing and affine in δ and satisfies

$$\ell_k(\delta) = \min_{\mathsf{y} \in C(\delta)} f_k(\mathsf{y}) \leq F(\delta) := \min_{\mathsf{y} \in C(\delta)} f(\mathsf{y}) \,. \tag{89}$$

Therefore, for

$$\bar{\ell}_k(\delta) = \max_{i \in [1:k]} \ell_i(\delta) \leq F(\delta)$$

the quantity δ_{k+1} can be defined as $\min\{\delta \geq 0 : \bar{\ell}_k(\delta) \leq 0\}$, hence $F(\delta_{k+1}) \geq 0$. A complexity bound of $\mathcal{O}(\frac{1}{\varepsilon} \ln(\frac{1}{\varepsilon}))$ was given to achieve precision ε applying this method, with $\mathcal{O}(1/\varepsilon)$ iterations per subproblem and length of the sequence $\{\delta_k\}$ at most $\mathcal{O}(\ln(1/\varepsilon))$ (see [48, Theorem 2] for details).

7.3 Variants for optimization over the trace norm ball

The FW method has found many applications for optimization problems over the trace norm ball. In this case, as explained in Example 3.4, linear optimization can be obtained by computing the top left and right singular vectors of the matrix $-\nabla f(\mathsf{X}_k)$, an operation referred to as 1-SVD (see [3]) .

In the work [38], the FDFW is applied to the matrix completion problem (12), thus generating a sequence of matrices $\{\mathsf{X}_k\}$ with $\|\mathsf{X}_k\|_* \leq \delta$ for every k. The method can be implemented efficiently exploiting the fact that for X on the boundary of the nuclear norm ball, there is a simple expression for the face $\mathcal{F}(\mathsf{X})$. For $\mathsf{X} \in \mathbb{R}^{m \times n}$ with $\text{rank}(\mathsf{X}) = k$ let $\mathsf{UDV}^\intercal$ be the thin SVD of X, so that $\mathsf{D} \in \mathbb{R}^{k \times k}$ is the diagonal matrix of non zero singolar values for X, with corresponding left and right singular vectors in the columns of $\mathsf{U} \in \mathbb{R}^{m \times k}$ and $\mathsf{V} \in \mathbb{R}^{n \times k}$ respectively. If $\|\mathsf{X}\|_* = \delta$ then the minimal face of the domain containing X is the set

$$\mathcal{F}(\mathsf{X}) = \{\mathsf{X} \in \mathbb{R}^{m \times n} : \mathsf{X} = \mathsf{UMV}^\intercal \text{ for } \mathsf{M} = \mathsf{M}^\intercal \text{ psd with } \|\mathsf{M}\|_* = \delta\} \,. \tag{90}$$

It is not difficult to see that we have $\text{rank}(\mathsf{X}_k) \leq k + 1$ for every $k \in \mathbb{N}$, as well. Furthermore, the thin SVD of the current iterate X_k can be updated efficiently both after FW steps and after in face steps. The convergence rate of the FDFW in this setting is still $\mathcal{O}(1/k)$.

In the recent work [84], an unbounded variant of the FW method is applied to solve a generalized version of the trace norm ball optimization problem:

$$\min_{\mathsf{X} \in \mathbb{R}^{m \times n}} \{f(\mathsf{X}) : \|\mathsf{PXQ}\|_* \leq \delta\} \tag{91}$$

with P, Q singular matrices. The main idea of the method is to decompose the domain in the sum $S + T$ between the kernel T of the linear function $\varphi_{\mathsf{P},\mathsf{Q}}(\mathsf{X}) = \mathsf{PXQ}$ and a bounded set $S \subset T^\perp$. Then gradient descent steps in the unbounded component T are alternated to FW steps in the bounded component S. The authors apply this approach to the generalized LASSO as well, using the AFW for the bounded component.

In [3], a variant of the classic FW using k-SVD (computing the top k left and right singular vectors for the SVD) is introduced, and it is proved that it converges linearly for strongly convex objectives when the solution has rank at most k. In [71], the FW step is combined with a proximal gradient step for a quadratic problem on the product of the nuclear norm ball with the ℓ_1 ball. Approaches using an equivalent formulation on the spectrahedron introduced in [54] are analyzed in [32, 41].

