~~Breaking the cycle~~: Flexible block-iterative analysis for the Frank-Wolfe algorithm

ISMP 2024, Montréal, QC

Zev Woodstock*, Gábor Braun, and Sebastian Pokutta

Zuse Institute Berlin (ZIB) & Technische Universität Berlin
Interactive Optimization and Learning (IOL) Lab

July 2024

*- also James Madison University starting Aug. 2024

Flexible Block-Coordinate Frank-Wolfe Algorithm

1. Motivation
2. Our approach
3. Analysis
4. Numerical experiments

Problem setting

Given m nonempty closed convex sets $C_i \subset \mathbb{R}^{n_i}$ with $i \in \{1, \ldots, m\} =: I$ and a smooth function $f: \mathbb{R}^N \to \mathbb{R}$ with $N = \sum_{i \in I} n_i$, solve

$$\underset{\boldsymbol{x} \in C_1 \times \ldots \times C_m}{\text{minimize}} \quad f(\boldsymbol{x}). \tag{1}$$

Applications: matrix factorization, SVM training, sequence labeling, splitting, ...

Problem setting

Given m nonempty closed convex sets $C_i \subset \mathbb{R}^{n_i}$ with $i \in \{1, \ldots, m\} =: I$ and a smooth function $f : \mathbb{R}^N \to \mathbb{R}$ with $N = \sum_{i \in I} n_i$, solve

$$\underset{\boldsymbol{x} \in C_1 \times \ldots \times C_m}{\text{minimize}} \quad f(\boldsymbol{x}). \tag{1}$$

Applications: matrix factorization, SVM training, sequence labeling, splitting, ...

Two families of first-order methods to solve (1): projection methods and Frank-Wolfe AKA "CG" methods, which use linear minimization oracles.

$$\mathrm{proj}_C(\boldsymbol{x}) = \underset{\boldsymbol{v} \in C}{\mathrm{Argmin}} \, \|\boldsymbol{x} - \boldsymbol{v}\|^2 \qquad \mathrm{LMO}_C(\boldsymbol{x}) \in \underset{\boldsymbol{v} \in C}{\mathrm{Argmin}} \, \langle \boldsymbol{x} \mid \boldsymbol{v} \rangle \tag{2}$$

Problem setting

Given m nonempty closed convex sets $C_i \subset \mathbb{R}^{n_i}$ with $i \in \{1, \ldots, m\} =: I$ and a smooth function $f : \mathbb{R}^N \to \mathbb{R}$ with $N = \sum_{i \in I} n_i$, solve

$$\underset{\boldsymbol{x} \in C_1 \times \ldots \times C_m}{\text{minimize}} \quad f(\boldsymbol{x}). \tag{1}$$

Applications: matrix factorization, SVM training, sequence labeling, splitting, ...

Two families of first-order methods to solve (1): projection methods and Frank-Wolfe AKA "CG" methods, which use linear minimization oracles.

$$\text{proj}_C(\boldsymbol{x}) = \underset{\boldsymbol{v} \in C}{\text{Argmin}} \, \|\boldsymbol{x} - \boldsymbol{v}\|^2 \qquad \text{LMO}_C(\boldsymbol{x}) \in \underset{\boldsymbol{v} \in C}{\text{Argmin}} \, \langle \boldsymbol{x} \mid \boldsymbol{v} \rangle \tag{2}$$

[Combettes/Pokutta, '21]: For many $\underbrace{\text{constraints}}$, C, proj_C is **more expensive** than LMO_C.
(e.g., nuclear norm ball, ℓ_1 ball, probability simplex, Birkhoff polytope, general LP, ...)

Problem setting

Given m nonempty closed convex sets $C_i \subset \mathbb{R}^{n_i}$ with $i \in \{1, \ldots, m\} =: I$ and a smooth function $f \colon \mathbb{R}^N \to \mathbb{R}$ with $N = \sum_{i \in I} n_i$, solve

$$\underset{\boldsymbol{x} \in C_1 \times \ldots \times C_m}{\text{minimize}} \quad f(\boldsymbol{x}). \tag{1}$$

Applications: matrix factorization, SVM training, sequence labeling, splitting, ...

For $\boldsymbol{x} \in \mathbb{R}^N$ with components $\boldsymbol{x} = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m)$ $(\boldsymbol{x}_i \in \mathbb{R}^{n_i})$,

$$\mathrm{LMO}_{C_1 \times \ldots \times C_m}(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m) = (\mathrm{LMO}_{C_1} \boldsymbol{x}^1, \ldots, \mathrm{LMO}_{C_m} \boldsymbol{x}^m) \tag{\$\$\$}$$

Problem setting

Given m nonempty closed convex sets $C_i \subset \mathbb{R}^{n_i}$ with $i \in \{1, \ldots, m\} =: I$ and a smooth function $f : \mathbb{R}^N \to \mathbb{R}$ with $N = \sum_{i \in I} n_i$, solve

$$\underset{\boldsymbol{x} \in C_1 \times \ldots \times C_m}{\text{minimize}} \quad f(\boldsymbol{x}). \tag{1}$$

Applications: matrix factorization, SVM training, sequence labeling, splitting, ...

For $\boldsymbol{x} \in \mathbb{R}^N$ with components $\boldsymbol{x} = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m)$ ($\boldsymbol{x}_i \in \mathbb{R}^{n_i}$),

$$\mathrm{LMO}_{C_1 \times \ldots \times C_m}(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m) = (\mathrm{LMO}_{C_1} \boldsymbol{x}^1, \ldots, \mathrm{LMO}_{C_m} \boldsymbol{x}^m) \tag{\$\$\$}$$

"Let's avoid computing so many LMOs per iteration!" (paraphrased)

– [Patriksson, '98], [Lacoste-Julien et al., 2013], [Beck et al., 2015], [Wang et al., 2016], [Osokin et al., 2016], [Bomze et al., 2024], ...

(Generic) BCFW Algorithm

Known modes of convergence:

for $t = 0, 1$ **to** $\dots$ **do**
  Select $I_t \subset \{1, \dots, m\}$
  $\boldsymbol{g}_t \leftarrow \nabla f(\boldsymbol{x}_t)$
  for $i = 1$ **to** m **do**
    if $i \in I_t$ **then**
      $\boldsymbol{v}_t^i \leftarrow \mathrm{LMO}_i(\boldsymbol{g}_t^i)$
      $\gamma_t^i \leftarrow$ Step size
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i + \gamma_t^i(\boldsymbol{v}_t^i - \boldsymbol{x}_t^i)$
    else
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i$
    end if
  end for
end for

(Generic) BCFW Algorithm

for $t = 0, 1$ **to** $\ldots$ **do**
  Select $I_t \subset \{1, \ldots, m\}$
  $\boldsymbol{g}_t \leftarrow \nabla f(\boldsymbol{x}_t)$
  for $i = 1$ **to** m **do**
    if $i \in I_t$ **then**
      $\boldsymbol{v}_t^i \leftarrow \mathrm{LMO}_i(\boldsymbol{g}_t^i)$
      $\gamma_t^i \leftarrow$ Step size
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i + \gamma_t^i(\boldsymbol{v}_t^i - \boldsymbol{x}_t^i)$
    else
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i$
    end if
  end for
end for

Known modes of convergence:

- [Patriksson, 1998]:
  - Asymptotic convergence if f convex
  - Exact and Armijo linesearches fixed across all components $\gamma_t^i = \gamma_t$
  - Full update $(I_t = \{1, \ldots, m\})$
  - Deterministic essentially cyclic $(\exists K > 0)$:

$$I_t = \{\mathrm{i}_t\}, \text{ with } \{\mathrm{i}_t, \ldots, \mathrm{i}_{t+K}\} = \{1, \ldots, m\}$$

