Efficient Quadratic Corrections for Frank-Wolfe Algorithms

Sebastian Pokutta

joint work with: Jannis Halbey, Seta Rakotomandimby,
Mathieu Besançon, and Sébastien Designolle

Technische Universität Berlin
and
Zuse Institute Berlin

pokutta@math.tu-berlin.de
@spokutta

SIAM Conference on Optimization (OP26)

June 5th, 2026 · Edinburgh, Scotland

ZUSE INSTITUTE BERLIN

Berlin Mathematics Research Center MATH+

What is this talk about?

Introduction

How can we make active-set Frank–Wolfe methods much faster on quadratic subproblems?

What is this talk about?

Introduction

How can we make active-set Frank-Wolfe methods much faster on quadratic subproblems?

Why? Corrective FW steps are powerful, but exact fully-corrective updates are often too expensive.

What is this talk about?

Introduction

How can we make active-set Frank-Wolfe methods much faster on quadratic subproblems?

Why? Corrective FW steps are powerful, but exact fully-corrective updates are often too expensive.

Today.

- A generic Corrective Frank-Wolfe (CFW) framework
- Two efficient quadratic corrections: QC-LP and QC-MNP
- Theory + computational experiments (sparse regression, entanglement, splitting)

(Hyperlinked) References are not exhaustive; check references contained therein.

Conditional Gradients
a.k.a. the Frank-Wolfe algorithm

—The Basics—

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

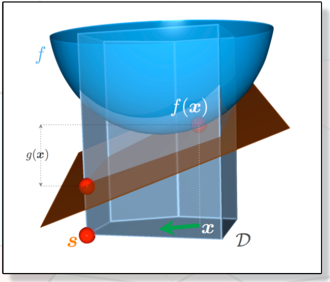
Given a smooth and convex function f and a polytope P, solve optimization problem:

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

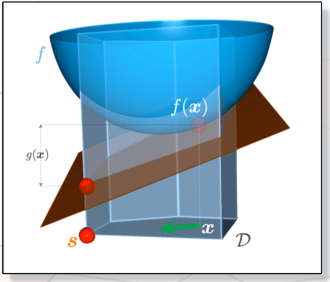
$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



Source: [Jaggi, 2013]

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm



Source: [Jaggi, 2013]

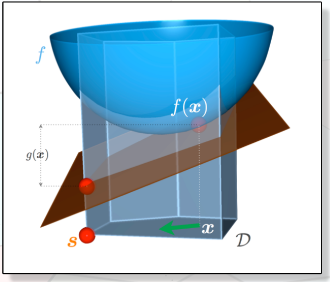
Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$

1. Very versatile model
2. Can use various types of information about both f and P
3. Works very well in (continuous) real-world applications
4. At the core of many (all?) learning algorithms (albeit mostly non-convex case)

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm



Source: [Jaggi, 2013]

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$

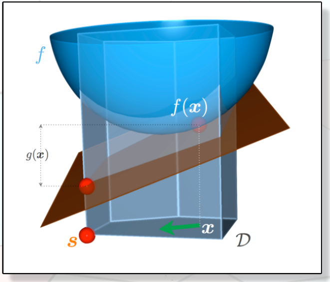
Our setup.

1. Access to P. Linear Minimization Oracle (LMO): Given linear objective c return

$$x \leftarrow \arg\min_{v \in P} c^T v.$$

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm



Source: [Jaggi, 2013]

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$

Our setup.

1. Access to P. Linear Minimization Oracle (LMO): Given linear objective c return

$$x \leftarrow \arg\min_{v \in P} c^T v.$$

2. Access to f. First-Order Oracle (FO): Given x return

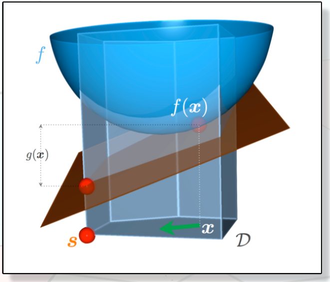
$$\nabla f(x) \quad \text{and} \quad f(x).$$

The basic problem

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Given a smooth and convex function f and a polytope P, solve optimization problem:

$$\min_{x \in P} f(x) \qquad \text{(baseProblem)}$$



Source: [Jaggi, 2013]

Our setup.