8 Conclusions

While the concept of the FW method is quite easy to understand, its advantages, witnessed by a multitude of related work, may not be apparent to someone not closely familiar with the subject. Therefore we considered, in Section 3, several motivating applications, ranging from classic optimization to more recent machine learning problems. As in any line search-based method, the proper choice of stepsize is an important ingredient to achieve satisfactory performance. In Section 4, we review several options for stepsizes in first order methods, which are closely related both to the theoretical analysis as well as to practical implementation issues, guaranteeing fast convergence. This scope was investigated in more detail in Section 5 covering main results about the FW method and its most popular variants, including the $\mathcal{O}(1/k)$ convergence rate for convex objectives, affine invariance, the sparse approximation property, and support identification. The account is complemented by a report on recent progress in improving on the $\mathcal{O}(1/k)$ convergence rate in Section 6. Versatility and efficiency of this approach was also illustrated in the final Section 7 describing present recent FW variants fitting different optimization frameworks and computational environments, in particular block coordinate, distributed, accelerated, and trace norm optimization. For sure many other interesting and relevant aspects of FW and friends could not find their way into this review because of space and time limitations, but the authors hope to have convinced readers that FW merits a consideration even by non-experts in first-order optimization.

References

1. Ahipaşaoğlu, S.D., Sun, P., Todd, M.J.: Linear convergence of a modified Frank–Wolfe algorithm for computing minimum-volume enclosing ellipsoids. Optimisation Methods and Software **23**(1), 5–19 (2008)
2. Ahipaşaoğlu, S.D., Todd, M.J.: A modified Frank–Wolfe algorithm for computing minimum-area enclosing ellipsoidal cylinders: Theory and algorithms. Computational Geometry **46**(5), 494–519 (2013)
3. Allen-Zhu, Z., Hazan, E., Hu, W., Li, Y.: Linear convergence of a Frank-Wolfe type algorithm over trace-norm balls. Advances in Neural Information Processing Systems **2017**, 6192–6201 (2017)
4. Bach, F., et al.: Learning with submodular functions: A convex optimization perspective. Foundations and Trends® in Machine Learning **6**(2-3), 145–373 (2013)
5. Bashiri, M.A., Zhang, X.: Decomposition-invariant conditional gradient for general polytopes with line search. In: Advances in Neural Information Processing Systems, pp. 2690–2700 (2017)
6. Beck, A., Pauwels, E., Sabach, S.: The cyclic block conditional gradient method for convex optimization problems. SIAM Journal on Optimization **25**(4), 2024–2049 (2015)
7. Beck, A., Shtern, S.: Linearly convergent away-step conditional gradient for non-strongly convex functions. Mathematical Programming **164**(1-2), 1–27 (2017)
8. Berrada, L., Zisserman, A., Kumar, M.P.: Deep Frank-Wolfe for neural network optimization. In: International Conference on Learning Representations (2018)
9. Bertsekas, D.P.: Convex optimization algorithms. Athena Scientific (2015)
10. Bomze, I.M.: Evolution towards the maximum clique. Journal of Global Optimization **10**(2), 143–164 (1997)
11. Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M.: The maximum clique problem. In: Handbook of Combinatorial Optimization, pp. 1–74. Springer (1999)
12. Bomze, I.M., de Klerk, E.: Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. Journal of Global Optimization **24**(2), 163–185 (2002)