(Generic) BCFW Algorithm

for $t = 0, 1$ **to** $\ldots$ **do**
  Select $I_t \subset \{1, \ldots, m\}$
  $\boldsymbol{g}_t \leftarrow \nabla f(\boldsymbol{x}_t)$
  for $i = 1$ **to** m **do**
    if $i \in I_t$ **then**
      $\boldsymbol{v}_t^i \leftarrow \mathrm{LMO}_i(\boldsymbol{g}_t^i)$
      $\gamma_t^i \leftarrow$ Step size
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i + \gamma_t^i(\boldsymbol{v}_t^i - \boldsymbol{x}_t^i)$
    else
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i$
    end if
  end for
end for

Known modes of convergence:

- [Patriksson, 1998]:
  - Asymptotic convergence if f convex
  - Exact and Armijo linesearches fixed across all components $\gamma_t^i = \gamma_t$
  - Full update ($I_t = \{1, \ldots, m\}$)
  - Deterministic essentially cyclic ($\exists K > 0$):

$$I_t = \{\mathrm{i}_t\}, \text{ with } \{\mathrm{i}_t, \ldots, \mathrm{i}_{t+K}\} = \{1, \ldots, m\}$$

- [Beck et al., 2015]:
  - $\mathcal{O}(1/t)$ convergence (f convex)
  - open-loop, short-step, and backtracking γ_t^i
  - Deterministic cyclic updates

$$I_t = \{\mathrm{i}_t\}, \text{ with } \{\mathrm{i}_t, \ldots, \mathrm{i}_{t+m}\} = \{1, \ldots, m\}$$

(Generic) BCFW Algorithm

for $t = 0, 1$ **to** $\ldots$ **do**
  Select $I_t \subset \{1, \ldots, m\}$
  $\boldsymbol{g}_t \leftarrow \nabla f(\boldsymbol{x}_t)$
  for $i = 1$ **to** m **do**
    if $i \in I_t$ **then**
      $\boldsymbol{v}_t^i \leftarrow \mathrm{LMO}_i(\boldsymbol{g}_t^i)$
      $\gamma_t^i \leftarrow$ Step size
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i + \gamma_t^i(\boldsymbol{v}_t^i - \boldsymbol{x}_t^i)$
    else
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i$
    end if
  end for
end for

Known modes of convergence:

- Stochastic variants:
  - $\mathcal{O}(1/t)$ primal convergence rate (f convex)
  - Uniform singleton selection [Lacoste-Julien et al., 2013]
  - Non-uniform singleton selection (based on suboptimality criterion) [Osokin et al., 2016]
  - Uniform parallel selection with fixed block-sizes $|I_t| = p$ [Wang et al., 2016]

(Generic) BCFW Algorithm

for $t = 0, 1$ **to** $\ldots$ **do**
  Select $I_t \subset \{1, \ldots, m\}$
  $\boldsymbol{g}_t \leftarrow \nabla f(\boldsymbol{x}_t)$
  for $i = 1$ **to** m **do**
    if $i \in I_t$ **then**
      $\boldsymbol{v}_t^i \leftarrow \mathrm{LMO}_i(\boldsymbol{g}_t^i)$
      $\gamma_t^i \leftarrow$ Step size
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i + \gamma_t^i(\boldsymbol{v}_t^i - \boldsymbol{x}_t^i)$
    else
      $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i$
    end if
  end for
end for

Known modes of convergence:

- Stochastic variants:
  - $\mathcal{O}(1/t)$ primal convergence rate (f convex)
  - Uniform singleton selection [Lacoste-Julien et al., 2013]
  - Non-uniform singleton selection (based on suboptimality criterion) [Osokin et al., 2016]
  - Uniform parallel selection with fixed block-sizes $|I_t| = p$ [Wang et al., 2016]
- [Bomze et al., 2024]:
  - Linear convergence (KL condition $+ \cdots$)
  - Short-Step Chain (SSC) procedure: γ_t^i, $\boldsymbol{v}_t^i$
  - Full updates ($I_t = \{1, \ldots, m\}$)
  - Uniform singleton selection ($I_t = \{\mathrm{i}_t\}$)
  - Gauss-Southwell “greedy” singleton updates (based on suboptimality criterion).

Let's recap. . .

- Singleton updates:

  $\rightarrow$ cyclic, essentially cyclic, Gauss-Southwell, (uniform or non-uniform) random

- Parallel updates:

  $\rightarrow$ Full ($I_t = \{1, \ldots, m\}$), or uniformly-random blocks of fixed size $|I_t| = p$

What if my LMOs have very different costs? What if I only have 4 processor cores?

Let's recap. . .

- Singleton updates:

  → cyclic, essentially cyclic, Gauss-Southwell, (uniform or non-uniform) random

- Parallel updates:

  → Full ($I_t = \{1, \ldots, m\}$), or uniformly-random blocks of fixed size $|I_t| = p$

What if my LMOs have very different costs? What if I only have 4 processor cores?

What about. . .

- **deterministic** parallel updates?

Let's recap. . .

- Singleton updates:

  $\rightarrow$ cyclic, essentially cyclic, Gauss-Southwell, (uniform or non-uniform) random

- Parallel updates:

  $\rightarrow$ Full ($I_t = \{1, \dots, m\}$), or uniformly-random blocks of fixed size $|I_t| = p$

What if my LMOs have very different costs? What if I only have 4 processor cores?

What about. . .

- **deterministic** parallel updates?
- blocks with **different sizes**?

Let's recap...

- Singleton updates:

  → cyclic, essentially cyclic, Gauss-Southwell, (uniform or non-uniform) random
- Parallel updates:

  → Full ($I_t = \{1, \ldots, m\}$), or uniformly-random blocks of fixed size $|I_t| = p$

What if my LMOs have very different costs? What if I only have 4 processor cores?

What about...

- **deterministic** parallel updates?
- blocks with **different sizes**?
- **cost-aware** methodologies? (e.g., if some LMOs are numerically expensive, and others are cheap)

Flexible Block-Coordinate Frank-Wolfe Algorithm

1. Motivation
2. Our approach
3. Analysis
4. Numerical experiments

A bit of history

Assumption

There exists a positive integer K such that, for every iteration t,

$$(\forall 1 \leqslant i \leqslant m) \quad i \in \bigcup_{n=t}^{t+K-1} I_n. \tag{$\star$}$$

A bit of history

Assumption

There exists a positive integer K such that, for every iteration t,

$$(\forall 1 \leqslant i \leqslant m) \quad i \in \bigcup_{n=t}^{t+K-1} I_n. \tag{$\star$}$$

Allows for:

- Deterministic, variable-size, parallel updates

A bit of history

Assumption

There exists a positive integer K such that, for every iteration t,

$$(\forall 1 \leqslant i \leqslant m) \quad i \in \bigcup_{n=t}^{t+K-1} I_n. \tag{$\star$}$$

Allows for:

- Deterministic, variable-size, parallel updates
- Already known to converge: Full, cyclic, essentially cyclic, ...

A bit of history

Assumption

There exists a positive integer K such that, for every iteration t,

$$(\forall 1 \leqslant i \leqslant m) \quad i \in \bigcup_{n=t}^{t+K-1} I_n. \qquad (\star)$$

Allows for:

- Deterministic, variable-size, parallel updates
- Already known to converge: Full, cyclic, essentially cyclic, ...
- **"Lazy" updates**: Over K iterations, update expensive LMO(s) once, and update cheap LMOs multiple times.