1. Access to P. Linear Minimization Oracle (LMO): Given linear objective c return

$$x \leftarrow \arg\min_{v \in P} c^T v.$$

2. Access to f. First-Order Oracle (FO): Given x return

$$\nabla f(x) \quad \text{and} \quad f(x).$$

⇒ *Complexity of convex optimization relative to LO/FO oracle*

The Frank-Wolfe Algorithm

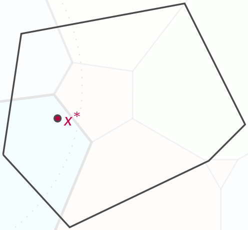
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

for $t = 0$ **to** $T - 1$ **do**
    $v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
    $x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

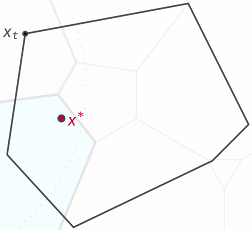
The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$
Output: Iterates $x_1, x_2, \ldots \in P$

for $t = 0$ **to** $T - 1$ **do**
    $v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
    $x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

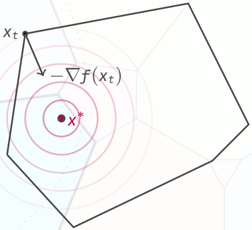
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

for $t = 0$ **to** $T-1$ **do**
    $v_t \leftarrow \mathrm{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
    $x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

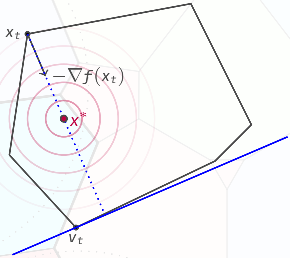
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

for $t = 0$ **to** $T - 1$ **do**
- $v_t \leftarrow \mathrm{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
- $x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

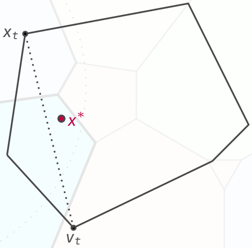
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$,
and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

for $t = 0$ **to** $T - 1$ **do**
$\quad v_t \leftarrow \text{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
$\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

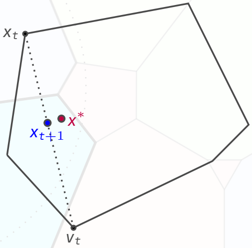
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**
2: $\quad v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
3: $\quad x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

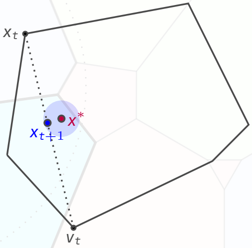
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

for $t = 0$ **to** $T - 1$ **do**
  $v_t \leftarrow \mathrm{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
  $x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

The Frank-Wolfe Algorithm

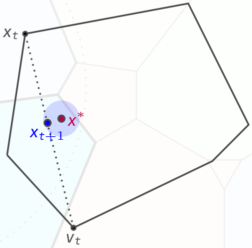
Conditional Gradients a.k.a. the Frank-Wolfe algorithm

Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$,
and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

for $t = 0$ **to** $T - 1$ **do**
  $v_t \leftarrow \text{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
  $x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$



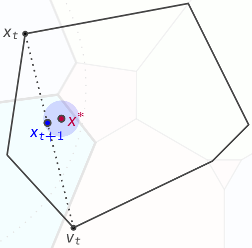
[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

Advantages:

- Extremely simple and robust: no complicated data structures to maintain
- Easy to implement: requires only the two oracles
- Projection-free: feasibility convex combination and LO oracle.
- Sparsity: optimal solution is a convex combination of (usually) vertices.

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm



Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$
Output: Iterates $x_1, x_2, \ldots \in P$

1: **for** $t = 0$ **to** $T - 1$ **do**
2: $v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
3: $x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$

[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

Advantages:

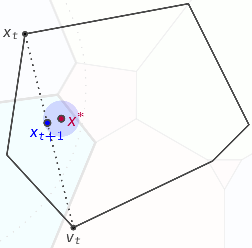
- Extremely simple and robust: no complicated data structures to maintain
- Easy to implement: requires only the two oracles
- Projection-free: feasibility convex combination and LO oracle.
- Sparsity: optimal solution is a convex combination of (usually) vertices.

Disadvantages:

- Suboptimal convergence rate of $O(1/T)$

The Frank-Wolfe Algorithm

Conditional Gradients a.k.a. the Frank-Wolfe algorithm



Algorithm Frank-Wolfe Algorithm (FW)

Input: Point $x_0 \in P$, feasible region $P \subseteq \mathbb{R}^n$, and step sizes $0 < \gamma_t \leq 1$

Output: Iterates $x_1, x_2, \ldots \in P$

for $t = 0$ **to** $T - 1$ **do**
  $v_t \leftarrow \operatorname{argmin}_{v \in P} \langle \nabla f(x_t), v \rangle$
  $x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$

[Frank and Wolfe, 1956, Levitin and Polyak, 1966]

Advantages:

- Extremely simple and robust: no complicated data structures to maintain
- Easy to implement: requires only the two oracles
- Projection-free: feasibility convex combination and LO oracle.
- Sparsity: optimal solution is a convex combination of (usually) vertices.

Disadvantages:

- Suboptimal convergence rate of $O(1/T)$

$\Rightarrow$ Despite (theoretically) suboptimal rate heavily used in applications due to simplicity.