13. Bomze, I.M., Rinaldi, F., Rota Bulò, S.: First-order methods for the impatient: Support identification in finite time with convergent Frank-Wolfe variants. SIAM Journal on Optimization **29**(3), 2211–2226 (2019)
14. Bomze, I.M., Rinaldi, F., Zeffiro, D.: Active set complexity of the away-step Frank–Wolfe algorithm. SIAM Journal on Optimization **30**(3), 2470–2500 (2020)
15. Boyd, S., Boyd, S.P., Vandenberghe, L.: Convex optimization. Cambridge university press (2004)
16. Braun, G., Pokutta, S., Tu, D., Wright, S.: Blended conditonal gradients. In: International Conference on Machine Learning, pp. 735–743. PMLR (2019)
17. Braun, G., Pokutta, S., Zink, D.: Lazifying conditional gradient algorithms. In: ICML, pp. 566–575 (2017)
18. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. Foundations of Computational mathematics **9**(6), 717–772 (2009)
19. Canon, M.D., Cullum, C.D.: A tight upper bound on the rate of convergence of Frank-Wolfe algorithm. SIAM Journal on Control **6**(4), 509–516 (1968)
20. Carderera, A., Pokutta, S.: Second-order conditional gradient sliding. arXiv preprint arXiv:2002.08907 (2020)
21. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE symposium on security and privacy (sp), pp. 39–57. IEEE (2017)
22. Chakrabarty, D., Jain, P., Kothari, P.: Provable submodular minimization using Wolfe's algorithm. Advances in Neural Information Processing Systems **27**, 802–809 (2014)
23. Chen, J., Zhou, D., Yi, J., Gu, Q.: A Frank-Wolfe framework for efficient and effective adversarial attacks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3486–3494 (2020)
24. Chen, P.Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.J.: ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 15–26 (2017)
25. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Review **43**(1), 129–159 (2001)
26. Clarkson, K.L.: Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. ACM Transactions on Algorithms **6**(4), 1–30 (2010)
27. Combettes, C., Pokutta, S.: Boosting frank-wolfe by chasing gradients. In: International Conference on Machine Learning, pp. 2111–2121. PMLR (2020)
28. Combettes, C.W., Pokutta, S.: Complexity of linear minimization and projection on some sets. arXiv preprint arXiv:2101.10040 (2021)
29. Cristofari, A., De Santis, M., Lucidi, S., Rinaldi, F.: An active-set algorithmic framework for non-convex optimization problems over the simplex. Computational Optimization and Applications **77**, 57–89 (2020)
30. Demyanov, V.F., Rubinov, A.M.: Approximate methods in optimization problems. American Elsevier (1970)
31. Devolder, O., Glineur, F., Nesterov, Y.: First-order methods of smooth convex optimization with inexact oracle. Mathematical Programming **146**(1), 37–75 (2014)
32. Ding, L., Fei, Y., Xu, Q., Yang, C.: Spectral Frank-Wolfe algorithm: Strict complementarity and linear convergence. In: International Conference on Machine Learning, pp. 2535–2544. PMLR (2020)
33. Dunn, J.C.: Rates of convergence for conditional gradient algorithms near singular and nonsingular extremals. SIAM Journal on Control and Optimization **17**(2), 187–211 (1979)
34. Dunn, J.C., Harshbarger, S.: Conditional gradient algorithms with open loop step size rules. Journal of Mathematical Analysis and Applications **62**(2), 432–444 (1978)
35. Ferreira, O., Sosa, W.: On the Frank–Wolfe algorithm for non-compact constrained optimization problems. Optimization pp. 1–15 (2021)
36. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Research Logistics Quarterly **3**(1-2), 95–110 (1956)
37. Freund, R.M., Grigas, P.: New analysis and results for the Frank–Wolfe method. Mathematical Programming **155**(1-2), 199–230 (2016)
38. Freund, R.M., Grigas, P., Mazumder, R.: An extended Frank–Wolfe method with in-face directions, and its application to low-rank matrix completion. SIAM Journal on Optimization **27**(1), 319–346 (2017)
39. Fujishige, S.: Lexicographically optimal base of a polymatroid with respect to a weight vector. Mathematics of Operations Research **5**(2), 186–196 (1980)

