Splitting the Conditional Gradient Algorithm

INFORMS Annual Meeting

Zev Woodstock

October 2023

Results are joint work with...

Sebastian Pokutta
ZIB & Technische Universität Berlin

Interactive Optimization & Learning (IOL) Lab
iol.zib.de

Splitting the Conditional Gradient Algorithm

1. Motivation: History of splitting and CG / “Frank-Wolfe” algorithms
2. Algorithm design
3. Convergence guarantees

Algorithms for one constraint

Classical problem setup

Given a smooth function $f : \mathbb{R}^n \to \mathbb{R}$ and a nonempty compact convex set C,

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

Algorithms for one constraint

Classical problem setup

Given a smooth function $f : \mathbb{R}^n \to \mathbb{R}$ and a nonempty compact convex set C,

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

Two iterative first-order algorithms for solving (1)

Projected gradient descent:

Conditional gradient:

Algorithms for one constraint

Classical problem setup

Given a smooth function $f : \mathbb{R}^n \to \mathbb{R}$ and a nonempty compact convex set C,

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

Two iterative first-order algorithms for solving (1) differ in how $x \in C$ is enforced.

Projected gradient descent:

Conditional gradient:

Algorithms for one constraint

Classical problem setup

Given a smooth function $f : \mathbb{R}^n \to \mathbb{R}$ and a nonempty compact convex set C,

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

Two iterative first-order algorithms for solving (1) differ in how $x \in C$ is enforced.

Projected gradient descent: Requires the *projection onto C*, proj_C:

$$y \mapsto \underset{x \in C}{\arg\min} \|x - y\|^2 \tag{PROJ}$$

Conditional gradient:

Algorithms for one constraint

Classical problem setup

Given a smooth function $f : \mathbb{R}^n \to \mathbb{R}$ and a nonempty compact convex set C,

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

Two iterative first-order algorithms for solving (1) differ in how $x \in C$ is enforced.

Projected gradient descent: Requires the *projection onto C*, proj_C:

$$y \mapsto \arg\min_{x \in C} \|x - y\|^2 \qquad (\texttt{PROJ})$$

Conditional gradient: Requires the *linear minimization oracle of C*, LMO_C:

$$y \mapsto p \in \arg\min_{x \in C} \langle y \mid x \rangle \qquad (\texttt{LMO})$$

Algorithms for one constraint

Classical problem setup

Given a smooth function $f : \mathbb{R}^n \to \mathbb{R}$ and a nonempty compact convex set C,

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

Two iterative first-order algorithms for solving (1) differ in how $x \in C$ is enforced.

Projected gradient descent: Requires the *projection onto C*, proj_C:

$$y \mapsto \operatorname*{arg\,min}_{x \in C} \|x - y\|^2 \tag{PROJ}$$

Conditional gradient: Requires the *linear minimization oracle of C*, LMO_C:

$$y \mapsto p \in \operatorname*{arg\,min}_{x \in C} \langle y \mid x \rangle \tag{LMO}$$

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

What about multiple constraints?

Splitting problem setup

Given a smooth function $f\colon \mathbb{R}^n \to \mathbb{R}$ and compact convex sets $(C_i)_{i\in I}$ $(I = \{1, \ldots, m\})$,

$$\text{minimize } \; f(x) \; \text{ subject to } \; x \in \bigcap_{i\in I} C_i, \tag{$\star$}$$

Applications: data science, matrix decomposition, quantum computing, combinatorial graph theory

What about multiple constraints?

Splitting problem setup

Given a smooth function $f: \mathbb{R}^n \to \mathbb{R}$ and compact convex sets $(C_i)_{i \in I}$ $(I = \{1, \dots, m\})$,

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \bigcap_{i \in I} C_i, \tag{$\star$}$$

Applications: data science, matrix decomposition, quantum computing, combinatorial graph theory

Issue: Computing the projection or LMO for $\bigcap_{i \in I} C_i$ is prohibitively expensive.

What about multiple constraints?