Significant progress over the recent years (incomplete list)

Conditional Gradients a.k.a. the Frank-Wolfe algorithm

1. Strongly convex case [Garber and Hazan, 2013, Lacoste-Julien and Jaggi, 2015, Lan and Zhou, 2016, Garber and Meshi, 2016]
2. Non-convex case [Lacoste-Julien, 2016]
3. Online case [Hazan and Kale, 2012]
4. Stochastic variants and adaptive gradients [Hazan and Luo, 2016, Reddi et al., 2016, Combettes et al., 2020]
5. Sharp functions and sharp regions [Kerdreux et al., 2019, 2021, 2025]
6. Acceleration [Diakonikolas et al., 2020, Bach, 2020, Carderera et al., 2021]
7. Specialized variants [Freund et al., 2017, Braun et al., 2017b, 2019b,a]

Conditional Gradients very competitive: simple, robust, real-world performance.

For more background etc see our survey! [Braun et al., 2025]

A generic corrective FW algorithm

[Halbey et al., 2025]

Algorithm Corrective Frank-Wolfe (CFW)

Input: f, $x_0 \in V(\mathcal{X})$, corrective step CS

$S_0 \leftarrow \{x_0\}$
for $t = 0$ **to** $T - 1$ **do**
    $a_t \leftarrow \text{argmax}_{v \in S_t} \langle \nabla f(x_t), v \rangle$
    $s_t \leftarrow \text{argmin}_{v \in S_t} \langle \nabla f(x_t), v \rangle$
    $v_t \leftarrow \text{argmin}_{v \in V(\mathcal{X})} \langle \nabla f(x_t), v \rangle$
    if $\langle \nabla f(x_t), a_t - s_t \rangle \geq \langle \nabla f(x_t), x_t - v_t \rangle$ **then**
        $(x_{t+1}, S_{t+1}) \leftarrow \text{CS}(S_t, x_t, a_t, s_t)$
    else
        $\gamma_t \leftarrow \text{argmin}_{\gamma \in [0,1]} f(x_t + \gamma(v_t - x_t))$
        $x_{t+1} \leftarrow x_t + \gamma_t (v_t - x_t)$
        $S_{t+1} \leftarrow S_t \cup \{v_t\}$

Admissibility of Corrective Step:

- Descent step:

$$f(x) - f(x') \geq \frac{\langle \nabla f(x), a - s \rangle^2}{2LD^2}$$

- or Drop step: $f(x') \leq f(x)$, $S' \subsetneq S$

Step Types That Satisfy It

A generic corrective FW algorithm

Three common step types that satisfy the admissibility conditions:

- Blended Conditional Gradients (BCG): Simplex-Gradient Step [Braun et al., 2019a]
- Blended Pairwise Conditional Gradients (BPCG): Local Pairwise Step [Tsuji et al., 2022]
- Fully-Corrective Frank-Wolfe (FCFW): Fully-Corrective Step [Holloway, 1974]

Step Types That Satisfy It

A generic corrective FW algorithm

Three common step types that satisfy the admissibility conditions:

- **Blended Conditional Gradients (BCG)**: Simplex-Gradient Step [Braun et al., 2019a]
- **Blended Pairwise Conditional Gradients (BPCG)**: Local Pairwise Step [Tsuji et al., 2022]
- **Fully-Corrective Frank-Wolfe (FCFW)**: Fully-Corrective Step [Holloway, 1974]

Examples. (two extremes)

Algorithm Local Pairwise Step (LPS)

Input: S, x, a, s

```
γ* ← argmin_{γ∈[0,λ_a(x)]} f(x + γ(s − a))
x' ← x + γ*(s − a)
if γ* = λ_a(x) then
    S' ← S \ {a}
else
    S' ← S
```

$\gamma^\star \leftarrow \mathrm{argmin}_{\gamma \in [0, \lambda_a(x)]} f(x + \gamma(s - a))$

$x' \leftarrow x + \gamma^\star (s - a)$

if $\gamma^\star = \lambda_a(x)$ **then**

  $S' \leftarrow S \setminus \{a\}$

else

  $S' \leftarrow S$

Algorithm Fully-Corrective Step (FCS)

Input: S

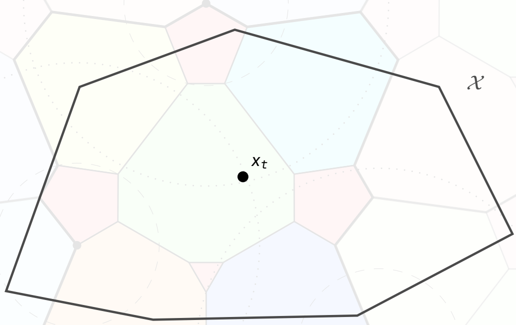
$x' \leftarrow \mathrm{argmin}_{y \in \mathrm{conv}(S)} f(y)$