A bit of history

Assumption

There exists a positive integer K such that, for every iteration t,

$$(\forall 1 \leqslant i \leqslant m) \quad i \in \bigcup_{n=t}^{t+K-1} I_n. \tag{$\star$}$$

Allows for:

- Deterministic, variable-size, parallel updates
- Already known to converge: Full, cyclic, essentially cyclic, ...
- **"Lazy" updates**: Over K iterations, update expensive LMO(s) once, and update cheap LMOs multiple times.

  $\rightarrow$ We can set the ratio of $\dfrac{\text{(expensive LMO evals)}}{\text{(cheap LMO evals)}} = \dfrac{1}{K}$ arbitrarily small.

A bit of history

Assumption

There exists a positive integer K such that, for every iteration t,

$$(\forall 1 \leqslant i \leqslant m) \quad i \in \bigcup_{n=t}^{t+K-1} I_n. \tag{$\star$}$$

To my knowledge, first appears in [Ottavy, 1988].

A bit of history

Assumption

There exists a positive integer K such that, for every iteration t,

$$(\forall 1 \leqslant i \leqslant m) \quad i \in \bigcup_{n=t}^{t+K-1} I_n. \tag{$\star$}$$

To my knowledge, first appears in [Ottavy, 1988].

Related to lazily updating Hessians in Newton's method [Shamanskii, 1967]

1967:
Canada
turns 100!

A bit of history

Assumption

There exists a positive integer K such that, for every iteration t,

$$(\forall 1 \leqslant i \leqslant m) \quad i \in \bigcup_{n=t}^{t+K-1} I_n. \tag{$\star$}$$

To my knowledge, first appears in [Ottavy, 1988].

Related to lazily updating Hessians in Newton's method [Shamanskii, 1967]

Apparently **never considered for F-W algorithms** before!?

1967:
Canada
turns 100!

Goals

Under Assumption $(\star)$, establish competitive convergence rates.

What we did:

- f convex: $\mathcal{O}(K/t)$ rate (for primal gap) using:
  - Short-step γ_t^i
  - An adaptive stepsize scheme γ_t^i
- f nonconvex: $\mathcal{O}(K/\sqrt{t})$ rate (for F-W optimality gap) using short-step γ_t^i
- Some conjectures and interesting analysis along the way. . .

Flexible Block-Coordinate Frank-Wolfe Algorithm

1. Motivation
2. Our approach
3. Analysis
4. Numerical experiments

Notation and Background

Frank Wolfe gaps

Recall $I = \{1, \ldots, m\}$. The **Frank-Wolfe gap** at $\boldsymbol{x} \in \mathbb{R}^N$ is

$$G_I(\boldsymbol{x}) = \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{x} - \mathrm{LMO}_{\times_{i \in I} C_i}(\nabla f(\boldsymbol{x})) \rangle$$

Notation and Background

Frank Wolfe gaps

Recall $I = \{1, \ldots, m\}$. The **Frank-Wolfe gap** at $\boldsymbol{x} \in \mathbb{R}^N$ is

$$G_I(\boldsymbol{x}) = \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{x} - \mathrm{LMO}_{\times_{i \in I} C_i}(\nabla f(\boldsymbol{x})) \rangle = \sum_{i \in I} \langle \nabla^i f(\boldsymbol{x}) \mid \boldsymbol{x}^i - \mathrm{LMO}_{C_i}(\nabla^i f(\boldsymbol{x})) \rangle.$$

Notation and Background

Frank Wolfe gaps

Recall $I = \{1, \ldots, m\}$. The **Frank-Wolfe gap** at $\boldsymbol{x} \in \mathbb{R}^N$ is

$$G_I(\boldsymbol{x}) = \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{x} - \mathrm{LMO}_{\times_{i\in I} C_i}(\nabla f(\boldsymbol{x}))\rangle = \sum_{i\in I} \langle \nabla^i f(\boldsymbol{x}) \mid \boldsymbol{x}^i - \mathrm{LMO}_{C_i}(\nabla^i f(\boldsymbol{x}))\rangle.$$

A **partial Frank-Wolfe gap** is given by

$$(\forall J \subset I) \quad G_J(\boldsymbol{x}) = \sum_{i\in J} \langle \nabla^i f(\boldsymbol{x}) \mid \boldsymbol{x}^i - \mathrm{LMO}_{C_i}(\nabla^i f(\boldsymbol{x}))\rangle$$

Notation and Background

Frank Wolfe gaps

Recall $I = \{1, \ldots, m\}$. The **Frank-Wolfe gap** at $\boldsymbol{x} \in \mathbb{R}^N$ is

$$G_I(\boldsymbol{x}) = \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{x} - \mathrm{LMO}_{\times_{i\in I} C_i}(\nabla f(\boldsymbol{x}))\rangle = \sum_{i\in I} \langle \nabla^i f(\boldsymbol{x}) \mid \boldsymbol{x}^i - \mathrm{LMO}_{C_i}(\nabla^i f(\boldsymbol{x}))\rangle.$$

A **partial Frank-Wolfe gap** is given by

$$(\forall J \subset I) \quad G_J(\boldsymbol{x}) = \sum_{i\in J} \langle \nabla^i f(\boldsymbol{x}) \mid \boldsymbol{x}^i - \mathrm{LMO}_{C_i}(\nabla^i f(\boldsymbol{x}))\rangle$$

Fact

(A) If $\boldsymbol{x} \in \times_{i\in I} C_i$, then $(\forall J \subset I) \quad G_J(\boldsymbol{x}) \geqslant 0$.

(B) $\boldsymbol{x}$ is a stationary point of (1) if and only if $\boldsymbol{x} \in \times_{i\in I} C_i$ and $G_I(x) = 0$.

Notation and Background

Frank Wolfe gaps

Recall $I = \{1, \ldots, m\}$. The **Frank-Wolfe gap** at $\boldsymbol{x} \in \mathbb{R}^N$ is

$$G_I(\boldsymbol{x}) = \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{x} - \mathrm{LMO}_{\times_{i \in I} C_i}(\nabla f(\boldsymbol{x})) \rangle = \sum_{i \in I} \langle \nabla^i f(\boldsymbol{x}) \mid \boldsymbol{x}^i - \mathrm{LMO}_{C_i}(\nabla^i f(\boldsymbol{x})) \rangle.$$

A **partial Frank-Wolfe gap** is given by

$$(\forall J \subset I) \quad G_J(\boldsymbol{x}) = \sum_{i \in J} \langle \nabla^i f(\boldsymbol{x}) \mid \boldsymbol{x}^i - \mathrm{LMO}_{C_i}(\nabla^i f(\boldsymbol{x})) \rangle$$

Fact

(A) If $\boldsymbol{x} \in \times_{i \in I} C_i$, then $(\forall J \subset I) \quad G_J(\boldsymbol{x}) \geqslant 0$.

(B) $\boldsymbol{x}$ is a stationary point of (1) if and only if $\boldsymbol{x} \in \times_{i \in I} C_i$ and $G_I(x) = 0$.

$\Rightarrow$ nonconvex convergence results typically show **first order criticality**: $G_I(\boldsymbol{x}_t) \to 0$.