40. Fukushima, M.: A modified Frank-Wolfe algorithm for solving the traffic assignment problem. Transportation Research Part B: Methodological **18**(2), 169–177 (1984)
41. Garber, D.: Linear convergence of Frank-Wolfe for rank-one matrix recovery without strong convexity. arXiv preprint arXiv:1912.01467 (2019)
42. Garber, D.: Revisiting frank-wolfe for polytopes: Strict complementarity and sparsity. Advances in Neural Information Processing Systems **33** (2020)
43. Garber, D., Hazan, E.: Faster rates for the Frank-Wolfe method over strongly-convex sets. In: ICML, vol. 15, pp. 541–549 (2015)
44. Garber, D., Hazan, E.: A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization. SIAM Journal on Optimization **26**(3), 1493–1528 (2016)
45. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
46. Guelat, J., Marcotte, P.: Some comments on Wolfe's away step. Mathematical Programming **35**(1), 110–119 (1986)
47. Gutman, D.H., Pena, J.F.: The condition number of a function relative to a set. Mathematical Programming pp. 1–40 (2020)
48. Harchaoui, Z., Juditsky, A., Nemirovski, A.: Conditional gradient algorithms for norm-regularized smooth convex optimization. Mathematical Programming **152**(1), 75–112 (2015)
49. Hogan, W.W.: Convergence results for some extensions of the Frank-Wolfe method. Tech. rep., CALIFORNIA UNIV LOS ANGELES WESTERN MANAGEMENT SCIENCE INST (1971)
50. Holloway, C.A.: An extension of the Frank and Wolfe method of feasible directions. Mathematical Programming **6**(1), 14–27 (1974)
51. Hungerford, J.T., Rinaldi, F.: A general regularized continuous formulation for the maximum clique problem. Mathematics of Operations Research **44**(4), 1161–1173 (2019)
52. Jaggi, M.: Sparse convex optimization methods for machine learning. Ph.D. thesis, ETH Zurich (2011)
53. Jaggi, M.: Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In: ICML (1), pp. 427–435 (2013)
54. Jaggi, M., Sulovský, M.: A simple algorithm for nuclear norm regularized problems. In: ICML, pp. 471–478 (2010)
55. Joulin, A., Tang, K., Fei-Fei, L.: Efficient image and video co-localization with Frank-Wolfe algorithm. In: European Conference on Computer Vision, pp. 253–268. Springer (2014)
56. Kazemi, E., Kerdreux, T., Wang, L.: Generating structured adversarial attacks using Frank-Wolfe method. arXiv preprint arXiv:2102.07360 (2021)
57. Kerdreux, T., d'Aspremont, A., Pokutta, S.: Projection-free optimization on uniformly convex sets. In: International Conference on Artificial Intelligence and Statistics, pp. 19–27. PMLR (2021)
58. Kerdreux, T., Liu, L., Lacoste-Julien, S., Scieur, D.: Affine invariant analysis of Frank-Wolfe on strongly convex sets. arXiv preprint arXiv:2011.03351 (2020)
59. Konnov, I.: Simplified versions of the conditional gradient method. Optimization **67**(12), 2275–2290 (2018)
60. Kumar, P., Mitchell, J.S., Yıldırım, E.A.: Approximate minimum enclosing balls in high dimensions using core-sets. Journal of Experimental Algorithmics **8**, 1–1 (2003)
61. Lacoste-Julien, S.: Convergence rate of Frank-Wolfe for non-convex objectives. arXiv preprint arXiv:1607.00345 (2016)
62. Lacoste-Julien, S., Jaggi, M.: On the global linear convergence of Frank-Wolfe optimization variants. In: Advances in Neural Information Processing Systems, pp. 496–504 (2015)
63. Lacoste-Julien, S., Jaggi, M., Schmidt, M., Pletscher, P.: Block-coordinate Frank-Wolfe optimization for structural SVMs. In: S. Dasgupta, D. McAllester (eds.) Proceedings of the 30th International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 28, pp. 53–61. PMLR, Atlanta, Georgia, USA (2013)
64. Lan, G.: First-order and Stochastic Optimization Methods for Machine Learning. Springer (2020)
65. Lan, G., Zhou, Y.: Conditional gradient sliding for convex optimization. SIAM Journal on Optimization **26**(2), 1379–1409 (2016)