$S' \leftarrow \{v \in S \mid \lambda_v(x') > 0\}$

A Generic Corrective FW Algorithm: Illustration

A generic corrective FW algorithm

[Halbey et al., 2025]

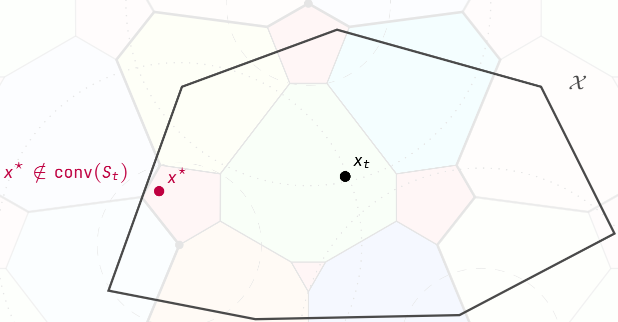


CFW performs local corrective optimization inside $\text{conv}(S_t)$ until progress stalls; then a global FW step adds a new atom and updates the active set S_t.

A Generic Corrective FW Algorithm: Illustration

A generic corrective FW algorithm

[Halbey et al., 2025]

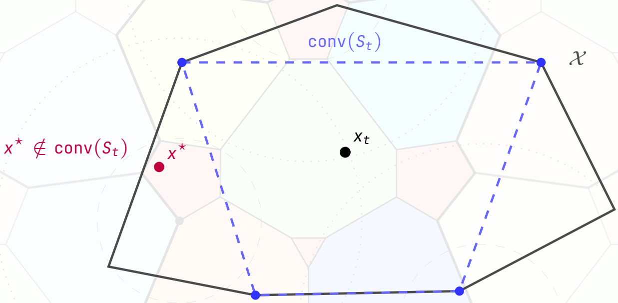


CFW performs local corrective optimization inside $\mathrm{conv}(S_t)$ until progress stalls; then a global FW step adds a new atom and updates the active set S_t.

A Generic Corrective FW Algorithm: Illustration

A generic corrective FW algorithm

[Halbey et al., 2025]

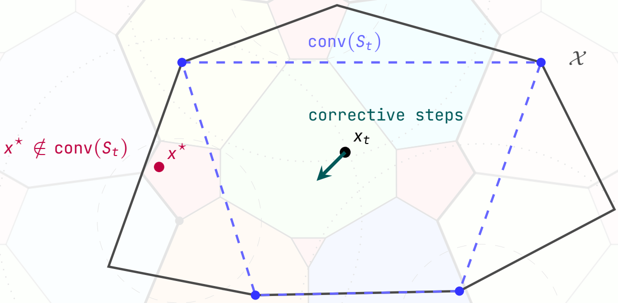


CFW performs local corrective optimization inside conv(S_t) until progress stalls; then a global FW step adds a new atom and updates the active set S_t.

A Generic Corrective FW Algorithm: Illustration

A generic corrective FW algorithm

[Halbey et al., 2025]

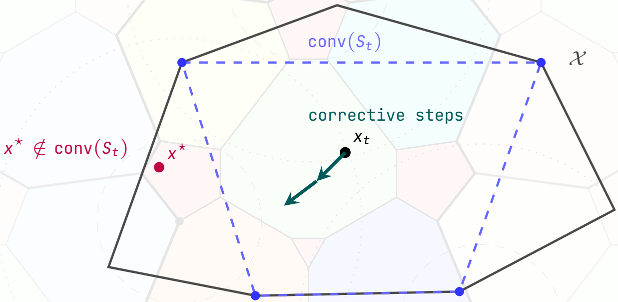


CFW performs local corrective optimization inside conv(S_t) until progress stalls; then a global FW step adds a new atom and updates the active set S_t.

A Generic Corrective FW Algorithm: Illustration

A generic corrective FW algorithm

[Halbey et al., 2025]

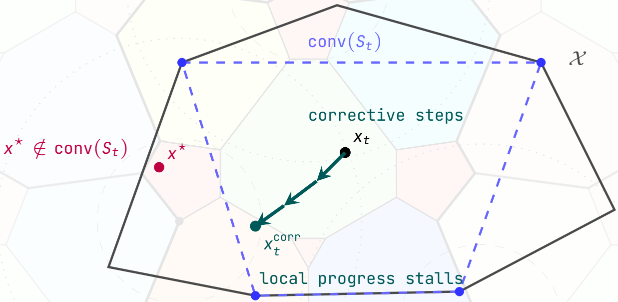


CFW performs local corrective optimization inside $\mathrm{conv}(S_t)$ until progress stalls; then a global FW step adds a new atom and updates the active set S_t.