Notation and Background

Smoothness and short-steps

For $L_f > 0$, the function f is L_f-**smooth** on a convex set C if

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in C) \quad f(\boldsymbol{y}) - f(\boldsymbol{x}) \leqslant \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L_f}{2} \|\boldsymbol{y} - \boldsymbol{x}\|^2.$$

Notation and Background

Smoothness and short-steps

For $L_f > 0$, the function f is L_f-**smooth** on a convex set C if

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in C) \quad f(\boldsymbol{y}) - f(\boldsymbol{x}) \leqslant \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L_f}{2} \|\boldsymbol{y} - \boldsymbol{x}\|^2.$$

For BCFW, this means

$$f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t) \leqslant \sum_{i \in I_t} \gamma_t^i \underbrace{\langle \nabla^i f(\boldsymbol{x}_t) \mid \boldsymbol{v}_t^i - \boldsymbol{x}_t^i \rangle}_{-G_i(\boldsymbol{x}_t)} + \frac{L_f}{2} (\gamma_t^i)^2 \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2.$$

Notation and Background

Smoothness and short-steps

For $L_f > 0$, the function f is L_f-**smooth** on a convex set C if

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in C) \quad f(\boldsymbol{y}) - f(\boldsymbol{x}) \leqslant \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L_f}{2} \|\boldsymbol{y} - \boldsymbol{x}\|^2.$$

For BCFW, this means

$$f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t) \leqslant \sum_{i \in I_t} \gamma_t^i \underbrace{\langle \nabla^i f(\boldsymbol{x}_t) \mid \boldsymbol{v}_t^i - \boldsymbol{x}_t^i \rangle}_{-G_i(\boldsymbol{x}_t)} + \frac{L_f}{2} (\gamma_t^i)^2 \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2.$$

To tighten the inequality, the stepsize

$$\gamma_t^i = \operatorname*{Argmin}_{\gamma \in [0,1]} \left(-\gamma G_i(\boldsymbol{x}_t) + \gamma^2 \frac{L_f}{2} \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2 \right) = \min \left\{ \frac{G_i(\boldsymbol{x}_t)}{L_f \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2}, 1 \right\}, \qquad \text{(short)}$$

is known as the componentwise **short step**.

Notation and Background

Smoothness and short-steps

For $L_f > 0$, the function f is L_f-**smooth** on a convex set C if

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in C) \quad f(\boldsymbol{y}) - f(\boldsymbol{x}) \leqslant \langle \nabla f(\boldsymbol{x}) \mid \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{L_f}{2} \|\boldsymbol{y} - \boldsymbol{x}\|^2.$$

For BCFW, this means

$$f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t) \leqslant \sum_{i \in I_t} \gamma_t^i \underbrace{\langle \nabla^i f(\boldsymbol{x}_t) \mid \boldsymbol{v}_t^i - \boldsymbol{x}_t^i \rangle}_{-G_i(\boldsymbol{x}_t)} + \frac{L_f}{2} (\gamma_t^i)^2 \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2.$$

To tighten the inequality, the stepsize

$$\gamma_t^i = \underset{\gamma \in [0,1]}{\operatorname{Argmin}} \left(-\gamma G_i(\boldsymbol{x}_t) + \gamma^2 \frac{L_f}{2} \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2 \right) = \min \left\{ \frac{G_i(\boldsymbol{x}_t)}{L_f \|\boldsymbol{v}_t^i - \boldsymbol{x}_t^i\|^2}, 1 \right\}, \qquad \text{(short)}$$

is known as the componentwise **short step**. Downside: requires upper-estimate of L_f.

Adaptive step-size algorithm for convex functions

Typical adaptive setup [Pedregosa et al., 2020], [Pokutta, 2023]:

Adaptive step-size algorithm for convex functions

Typical adaptive setup [Pedregosa et al., 2020], [Pokutta, 2023]:

1. Update γ_t^i based on an estimated the smoothness constant $\widetilde{M}$.

Adaptive step-size algorithm for convex functions

Typical adaptive setup [Pedregosa et al., 2020], [Pokutta, 2023]:

1. Update γ_t^i based on an estimated the smoothness constant $\widetilde{M}$.
2. If a desired inequality holds between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$: done.

Adaptive step-size algorithm for convex functions

Typical adaptive setup [Pedregosa et al., 2020], [Pokutta, 2023]:

1. Update γ_t^i based on an estimated the smoothness constant $\widetilde{M}$.
2. If a desired inequality holds between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$: done.
3. Else, increase $\widetilde{M} \leftarrow \tau \widetilde{M}$ by $\tau > 1$ and recompute $\boldsymbol{x}_{t+1}$ until the desired inequality holds.

Adaptive step-size algorithm for convex functions

Typical adaptive setup [Pedregosa et al., 2020], [Pokutta, 2023]:

1. Update γ_t^i based on an estimated the smoothness constant $\widetilde{M}$.
2. If a desired inequality holds between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$: done.
3. Else, increase $\widetilde{M} \leftarrow \tau\widetilde{M}$ by $\tau > 1$ and recompute $\boldsymbol{x}_{t+1}$ until the desired inequality holds.

Pros: No a-priori knowledge of L_f; sometimes we get larger steps.

Cons: Extra function and/or gradient evaluations.

Adaptive step-size algorithm for convex functions

Typical adaptive setup [Pedregosa et al., 2020], [Pokutta, 2023]:

1. Update γ_t^i based on an estimated the smoothness constant $\widetilde{M}$.
2. If a desired inequality holds between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$: done.
3. Else, increase $\widetilde{M} \leftarrow \tau \widetilde{M}$ by $\tau > 1$ and recompute $\boldsymbol{x}_{t+1}$ until the desired inequality holds.

Pros: No a-priori knowledge of L_f; sometimes we get larger steps.

Cons: Extra function and/or gradient evaluations.

Fact (Hazan & Luo, 2016)

Let f be convex and L_f-smooth. Then,

$$(\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N) \quad f(\boldsymbol{x}) - f(\boldsymbol{y}) - \langle \nabla f(\boldsymbol{y}) \mid \boldsymbol{x} - \boldsymbol{y} \rangle \geqslant \frac{\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|^2}{2L_f}.$$

Adaptive step-size algorithm for convex functions

Typical adaptive setup [Pedregosa et al., 2020], [Pokutta, 2023]:

1. Update γ_t^i based on an estimated the smoothness constant $\widetilde{M}$.
2. If (2*) holds between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$: done.
3. Else, increase $\widetilde{M} \leftarrow \tau\widetilde{M}$ by $\tau > 1$ and recompute $\boldsymbol{x}_{t+1}$ until (2*) holds.

Pros: No a-priori knowledge of L_f; sometimes we get larger steps.

Cons: Extra function and/or gradient evaluations.

Fact (Hazan & Luo, 2016)

Let f be convex and L_f-smooth. Then, for $\widetilde{M}$ sufficiently large,

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1}) - \langle \nabla f(\boldsymbol{x}_{t+1}) \mid \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle \geqslant \frac{\|\nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{x}_{t+1})\|^2}{2\widetilde{M}}. \tag{2*}$$

Progress lemma

Lemma (Progress bound via smoothness and convexity, short-step)

Let $\bigtimes_{i\in I} C_i \subset \mathcal{H}$ be a product of m nonempty compact convex sets, let f be convex and L_f-smooth, let D be the diameter of $\bigtimes_{i\in I} C_i$, and assume $(\star)$. Let $\boldsymbol{x}^*$ solve (1), and set $H_t = f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)$. Then

$$H_t - H_{t+K} \geqslant \begin{cases} H_t + A_t - \frac{KL_f D^2}{2}, & \text{if } H_t + A_t \geqslant KL_f D^2; \\ \dfrac{(H_t + A_t)^2}{2KL_f D^2}, & \text{if } H_t + A_t \leqslant KL_f D^2, \end{cases} \text{ where}$$

$$A_t = \sum_{k=1}^{K-1} G_{\underbrace{I_{t+k-1} \cap (I_{t+k} \cup \cdots \cup I_{t+K-1})}_{J_k}}(\boldsymbol{x}_{t+k}) \geqslant 0$$

A_t describes partial F-W gaps for **all re-activated components**.