Splitting problem setup

Given a smooth function $f\colon \mathbb{R}^n \to \mathbb{R}$ and compact convex sets $(C_i)_{i\in I}$ $(I = \{1, \ldots, m\})$,

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \bigcap_{i\in I} C_i, \qquad (\star)$$

Applications: data science, matrix decomposition, quantum computing, combinatorial graph theory

Issue: Computing the projection or LMO for $\bigcap_{i\in I} C_i$ is prohibitively expensive.
Projection-based **splitting algorithms** (e.g., Forward-Backward, Douglas-Rachford, projective splitting, etc.), enforce constraints via projections onto the individual sets

Use $\mathrm{proj}_{C_1}, \quad \mathrm{proj}_{C_2}, \quad \ldots$ instead of $\mathrm{proj}_{(\bigcap_{i\in I} C_i)}$

What about multiple constraints?

Splitting problem setup

Given a smooth function $f : \mathbb{R}^n \to \mathbb{R}$ and compact convex sets $(C_i)_{i \in I}$ $(I = \{1, \ldots, m\})$,

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \bigcap_{i \in I} C_i, \tag{$\star$}$$

Applications: data science, matrix decomposition, quantum computing, combinatorial graph theory

Issue: Computing the projection or LMO for $\bigcap_{i \in I} C_i$ is prohibitively expensive.

Projection-based **splitting algorithms** (e.g., Forward-Backward, Douglas-Rachford, projective splitting, etc.), enforce constraints via projections onto the individual sets

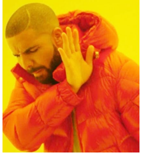
$$\text{Use } \mathrm{proj}_{C_1}, \quad \mathrm{proj}_{C_2}, \quad \ldots \quad \text{instead of} \quad \mathrm{proj}_{\left(\bigcap_{i \in I} C_i\right)}$$

Simpler tools $\to$ previously intractable problems become solvable on a larger scale.

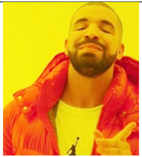
What if projections are too expensive?

LMO-based *splitting algorithms*, enforce constraints via LMOs for the individual sets

Use LMO_{C_1}, LMO_{C_2}, $\ldots$ instead of $\mathrm{LMO}_{(\bigcap_{i \in I} C_i)}$



$\mathrm{LMO}_{\bigcap_{i \in I} C_i}$



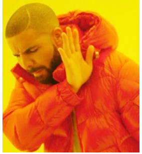
$\mathrm{LMO}_{C_1}, \ldots \mathrm{LMO}_{C_m}$

What if projections are too expensive?

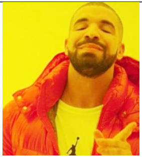
LMO-based *splitting algorithms*, enforce constraints via LMOs for the individual sets

Use $\mathrm{LMO}_{C_1}, \ \mathrm{LMO}_{C_2}, \ \ldots$ instead of $\mathrm{LMO}_{(\bigcap_{i \in I} C_i)}$

Relatively little has been done in this field.



$\mathrm{LMO}_{\bigcap_{i \in I} C_i}$



$\mathrm{LMO}_{C_1}, \ldots \mathrm{LMO}_{C_m}$

What if projections are too expensive?

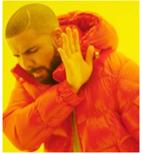
LMO-based *splitting algorithms*, enforce constraints via LMOs for the individual sets

Use $\mathrm{LMO}_{C_1}, \ \mathrm{LMO}_{C_2}, \ \ldots$ instead of $\mathrm{LMO}_{(\bigcap_{i\in I} C_i)}$

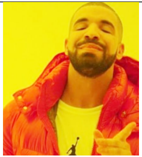
Relatively little has been done in this field.

→ Unlike projections, LMOs are discontinuous.

→ "CTRL+F / Replace proj with LMO" fails.



$\mathrm{LMO}_{\bigcap_{i\in I} C_i}$



$\mathrm{LMO}_{C_1}, \ldots \mathrm{LMO}_{C_m}$

What if projections are too expensive?