A Generic Corrective FW Algorithm: Illustration

A generic corrective FW algorithm

[Halbey et al., 2025]

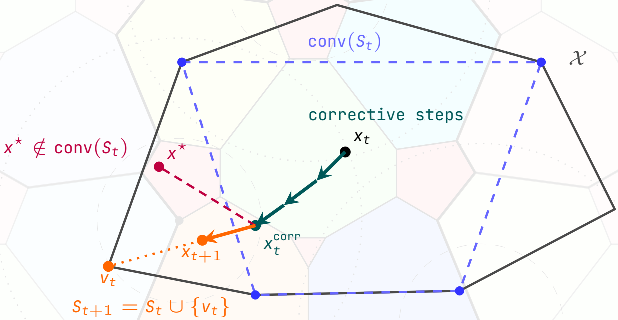


CFW performs local corrective optimization inside $\mathrm{conv}(S_t)$ until progress stalls; then a global FW step adds a new atom and updates the active set S_t.

A Generic Corrective FW Algorithm: Illustration

A generic corrective FW algorithm

[Halbey et al., 2025]

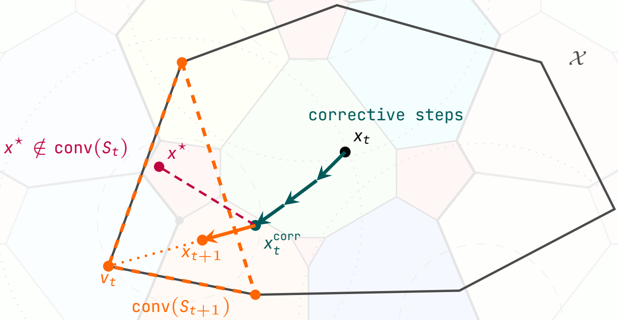


CFW performs local corrective optimization inside $\mathrm{conv}(S_t)$ until progress stalls; then a global FW step adds a new atom and updates the active set S_t.

A Generic Corrective FW Algorithm: Illustration

A generic corrective FW algorithm

[Halbey et al., 2025]



CFW performs local corrective optimization inside $\mathrm{conv}(S_t)$ until progress stalls; then a global FW step adds a new atom and updates the active set S_t.

Convergence of Corrective Frank-Wolfe

A generic corrective FW algorithm

[Halbey et al., 2025]

Theorem (CFW convergence)

Let f be convex and L-smooth over a compact convex set $\mathcal{X}$ with diameter D. For iterates $\{x_t\}$ generated by CFW, we have

$$f(x_T) - f^\star \leq \frac{4LD^2}{T}.$$

If additionally f is $(c, \frac{1}{2})$-sharp and $\mathcal{X}$ is a polytope with pyramidal width δ, then

$$f(x_T) - f^\star \leq (f(x_0) - f^\star) \exp(-c_{f,\mathcal{X}} T), \quad c_{f,\mathcal{X}} = \min\left\{\frac{1}{4}, \frac{\delta^2}{16Lc^2D^2}\right\}.$$

More generally, for (c, θ)-sharp objectives with $\theta < \frac{1}{2}$:

$$f(x_T) - f^\star = \mathcal{O}\left(T^{-\frac{1}{1-2\theta}}\right).$$

Note. CFW can also be lazified to significantly reduce the number of LMO calls; convergence rates are preserved.

[Braun et al., 2017b, 2019b]

Quadratic Corrections

Motivation: the Quadratic Special Case

Quadratic Corrections

All-important special case:

$$f(x) = \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle + c, \quad A \succeq 0.$$

Motivation: the Quadratic Special Case

Quadratic Corrections

All-important special case:

$$f(x) = \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle + c, \quad A \succeq 0.$$

For a fixed active set S_t with atoms $v \in S_t$, write

$$x = \sum_{v \in S_t} \lambda_v v, \qquad \sum_{v \in S_t} \lambda_v = 1, \qquad \lambda_v \geq 0.$$

Then

$$\nabla f(x) = Ax + b = \sum_{v \in S_t} \lambda_v (Av + b) = \sum_{v \in S_t} \lambda_v \nabla f(v).$$

Motivation: the Quadratic Special Case

Quadratic Corrections

All-important special case:

$$f(x) = \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle + c, \quad A \succeq 0.$$

For a fixed active set S_t with atoms $v \in S_t$, write

$$x = \sum_{v \in S_t} \lambda_v v, \qquad \sum_{v \in S_t} \lambda_v = 1, \qquad \lambda_v \geq 0.$$

Then

$$\nabla f(x) = Ax + b = \sum_{v \in S_t} \lambda_v (Av + b) = \sum_{v \in S_t} \lambda_v \nabla f(v).$$

Key obsevation: for quadratic objectives and fixed S_t

1. For $x \in \operatorname{conv}(S_t)$, we have $\nabla f(x)$ is a convex combination of gradients.
2. For $x \in \operatorname{aff}(S_t)$, we have $\nabla f(x)$ is an affine combination of gradients.

Optimality System and Two Paths

Quadratic Corrections

Let V collect active atoms of S_t as columns and $Av + b = \nabla f(v)$ for $v \in S_t$.