Progress lemma

Lemma (Progress bound via smoothness and convexity, short-step)

Let $\bigtimes_{i\in I} C_i \subset \mathcal{H}$ be a product of m nonempty compact convex sets, let f be convex and L_f-smooth, let D be the diameter of $\bigtimes_{i\in I} C_i$, and assume $(\star)$. Let $\boldsymbol{x}^*$ solve (1), and set $H_t = f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)$. Then

$$H_t - H_{t+K} \geqslant \begin{cases} H_t + A_t - \frac{KL_f D^2}{2}, & \text{if } H_t + A_t \geqslant KL_f D^2; \\ \dfrac{(H_t + A_t)^2}{2KL_f D^2}, & \text{if } H_t + A_t \leqslant KL_f D^2, \end{cases} \text{ where}$$

$$A_t = \sum_{k=1}^{K-1} G_{\underbrace{I_{t+k-1} \cap (I_{t+k} \cup \cdots \cup I_{t+K-1})}_{J_k}}(\boldsymbol{x}_{t+k}) \geqslant \sum_{k=1}^{K-1} f(\boldsymbol{x}_{t+k}) - \min_{\substack{\boldsymbol{x}\in\bigtimes_{i\in I} C_i \\ \boldsymbol{x}^{I\setminus J_k} = \boldsymbol{x}_{t+k}^{I\setminus J_k}}} f(\boldsymbol{x}) \geqslant 0.$$

A_t describes partial F-W gaps for **all re-activated components**.

Progress lemma

Lemma (Progress bound via smoothness and convexity, short-step)

Let $\bigtimes_{i\in I} C_i \subset \mathcal{H}$ be a product of m nonempty compact convex sets, let f be convex and L_f-smooth, let D be the diameter of $\bigtimes_{i\in I} C_i$, and assume $(\star)$. Let $\boldsymbol{x}^*$ solve (1), and set $H_t = f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)$. Then

$$H_t - H_{t+K} \geqslant \begin{cases} H_t + A_t - \frac{KL_f D^2}{2}, & \text{if } H_t + A_t \geqslant KL_f D^2; \\ \dfrac{(H_t + A_t)^2}{2KL_f D^2}, & \text{if } H_t + A_t \leqslant KL_f D^2, \end{cases} \text{ where}$$

$$A_t = \sum_{k=1}^{K-1} G_{\underbrace{I_{t+k-1} \cap (I_{t+k} \cup \cdots \cup I_{t+K-1})}_{J_k}}(\boldsymbol{x}_{t+k}) \geqslant \sum_{k=1}^{K-1} f(\boldsymbol{x}_{t+k}) - \min_{\substack{\boldsymbol{x}\in\bigtimes_{i\in I} C_i \\ \boldsymbol{x}^{I\setminus J_k} = \boldsymbol{x}_{t+k}^{I\setminus J_k}}} f(\boldsymbol{x}) \geqslant 0.$$

A_t may explain good behavior in experiments.

Progress lemma

Lemma (Progress bound via smoothness and convexity, short-step)

Let $\times_{i\in I} C_i \subset \mathcal{H}$ be a product of m nonempty compact convex sets, let f be convex and L_f-smooth, let D be the diameter of $\times_{i\in I} C_i$, and assume $(\star)$. Let $\boldsymbol{x}^*$ solve (1), and set $H_t = f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)$. Then

$$H_t - H_{t+K} \geqslant \begin{cases} H_t + A_t - \frac{KL_f D^2}{2}, & \text{if } H_t + A_t \geqslant KL_f D^2; \\ \frac{(H_t + A_t)^2}{2KL_f D^2}, & \text{if } H_t + A_t \leqslant KL_f D^2, \end{cases} \quad \text{where}$$

$$A_t = \sum_{k=1}^{K-1} G_{\underbrace{I_{t+k-1} \cap (I_{t+k} \cup \cdots \cup I_{t+K-1})}_{J_k}}(\boldsymbol{x}_{t+k}) \geqslant \sum_{k=1}^{K-1} f(\boldsymbol{x}_{t+k}) - \min_{\substack{\boldsymbol{x}\in\times_{i\in I} C_i \\ \boldsymbol{x}^{I\setminus J_k} = \boldsymbol{x}_{t+k}^{I\setminus J_k}}} f(\boldsymbol{x}) \geqslant 0.$$

We don't know how to leverage A_ts for an improved rate!

Progress lemma

Lemma (Progress bound via smoothness and convexity, adaptive step size strategy)

Let $\bigtimes_{i\in I} C_i \subset \mathcal{H}$ be a product of m nonempty compact convex sets, let f be convex and L_f-smooth, let D be the diameter of $\bigtimes_{i\in I} C_i$, let $0 < \eta \leqslant 1 < \tau$ and $M_0 > 0$, and assume $(\star)$. Let $\boldsymbol{x}^*$ solve (1), and set $H_t = f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)$. Then

$$H_t - H_{t+K} \geqslant \begin{cases} H_t + A_t - \frac{K\max\{\eta^t M_0, \tau L_f\} D^2}{2}, & \text{if } H_t + A_t \geqslant K\max\{\eta^t M_0, \tau L_f\} D^2; \\ \frac{(H_t + A_t)^2}{2K\max\{\eta^t M_0, \tau L_f\} D^2}, & \text{if } H_t + A_t \leqslant K\max\{\eta^t M_0, \tau L_f\} D^2, \end{cases}$$

$$A_t = \sum_{k=1}^{K-1} G_{\underbrace{I_{t+k-1} \cap (I_{t+k} \cup \cdots \cup I_{t+K-1})}_{J_k}}(\boldsymbol{x}_{t+k}) \geqslant \sum_{k=1}^{K-1} f(\boldsymbol{x}_{t+k}) - \min_{\substack{\boldsymbol{x}\in \bigtimes_{i\in I} C_i \\ \boldsymbol{x}^{I\setminus J_k} = \boldsymbol{x}_{t+k}^{I\setminus J_k}}} f(\boldsymbol{x}) \geqslant 0.$$

A_t describes partial F-W gaps for **all re-activated components**.

Convex setting: flexible stepsizes

Theorem

Let $\bigtimes_{i\in I} C_i \subset \mathcal{H}$ be a product of m nonempty compact convex sets, let f be convex and L_f-smooth, let $\tau > 1 \geqslant \eta$ and $M_0 > 0$ be approximation parameters, let D be the diameter of $\bigtimes_{i\in I} C_i$, let $\boldsymbol{x}_0 \in \mathbb{R}^N$, let $\boldsymbol{x}^*$ solve (1), and assume ($\star$). Set $n_0 := \max\{\lceil \log(\tau L_f/(\eta M_0))/(K \log \eta)\rceil, 0\}$. Then,

$$f(\boldsymbol{x}_{nK}) - f(\boldsymbol{x}^*) \leqslant \begin{cases} \min\limits_{0\leqslant p\leqslant n-1} \left\{ \dfrac{K\eta^{pK} M_0 D^2}{2} - A_{pK} \right\} & \text{if } 1 \leqslant n \leqslant n_0 + 1 \\ \dfrac{2K\tau L_f D^2}{n - n_0 + \sum_{p=n_0}^{n} \frac{2A_{pK}}{f(\boldsymbol{x}_{n_0}) - f(\boldsymbol{x}^*)} + \left(\frac{A_{pK}}{f(\boldsymbol{x}_{n_0}) - f(\boldsymbol{x}^*)}\right)^2} & \text{if } n > n_0 + 1. \end{cases}$$

After t iterations, Adaptive-BCFW has evaluated f and ∇f at-most $2 + \lceil \log_\tau(L_f/\eta^t M_0)\rceil$ times.