66. LeBlanc, L.J., Morlok, E.K., Pierskalla, W.P.: An efficient approach to solving the road network equilibrium traffic assignment problem. Transportation Research **9**(5), 309–318 (1975)
67. Levitin, E.S., Polyak, B.T.: Constrained minimization methods. USSR Computational Mathematics and Mathematical Physics **6**(5), 1–50 (1966)
68. Locatello, F., Khanna, R., Tschannen, M., Jaggi, M.: A unified optimization view on generalized matching pursuit and frank-wolfe. In: Artificial Intelligence and Statistics, pp. 860–868. PMLR (2017)
69. Luce, R.D., Perry, A.D.: A method of matrix analysis of group structure. Psychometrika **14**(2), 95–116 (1949)
70. Mitchell, B., Demyanov, V.F., Malozemov, V.: Finding the point of a polyhedron closest to the origin. SIAM Journal on Control **12**(1), 19–26 (1974)
71. Mu, C., Zhang, Y., Wright, J., Goldfarb, D.: Scalable robust matrix recovery: Frank–Wolfe meets proximal methods. SIAM Journal on Scientific Computing **38**(5), A3291–A3317 (2016)
72. Osokin, A., Alayrac, J.B., Lukasewitz, I., Dokania, P., Lacoste-Julien, S.: Minding the gaps for block Frank-Wolfe optimization of structured svms. In: International Conference on Machine Learning, pp. 593–602. PMLR (2016)
73. Peña, J., Rodriguez, D.: Polytope conditioning and linear convergence of the Frank-Wolfe algorithm. Mathematics of Operartions Research **44**(1), 1–18 (2018)
74. Pedregosa, F., Negiar, G., Askari, A., Jaggi, M.: Linearly convergent Frank-Wolfe with backtracking line-search. In: International Conference on Artificial Intelligence and Statistics, pp. 1–10. PMLR (2020)
75. Rademacher, L., Shu, C.: The smoothed complexity of Frank-Wolfe methods via conditioning of random matrices and polytopes. arXiv preprint arXiv:2009.12685 (2020)
76. Rinaldi, F., Zeffiro, D.: Avoiding bad steps in Frank Wolfe variants. arXiv preprint arXiv:2012.12737 (2020)
77. Rinaldi, F., Zeffiro, D.: A unifying framework for the analysis of projection-free first-order methods under a sufficient slope condition. arXiv preprint arXiv:2008.09781 (2020)
78. Sahu, A.K., Kar, S.: Decentralized zeroth-order constrained stochastic optimization algorithms: Frank–Wolfe and variants with applications to black-box adversarial attacks. Proceedings of the IEEE **108**(11), 1890–1905 (2020)
79. Shah, N., Kolmogorov, V., Lampert, C.H.: A multi-plane block-coordinate Frank-Wolfe algorithm for training structural svms with a costly max-oracle. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2737–2745 (2015)
80. Sun, Y.: Safe screening for the generalized conditional gradient method. image **1**, 2 (2020)
81. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) **58**(1), 267–288 (1996)
82. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (2013)
83. Von Hohenbalken, B.: Simplicial decomposition in nonlinear programming algorithms. Mathematical Programming **13**(1), 49–68 (1977)
84. Wang, H., Lu, H., Mazumder, R.: Frank-Wolfe methods with an unbounded feasible region and applications to structured learning. arXiv preprint arXiv:2012.15361 (2020)
85. Wang, Y.X., Sadhanala, V., Dai, W., Neiswanger, W., Sra, S., Xing, E.: Parallel and distributed block-coordinate Frank-Wolfe algorithms. In: International Conference on Machine Learning, pp. 1548–1557. PMLR (2016)
86. Wardrop, J.G.: Road paper. some theoretical aspects of road traffic research. Proceedings of the Institution of Civil Engineers **1**(3), 325–362 (1952)
87. Weintraub, A., Ortiz, C., González, J.: Accelerating convergence of the Frank-Wolfe algorithm. Transportation Research Part B: Methodological **19**(2), 113–122 (1985)
88. Wolfe, P.: Convergence theory in nonlinear programming. In: J. Abadie (ed.) Integer and nonlinear programming, pp. 1–36. North Holland (1970)
89. Wolfe, P.: Finding the nearest point in a polytope. Mathematical Programming **11**(1), 128–149 (1976)
90. Wu, Q., Hao, J.K.: A review on algorithms for maximum clique problems. European Journal of Operational Research **242**(3), 693–709 (2015)
91. Xu, Y., Yang, T.: Frank-Wolfe method is automatically adaptive to error bound condition. arXiv preprint arXiv:1810.04765 (2018). URL https://arxiv.org/abs/1810.04765
92. Yıldırım, E.A.: Two algorithms for the minimum enclosing ball problem. SIAM Journal on Optimization **19**(3), 1368–1391 (2008)