Optimality System and Two Paths

Quadratic Corrections

Let V collect active atoms of S_t as columns and $Av + b = \nabla f(v)$ for $v \in S_t$.

First-order optimality over $\mathrm{aff}(S_t)$**.** Linear system of the form.

$$\langle AV\lambda + b, v - w\rangle = 0 \quad \forall v, w \in S_t, v \neq w,$$
$$\mathbf{1}^\top \lambda = 1.$$

Optimality System and Two Paths

Quadratic Corrections

Let V collect active atoms of S_t as columns and $Av + b = \nabla f(v)$ for $v \in S_t$.

First-order optimality over $\mathrm{aff}(S_t)$. Linear system of the form.

$$\langle AV\lambda + b, v - w \rangle = 0 \quad \forall v, w \in S_t, v \neq w,$$
$$\mathbf{1}^\top \lambda = 1.$$

Lemma (Affine minimizer existence)

The system is feasible iff $b \perp \left(\mathrm{span}(S_t) \cap \ker(A)\right)$. *In particular, it is feasible if* $A \succ 0$.

Optimality System and Two Paths

Quadratic Corrections

Let V collect active atoms of S_t as columns and $Av + b = \nabla f(v)$ for $v \in S_t$.

First-order optimality over $\mathrm{aff}(S_t)$**.** Linear system of the form.

$$\langle AV\lambda + b, v - w \rangle = 0 \quad \forall v, w \in S_t, v \neq w,$$
$$\mathbf{1}^\top \lambda = 1.$$

Lemma (Affine minimizer existence)

The system is feasible iff $b \perp \big(\mathrm{span}(S_t) \cap \ker(A)\big)$*. In particular, it is feasible if* $A \succ 0$*.*

Need to ensure. Nonnegativity of multipliers, i.e., $\lambda \geq 0$. Two paths:

1. QC-LP enforce $\lambda \geq 0$ directly, which leads to an LP.
2. QC-MNP allow negatives in affine solve, then “truncate” to simplex.

Optimality System and Two Paths

Quadratic Corrections

Let V collect active atoms of S_t as columns and $Av + b = \nabla f(v)$ for $v \in S_t$.

First-order optimality over aff(S_t). Linear system of the form.

$$\langle AV\lambda + b, v - w\rangle = 0 \quad \forall v, w \in S_t, v \neq w,$$
$$\mathbf{1}^\top \lambda = 1.$$

Lemma (Affine minimizer existence)

The system is feasible iff $b \perp \left(\operatorname{span}(S_t) \cap \ker(A)\right)$. *In particular, it is feasible if* $A \succ 0$.

Need to ensure. Nonnegativity of multipliers, i.e., $\lambda \geq 0$. Two paths:

1. QC-LP enforce $\lambda \geq 0$ directly, which leads to an LP.
2. QC-MNP allow negatives in affine solve, then “truncate” to simplex.

Note. In practice, only attempt QC-LP or QC-MNP once in a while, e.g., exponentially with period T.

Quadratic Correction via LP (QC-LP)

Linear system is a feasibility problem of the form:

$$
\begin{aligned}
&\text{find } \lambda \text{ s.t.} \\
&\langle AV\lambda + b, v - w \rangle = 0 \quad \forall v, w \in S_t, v \neq w, \\
&\mathbf{1}^\top \lambda = 1, \\
&\lambda \geq 0.
\end{aligned}
$$

Quadratic Correction via LP (QC-LP)

Quadratic Corrections

Linear system is a feasibility problem of the form:

$$\text{find } \lambda \text{ s.t.}$$
$$\langle AV\lambda + b, v - w\rangle = 0 \quad \forall v, w \in S_t, v \neq w,$$
$$\mathbf{1}^\top \lambda = 1,$$
$$\lambda \geq 0.$$

Algorithm Quadratic Correction LP (QC-LP)

Input: S_t, x_t, a_t, s_t

1: Solve linear system + simplex constraints for λ

2: **if** feasible **then**

3: $\quad x_{t+1} \leftarrow \sum_{v \in S_t} \lambda_v v, \quad S_{t+1} \leftarrow S_t$

4: **else**

5: $\quad (x_{t+1}, S_{t+1}) \leftarrow \mathrm{LPS}(S_t, x_t, a_t, s_t)$

Quadratic Correction via MNP (QC-MNP)

Try to solve the unconstrained affine problem for $\tilde{\lambda}$ first, then truncate or fallback to LPS.

(basically: Minimum Norm Point problem).

Quadratic Correction via MNP (QC-MNP)

Try to solve the unconstrained affine problem for $\tilde{\lambda}$ first, then truncate or fallback to LPS.

(basically: Minimum Norm Point problem).