Convex setting: flexible stepsizes

Theorem

Let $\bigtimes_{i\in I} C_i \subset \mathcal{H}$ be a product of m nonempty compact convex sets, let f be convex and L_f-smooth, let $\tau > 1 \geqslant \eta$ and $M_0 > 0$ be approximation parameters, let D be the diameter of $\bigtimes_{i\in I} C_i$, let $\boldsymbol{x}_0 \in \mathbb{R}^N$, let $\boldsymbol{x}^*$ solve (1), and assume ($\star$). Set $n_0 := \max\{\lceil \log(\tau L_f/(\eta M_0))/(K \log \eta)\rceil, 0\}$. Then,

$$f(\boldsymbol{x}_{nK}) - f(\boldsymbol{x}^*) \leqslant \begin{cases} \min\limits_{0\leqslant p\leqslant n-1} \left\{ \frac{K\eta^{pK} M_0 D^2}{2} - A_{pK} \right\} & \text{if } 1 \leqslant n \leqslant n_0 + 1 \\ \dfrac{2K\tau L_f D^2}{n - n_0 + \sum_{p=n_0}^{n} \frac{2A_{pK}}{f(\boldsymbol{x}_{n_0}) - f(\boldsymbol{x}^*)} + \left(\frac{A_{pK}}{f(\boldsymbol{x}_{n_0}) - f(\boldsymbol{x}^*)}\right)^2} & \text{if } n > n_0 + 1. \end{cases}$$

After t iterations, Adaptive-BCFW has evaluated f and ∇f at-most $2 + \lceil \log_\tau (L_f/\eta^t M_0)\rceil$ times.

$\rightarrow$ After t iterations, matches $\mathcal{O}(K/t)$ rate for convex cyclic setting

Corollary: Parallelized short-step BCFW

Corollary

Let $\bigtimes_{i \in I} C_i \subset \mathcal{H}$ be a product of m nonempty compact convex sets, let f be convex and L_f-smooth, let D be the diameter of $\bigtimes_{i \in I} C_i$, let $\boldsymbol{x}^*$ solve (1), and assume ($\star$). Then,

$$(\forall n \in \mathbb{N}) \quad f(\boldsymbol{x}_{nK}) - f(\boldsymbol{x}^*) \leqslant \begin{cases} \dfrac{KL_f D^2}{2} - A_0 & \text{if } n = 1 \\ \dfrac{2KL_f D^2}{n - 1 + \sum_{p=1}^{n} \frac{2A_{pK}}{f(\boldsymbol{x}_1) - f(\boldsymbol{x}^*)} + \left(\frac{A_{pK}}{f(\boldsymbol{x}_1) - f(\boldsymbol{x}^*)}\right)^2} & \text{if } n \geqslant 2. \end{cases}$$

Furthermore, Short-step BCFW requires one gradient evaluation per iteration.

Corollary: Parallelized short-step BCFW

Corollary

Let $\bigtimes_{i\in I} C_i \subset \mathcal{H}$ be a product of m nonempty compact convex sets, let f be convex and L_f-smooth, let D be the diameter of $\bigtimes_{i\in I} C_i$, let $\boldsymbol{x}^*$ solve (1), and assume ($\star$). Then,

$$(\forall n \in \mathbb{N}) \quad f(\boldsymbol{x}_{nK}) - f(\boldsymbol{x}^*) \leqslant \begin{cases} \dfrac{KL_f D^2}{2} - A_0 & \text{if } n = 1 \\ \dfrac{2KL_f D^2}{n - 1 + \sum_{p=1}^{n} \frac{2A_{pK}}{f(\boldsymbol{x}_1) - f(\boldsymbol{x}^*)} + \left(\frac{A_{pK}}{f(\boldsymbol{x}_1) - f(\boldsymbol{x}^*)}\right)^2} & \text{if } n \geqslant 2. \end{cases}$$

Furthermore, Short-step BCFW requires one gradient evaluation per iteration.

→ Matches rate **and** constant for non-block Short-step FW.

→ Easier to parallelize than Adaptive BCFW.

Nonconvex convergence

Theorem (Nonconvex convergence)

Let $\bigtimes_{i\in I} C_i \subset \mathcal{H}$ be a product of m nonempty compact convex sets with diameter D. Let ∇f be L_f-Lipschitz continuous on $\bigtimes_{i\in I} C_i$, set $H_0 = f(\boldsymbol{x}_0) - \inf f(\bigtimes_{i\in I} C_i)$. Suppose that $(\star)$ holds. Then, for every $n \in \mathbb{N}$, Short-step BCFW guarantees

$$\min_{0\leqslant p\leqslant n-1} G_I(\boldsymbol{x}_{pK}) \leqslant \frac{1}{n}\sum_{p=0}^{n-1} G_I(\boldsymbol{x}_{pK}) \leqslant \begin{cases} \frac{2H_0 - \sum_{p=0}^{n-1} A_{pK}}{n} + \frac{KL_f D^2}{2} & \text{if } n \leqslant \frac{2H_0}{KL_f D^2} \\ 2D\sqrt{\frac{H_0 K L_f}{n}} - \frac{\sum_{p=0}^{n-1} A_{pK}}{n} & \text{otherwise.} \end{cases}$$

In particular, there exists a subsequence $(n_k)_{k\in\mathbb{N}}$ such that $G_I(\boldsymbol{x}_{n_k K}) \to 0$, and every accumulation point of $(\boldsymbol{x}_{n_k K})_{k\in\mathbb{N}}$ is a stationary point of (1).

$\to$ Reactivated gap terms reappear!

Nonconvex convergence

Theorem (Nonconvex convergence)

Let $\bigtimes_{i\in I} C_i \subset \mathcal{H}$ be a product of m nonempty compact convex sets with diameter D. Let ∇f be L_f-Lipschitz continuous on $\bigtimes_{i\in I} C_i$, set $H_0 = f(\boldsymbol{x}_0) - \inf f(\bigtimes_{i\in I} C_i)$. Suppose that $(\star)$ holds. Then, for every $n \in \mathbb{N}$, Short-step BCFW guarantees

$$\min_{0\leqslant p\leqslant n-1} G_I(\boldsymbol{x}_{pK}) \leqslant \frac{1}{n}\sum_{p=0}^{n-1} G_I(\boldsymbol{x}_{pK}) \leqslant \begin{cases} \frac{2H_0 - \sum_{p=0}^{n-1} A_{pK}}{n} + \frac{KL_f D^2}{2} & \text{if } n \leqslant \frac{2H_0}{KL_f D^2} \\ 2D\sqrt{\frac{H_0 K L_f}{n}} - \frac{\sum_{p=0}^{n-1} A_{pK}}{n} & \text{otherwise.} \end{cases}$$

In particular, there exists a subsequence $(n_k)_{k\in\mathbb{N}}$ such that $G_I(\boldsymbol{x}_{n_k K}) \to 0$, and every accumulation point of $(\boldsymbol{x}_{n_k K})_{k\in\mathbb{N}}$ is a stationary point of (1).

$\to$ Reactivated gap terms reappear!

$\to$ After t iterations, minimal F-W gap converges like $\mathcal{O}(K/\sqrt{t})$.