LMO-based *splitting algorithms*, enforce constraints via LMOs for the individual sets

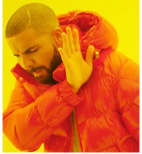
Use $\mathrm{LMO}_{C_1}, \ \mathrm{LMO}_{C_2}, \ \ldots$ instead of $\mathrm{LMO}_{(\bigcap_{i\in I} C_i)}$

Relatively little has been done in this field.

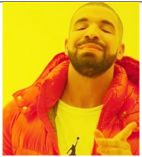
→ Unlike projections, LMOs are discontinuous.

→ “CTRL+F / Replace proj with LMO” fails.

→ “State-of-the-art” relies on inexact prox-based algorithms (mostly Augmented Lagrangians).



$\mathrm{LMO}_{\bigcap_{i\in I} C_i}$



$\mathrm{LMO}_{C_1}, \ldots \mathrm{LMO}_{C_m}$

Previous work

"Use a CG subroutine to approximate a projection" $\Rightarrow$ high iteration complexity
[He/Harchaoui, '15], [Liu et al., '19] [Millan et al., '21], [Kolmogorov/Pock, '21]

Previous work

“Use a CG subroutine to approximate a projection” $\Rightarrow$ high iteration complexity
[He/Harchaoui, ’15], [Liu et al., ’19] [Millan et al., ’21], [Kolmogorov/Pock, ’21]

Currently, lowest iteration complexity is $\mathcal{O}(m)$: one LMO per set.

Previous work

"Use a CG subroutine to approximate a projection" $\Rightarrow$ high iteration complexity
[He/Harchaoui, '15], [Liu et al., '19] [Millan et al., '21], [Kolmogorov/Pock, '21]

Currently, lowest iteration complexity is $\mathcal{O}(m)$: one LMO per set.

	$m = 2$	$m > 2$	f convex	f nonconvex	Analysis
[Pedregosa et al., '20]	✘	✘	✔	✔	CG
[Braun et al., '22]	✔	✘	✘($f = 0$)	✘	CG

Previous work

“Use a CG subroutine to approximate a projection” $\Rightarrow$ high iteration complexity
[He/Harchaoui, ’15], [Liu et al., ’19] [Millan et al., ’21], [Kolmogorov/Pock, ’21]

Currently, lowest iteration complexity is $\mathcal{O}(m)$: one LMO per set.

	$m = 2$	$m > 2$	f convex	f nonconvex	Analysis
[Pedregosa et al., ’20]	✘	✘	✔	✔	CG
[Braun et al., ’22]	✔	✘	✘($f = 0$)	✘	CG
[Gidel et al. ’18]	✔	✘	✔	✘	AL+CG

(✔)- requires additional structure on $(C_i)_{i \in I}$

Previous work

"Use a CG subroutine to approximate a projection" $\Rightarrow$ high iteration complexity [He/Harchaoui, '15], [Liu et al., '19] [Millan et al., '21], [Kolmogorov/Pock, '21]

Currently, lowest iteration complexity is $\mathcal{O}(m)$: one LMO per set.

	$m = 2$	$m > 2$	f convex	f nonconvex	Analysis
[Pedregosa et al., '20]	✘	✘	✔	✔	CG
[Braun et al., '22]	✔	✘	✘($f = 0$)	✘	CG
[Gidel et al. '18]	✔	✘	✔	✘	AL+CG
[Yurtsever et al. '19], [Silvetti-Falls et al. '20]	✔	✔	✔	✘	AL+CG
[Lan et al., '21]	(✔)	(✔)	✔	✘	CG

(✔)- requires additional structure on $(C_i)_{i \in I}$

Previous work

“Use a CG subroutine to approximate a projection” $\Rightarrow$ high iteration complexity
[He/Harchaoui, ’15], [Liu et al., ’19] [Millan et al., ’21], [Kolmogorov/Pock, ’21]

Currently, lowest iteration complexity is $\mathcal{O}(m)$: one LMO per set.