Algorithm Quadratic Correction MNP (QC-MNP)

Input: S_t, x_t, a_t, s_t

Solve affine optimality system for $\tilde{\lambda}$ (without $\tilde{\lambda} \geq 0$)

if infeasible **then**

   $(x_{t+1}, S_{t+1}) \leftarrow \mathrm{LPS}(S_t, x_t, a_t, s_t)$

else if $\tilde{\lambda} \geq 0$ **then**

   $x_{t+1} \leftarrow \sum_{v \in S_t} \tilde{\lambda}_v v, \quad S_{t+1} \leftarrow S_t$

else

   $\tau \leftarrow \min\left\{ \frac{\lambda_v(x_t)}{\lambda_v(x_t) - \tilde{\lambda}_v} : \tilde{\lambda}_v < \lambda_v(x_t) \right\}$

   $\lambda' \leftarrow \tau \tilde{\lambda} + (1-\tau)\lambda(x_t)$

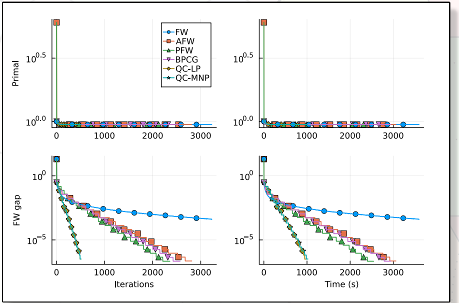
   $x_{t+1} \leftarrow \sum_{v \in S_t} \lambda'_v v, \quad S_{t+1} \leftarrow \{v \in S_t : \lambda'_v > 0\}$

Computational Experiments

Sparse Regression

Computational Experiments

[Halbey et al., 2025]



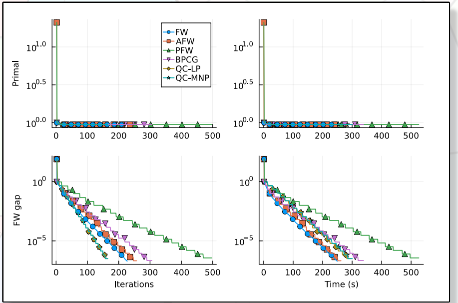


Figure: Sparse regression over the K-sparse polytope for $K \in \{5, 20\}$.

Entanglement Detection

Computational Experiments

[Halbey et al., 2025]

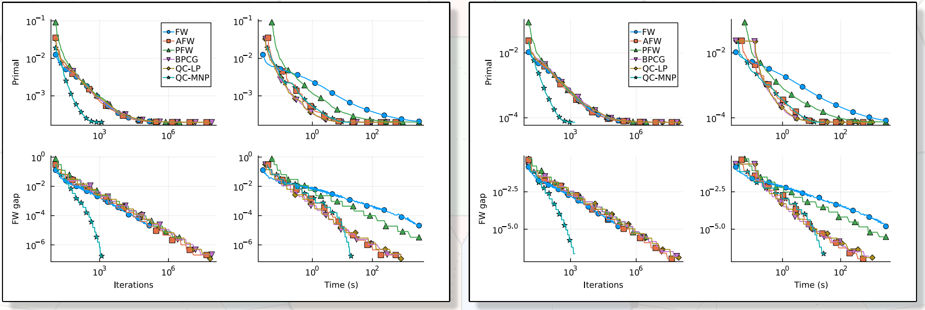


Figure: Entanglement detection for $\alpha \in \{0.25, 0.5\}$.

Split Projection

Computational Experiments

[Halbey et al., 2025]

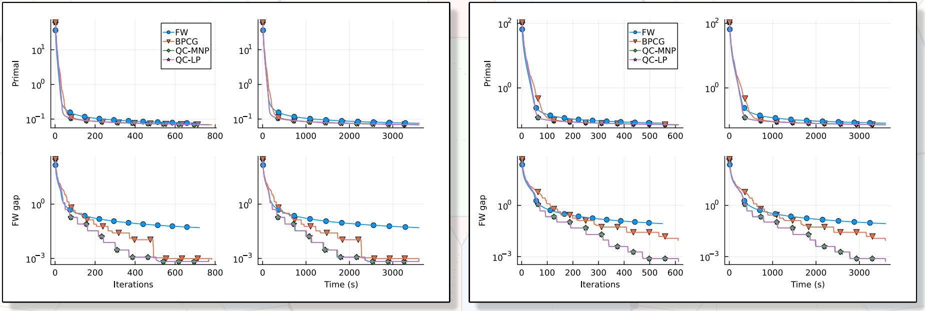
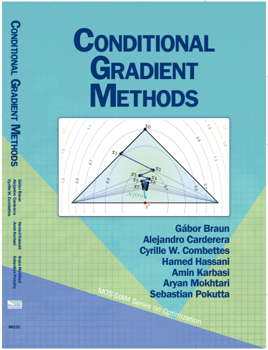


Figure: Projection onto $B(n) \cap B_2$ for $n \in \{300, 500\}$.