Flexible Block-Coordinate Frank-Wolfe Algorithm

Experiments

Toy intersection problem (convex)

Find a matrix in the intersection of the spectrahedron $C_1 = \{X \in \mathbb{S}_+^{r\times r} \mid \mathrm{Trace}(X) = 1\}$ and the hypercube $C_2 = [-5, \mu]^{r\times r}$ ($\mu = 1/r$).

$$\underset{\boldsymbol{x}\in C_1\times C_2}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{x}^1 - \boldsymbol{x}^2\|^2$$

Experiments

Toy intersection problem (convex)

Find a matrix in the intersection of the spectrahedron $C_1 = \{X \in \mathbb{S}_+^{r\times r} \mid \mathrm{Trace}(X) = 1\}$ and the hypercube $C_2 = [-5, \mu]^{r\times r}$ ($\mu = 1/r$).

$$\underset{\boldsymbol{x}\in C_1\times C_2}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{x}^1 - \boldsymbol{x}^2\|^2$$

→ LMO_{C_1} is far more expensive than LMO_{C_2}.

→ We use Short-step BCFW to compare the following block activations: full, cyclic, permuted-cyclic, and "q-lazy":

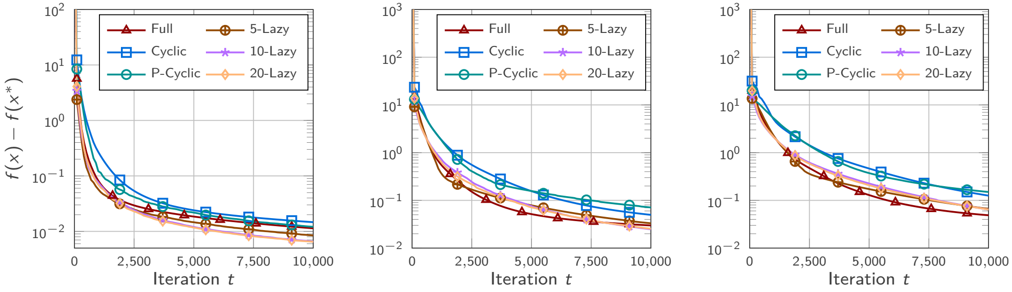
$$(\forall t \in \mathbb{N}) \quad I_t = \begin{cases} \{1,2\} & \text{if } t \equiv 0 \mod q; \\ \{2\} & \text{otherwise.} \end{cases} \tag{q-Lazy}$$

Experiments

Toy intersection problem (convex)

comparing block-activations: full, cyclic, permuted-cyclic, and

$$\underset{\boldsymbol{x}\in C_1\times C_2}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{x}^1-\boldsymbol{x}^2\|^2 \qquad (\forall t\in\mathbb{N}) \quad I_t=\begin{cases}\{1,2\} & \text{if } t\equiv 0 \mod q;\\ \{1\} & \text{otherwise.}\end{cases} \qquad (q\text{-lazy})$$



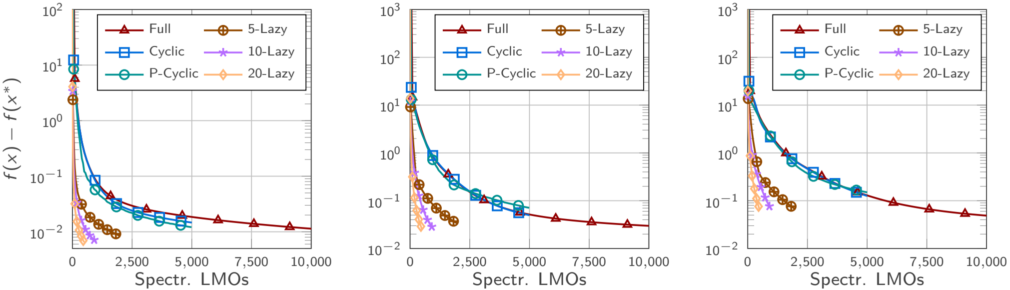
(a) $r=100$ (b) $r=300$ (c) $r=500$

Experiments

Toy intersection problem (convex)

comparing block-activations: full, cyclic, permuted-cyclic, and

$$\underset{\boldsymbol{x}\in C_1\times C_2}{\text{minimize}}\ \frac{1}{2}\|\boldsymbol{x}^1-\boldsymbol{x}^2\|^2 \qquad (\forall t\in\mathbb{N})\quad I_t=\begin{cases}\{1,2\} & \text{if } t\equiv 0 \mod q;\\ \{1\} & \text{otherwise.}\end{cases} \qquad (q\text{-lazy})$$



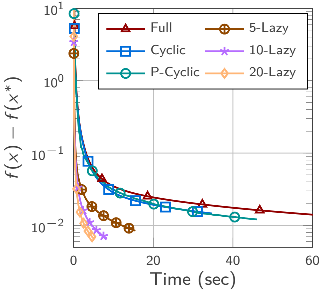
(d) $r=100$ (e) $r=300$ (f) $r=500$

Experiments

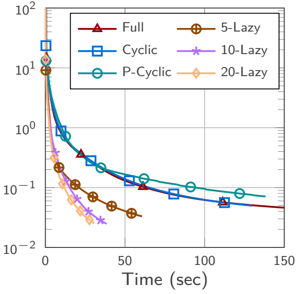
Toy intersection problem (convex)

comparing block-activations: full, cyclic, permuted-cyclic, and

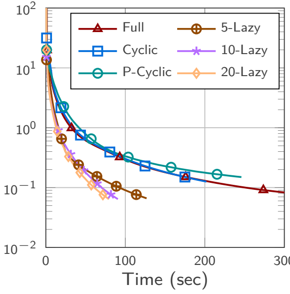
$$\underset{\boldsymbol{x}\in C_1\times C_2}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{x}^1-\boldsymbol{x}^2\|^2 \qquad (\forall t\in\mathbb{N}) \quad I_t=\begin{cases}\{1,2\} & \text{if } \ t\equiv 0 \mod q;\\ \{1\} & \text{otherwise.}\end{cases} \qquad (q\text{-lazy})$$



(g) $r=100$



(h) $r=300$



(i) $r=500$

Experiments

Toy Difference-of-Convex quadratic problem

Find a $2r \times r$ matrix such that its first $r \times r$ submatrix satisfies $\|X\|_\infty \leqslant 1$, and its second submatrix satisfies $\|X\|_{\text{nuc}} \leqslant 1$. To investigate BCFW when the number of components is large, we set $C_1 = \ldots = C_r = \{x \in \mathbb{R}^r \mid \|x\|_\infty \leqslant 1\}$ and $C_{r+1} = \{X \in \mathbb{R}^{r \times r} \mid \|X\|_{\text{nuc}} \leqslant 1\}$. For PSD $2r \times r$ matrices A and B, we seek to solve

$$\underset{\boldsymbol{x} \in \bigtimes_{1 \leqslant i \leqslant r+1} C_i}{\text{minimize}} \ \langle [x] \mid [x]A \rangle - \langle [x] \mid [x]B \rangle$$

$\rightarrow$ For each instance, we verify $A - B$ is indefinite.

$\rightarrow$ Problem is nonseparable

Experiments

Toy Difference-of-Convex quadratic problem

→ $\mathrm{LMO}_{C_{r+1}}$ is far more expensive than $(\mathrm{LMO}_{C_i})_{1\leqslant i\leqslant r}$.

→ We use Short-step BCFW to compare the following block activations: full, cyclic, permuted-cyclic, and "(p, q)-lazy":

$$(\forall t \in \mathbb{N}) \quad I_t = \begin{cases} I & \text{if } \; t \equiv 0 \pmod{q} \\ \{i_1, \ldots, i_p\} \subset_R I \setminus \{r+1\} & \text{otherwise.} \end{cases} \qquad ((p,q)\text{-Lazy})$$

Full update every q iterations; otherwise, update a random subset of p "cheap" coordinates in parallel.

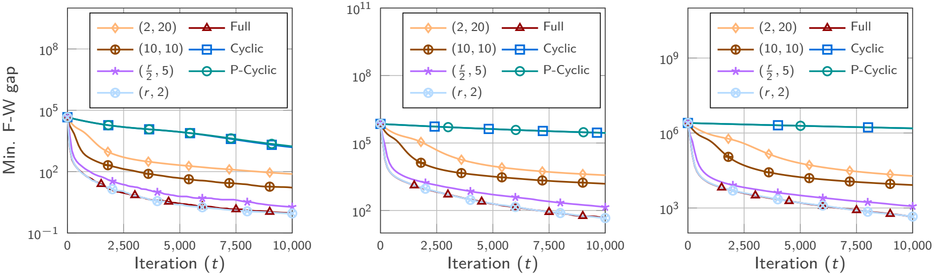
Experiments

Toy Difference-of-Convex quadratic problem

comparing full, cyclic, perm.-cyclic, and "(p, q)-lazy":

$$\underset{x \in \times_{1 \leqslant i \leqslant r+1} C_i}{\text{minimize}} \ \langle [x] \mid [x]A \rangle - \langle [x] \mid [x]B \rangle$$

$$I_t = \begin{cases} I & \text{if } t \equiv 0 \pmod{q} \\ \{i_1, \dots, i_p\} \subset_R I \setminus \{r+1\} & \text{otherwise.} \end{cases}$$



(j) $r = 100$ (k) $r = 300$ (l) $r = 500$

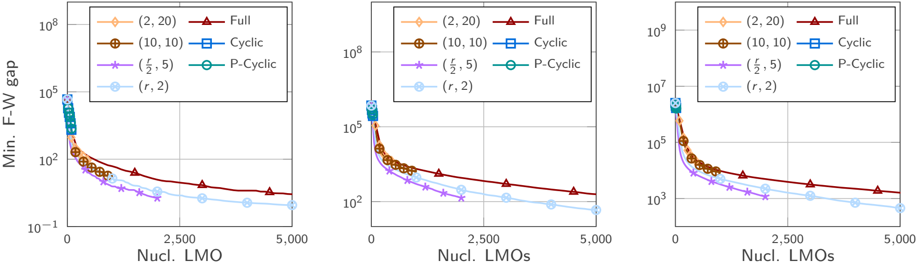
Experiments

Toy Difference-of-Convex quadratic problem

comparing full, cyclic, perm.-cyclic, and "(p, q)-lazy":

$$\underset{\boldsymbol{x} \in \bigtimes_{1 \leqslant i \leqslant r+1} C_i}{\text{minimize}} \ \langle [x] \mid [x]A \rangle - \langle [x] \mid [x]B \rangle$$

$$I_t = \begin{cases} I & \text{if } \ t \equiv 0 \pmod{q} \\ \{i_1, \ldots, i_p\} \subset_R I \setminus \{r+1\} & \text{otherwise.} \end{cases}$$



(m) $r = 100$ (n) $r = 300$ (o) $r = 500$

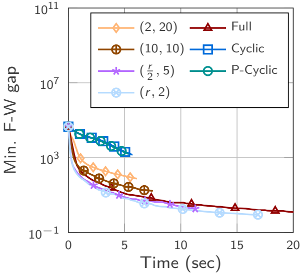
Experiments

Toy Difference-of-Convex quadratic problem

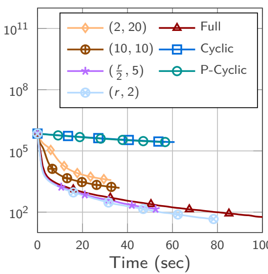
comparing full, cyclic, perm.-cyclic, and "(p, q)-lazy":

$$\underset{x \in \bigtimes_{1 \leqslant i \leqslant r+1} C_i}{\text{minimize}} \quad \langle [x] \mid [x]A \rangle - \langle [x] \mid [x]B \rangle$$

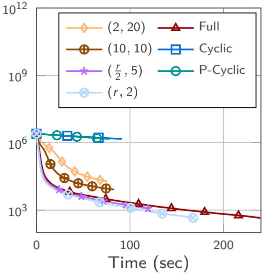
$$I_t = \begin{cases} I & \text{if} \quad t \equiv 0 \pmod{q} \\ \{i_1, \ldots, i_p\} \subset_R I \setminus \{r+1\} & \text{otherwise.} \end{cases}$$



(p) $r = 100$



(q) $r = 300$



(r) $r = 500$

Thank you for your attention!

References

A. Beck, E. Pauwels, and S. Sabach, The cyclic block conditional gradient method for convex optimization problems
SIAM J. Optim., vol. 25, no. 4, pp. 2024–2049, 2015

C. Combettes and S. Pokutta, Complexity of linear minimization and projection on some sets
Oper. Res. Lett., vol. 49, no. 4, pp. 565–571, 2021

P. L. Combettes and ZW, Signal recovery from inconsistent nonlinear observations
Proc. IEEE Int. Conf. Acoust. Speech Signal Process., pp 5872—5876, 2022.

P. L. Combettes and ZW, A variational inequality model for the construction of signals from inconsistent nonlinear equations
SIAM J. Imaging Sci., vol. 15, no. 1, pp. 84–109, 2022

M. Frank and P. Wolfe, An algorithm for quadratic programming
Naval Res. Logist. Quart., vol. 3, iss. 1–2, pp. 95–110, 1956

E. Hazan and H. Luo, Variance-Reduced and Projection-Free Stochastic Optimization
Proc. ICML, vol. 48, pp. 1263–1271, 2016

References

- C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations
  SIAM, Philadelphia, 1995.
- S. Lacoste-Julien, M. Jaggi, M. Schmidt, P. Pletscher, Block-Coordinate Frank-Wolfe Optimization for Structural SVMs
  Proc. ICML, vol. 28, pp. 53–61, 2013
- A. Osokin, J.-B. Alayrac, I. Lukasewitz, P. Dokania, S. Lacoste-Julien, Minding the Gaps for Block Frank-Wolfe Optimization of Structured SVMs
  Proc. ICML, vol. 48, pp. 593–602, 2016
- N. Ottavy, Strong convergence of projection-like methods in Hilbert spaces
  J. Optim. Theory Appl., vol. 56, pp. 433–461, 1988
- M. Patriksson, Decomposition methods for differentiable optimization problems over Cartesian product sets
  Comput. Optim. Appl., vol. 9, pp. 5–42, 1998

References

F. Pedregosa, G. Negiar, A. Askari, and M. Jaggi, Linearly convergent Frank-Wolfe with backtracking line-search
ICML, pp. 1–10, 2020

S. Pokutta, The Frank-Wolfe Algorithm: a Short Introduction
Jahresber. Dtsch. Math.-Ver., vol. 126, pp. 3—35, 2024

V. E. Shamanskii, A modification of Newton's method
Ukran. Mat. Zh., vol. 19, pp. 133–138, 1967 (in Russian)

Y.-X. Wang, V. Sadhanala, W. Dai, W. Neiswanger, S. Sra, E. Xing, Parallel and Distributed Block-Coordinate Frank-Wolfe Algorithms
Proc. ICML, vol. 48, pp. 1548–1557, 2016

ZW and S. Pokutta, Splitting the conditional gradient algorithm
arXiv:2311.05381, 2024