If you want to learn more...

Thank you!



Conditional Gradient Methods

Gábor Braun, Alejandro Carderera, Cyrille W Combettes, Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Sebastian Pokutta

https://conditional-gradients.org/
https://arxiv.org/abs/2211.14103

MOS-SIAM Series on Optimization

References I

F. Bach. On the effectiveness of Richardson extrapolation in machine learning. *arXiv preprint 2002.02835v3*, July 2020.

G. Braun and S. Pokutta. The matching polytope does not admit fully-polynomial size relaxation schemes. *Proceeedings of SODA*, 2015a.

G. Braun and S. Pokutta. The matching polytope does not admit fully-polynomial size relaxation schemes. *IEEE Transactions on Information Theory*, 61(10):1–11, 2015b.

G. Braun, S. Pokutta, and D. Zink. Inapproximability of combinatorial problems via small LPs and SDPs. *Proceeedings of STOC*, 2015.

G. Braun, R. Jain, T. Lee, and S. Pokutta. Information-theoretic approximations of the nonnegative rank. *Computational Complexity*, 26(1): 147–197, 2017a.

G. Braun, S. Pokutta, and D. Zink. Lazifying Conditional Gradient Algorithms. *Proceedings of the International Conference on Machine Learning (ICML)*, 2017b.

G. Braun, S. Pokutta, D. Tu, and S. Wright. Blended Conditional Gradients: the unconditioning of conditional gradients. *Proceedings of ICML*, 2019a.

G. Braun, S. Pokutta, and D. Zink. Lazifying Conditional Gradient Algorithms. *Journal of Machine Learning Research (JMLR)*, 20(71):1–42, 2019b.

G. Braun, A. Carderera, C. W. Combettes, H. Hassani, A. Karbasi, A. Mokthari, and S. Pokutta. *Conditional Gradient Methods*. to appear in MOS-SIAM Series on Optimization, 1 2025.

A. Carderera, J. Diakonikolas, C. Y. Lin, and S. Pokutta. Parameter-free Locally Accelerated Conditional Gradients. *Proceedings of ICML*, 2 2021.

C. W. Combettes and S. Pokutta. Complexity of Linear Minimization and Projection on Some Sets. *Operations Research Letters*, 49, 7 2021.

C. W. Combettes, C. Spiegel, and S. Pokutta. Projection-Free Adaptive Gradients for Large-Scale Optimization. *preprint*, 10 2020.

J. Diakonikolas, A. Carderera, and S. Pokutta. Locally Accelerated Conditional Gradients. *Proceedings of AISTATS*, 2020.

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

R. M. Freund, P. Grigas, and R. Mazumder. 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.

D. Garber and E. Hazan. Playing non-linear games with linear oracles. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 420–428. IEEE, 2013.

D. Garber and O. Meshi. Linear-memory and decomposition-invariant linearly convergent conditional gradient algorithm for structured polytopes. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1001–1009. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6115-linear-memory-and-decomposition-invariant-linearly-convergent-conditional-gradient-algorithm-for-structured-polytopes.pdf.

J. Halbey, S. Rakotomandimby, M. Besançon, S. Designolle, and S. Pokutta. Efficient Quadratic Corrections for Frank-Wolfe Algorithms. *preprint*, 6 2025.

References II

E. Hazan and S. Kale. Projection-free online learning. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

E. Hazan and H. Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271, 2016.

C. A. Holloway. An extension of the Frank and Wolfe method of feasible directions. *Mathematical Programming*, 6:14–27, Dec. 1974. doi: 10.1007/BF01580219.

M. Jaggi. Revisiting Frank-Wolfe: projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435, 2013.

T. Kerdreux, A. d'Aspremont, and S. Pokutta. Restarting Frank-Wolfe. *Proceedings of AISTATS*, 2019.

T. Kerdreux, A. d'Aspremont, and S. Pokutta. Projection-Free Optimization on Uniformly Convex Sets. *Proceedings of AISTATS*, 1 2021.

T. Kerdreux, A. d'Aspremont, and S. Pokutta. Local and Global Uniform Convexity Conditions. *to appear in Special Issue of Fields Institute Communications*, 1 2025.

S. Lacoste-Julien. Convergence rate of Frank-Wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 496–504. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5925-on-the-global-linear-convergence-of-frank-wolfe-optimization-variants.pdf.

G. Lan and Y. Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.

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

S. J. Reddi, S. Sra, B. Póczos, and A. Smola. Stochastic Frank-Wolfe methods for nonconvex optimization. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1244–1251. IEEE, 2016.

T. Rothvoss. The matching polytope has exponential extension complexity. In *Symposium on Theory of Computing*, pages 263–272, 2014.

K. Tsuji, K. Tanaka, and S. Pokutta. Pairwise Conditional Gradients without Swap Steps and Sparser Kernel Herding. *Proceedings of ICML*, 5 2022.