	$m = 2$	$m > 2$	f convex	f nonconvex	Analysis
[Pedregosa et al., ’20]	✘	✘	✔	✔	CG
[Braun et al., ’22]	✔	✘	✘($f = 0$)	✘	CG
[Gidel et al. ’18]	✔	✘	✔	✘	AL+CG
[Yurtsever et al. ’19], [Silvetti-Falls et al. ’20]	✔	✔	✔	✘	AL+CG
[Lan et al., ’21]	(✔)	(✔)	✔	✘	CG
[ZW/Pokutta ’23]	✔	✔	✔	✔	CG

(✔)- requires additional structure on $(C_i)_{i \in I}$

Splitting the Conditional Gradient Algorithm

1. Motivation: History of splitting and CG / “Frank-Wolfe” algorithms
2. Algorithm design
3. Convergence guarantees

Tools from the projection literature

Product space construction

- $\{\omega_i\}_{i\in I} \subset]0,1]$, $\sum_{i\in I} \omega_i = 1$ (e.g., $\omega_i \equiv 1/m$)
- $\mathcal{H} = \mathbb{R}^n$ and $\boldsymbol{\mathcal{H}} = \bigtimes_{i\in I} \mathcal{H}$, with inner product $\sum_{i\in I} \omega_i \langle \cdot \mid \cdot \rangle$
- *Diagonal subspace* of $\boldsymbol{\mathcal{H}}$: $\boldsymbol{D} = \{(x,\ldots,x) \,|\, x \in \mathcal{H}\}$

Projecting onto $\boldsymbol{D}$ amounts to computing an average

$$\mathrm{proj}_{\boldsymbol{D}}\, x = A^* A = A^* \sum_{i\in I} \omega_i \boldsymbol{x}^i .$$

Tools from the projection literature

Product space construction

- $\{\omega_i\}_{i\in I} \subset]0,1]$, $\sum_{i\in I}\omega_i = 1$ (e.g., $\omega_i \equiv 1/m$)
- $\mathcal{H} = \mathbb{R}^n$ and $\boldsymbol{\mathcal{H}} = \bigtimes_{i\in I}\mathcal{H}$, with inner product $\sum_{i\in I}\omega_i\langle\cdot \mid \cdot\rangle$
- *Diagonal subspace* of $\boldsymbol{\mathcal{H}}$: $\boldsymbol{D} = \{(x,\ldots,x) \mid x \in \mathcal{H}\}$
- *Block-averaging operator* and its adjoint:

$$A\colon \boldsymbol{\mathcal{H}} \to \mathcal{H}\colon (\boldsymbol{x}^1,\ldots,\boldsymbol{x}^m) \mapsto \sum_{i\in I}\omega_i\boldsymbol{x}^i \qquad A^*\colon x \mapsto (x,\ldots,x).$$

Projecting onto $\boldsymbol{D}$ amounts to computing an average

$$\operatorname{proj}_{\boldsymbol{D}} x = A^*A = A^*\sum_{i\in I}\omega_i\boldsymbol{x}^i.$$

Tools from the projection literature

Product space construction

- $\{\omega_i\}_{i\in I} \subset]0,1]$, $\sum_{i\in I} \omega_i = 1$ (e.g., $\omega_i \equiv 1/m$)
- $\mathcal{H} = \mathbb{R}^n$ and $\boldsymbol{\mathcal{H}} = \bigtimes_{i\in I} \mathcal{H}$, with inner product $\sum_{i\in I} \omega_i \langle \cdot \mid \cdot \rangle$
- *Diagonal subspace* of $\boldsymbol{\mathcal{H}}$: $\boldsymbol{D} = \{(x,\dots,x) \mid x \in \mathcal{H}\}$
- *Block-averaging operator* and its adjoint:

$$A\colon \boldsymbol{\mathcal{H}} \to \mathcal{H}\colon (\boldsymbol{x}^1,\dots,\boldsymbol{x}^m) \mapsto \sum_{i\in I} \omega_i \boldsymbol{x}^i \qquad A^*\colon x \mapsto (x,\dots,x).$$

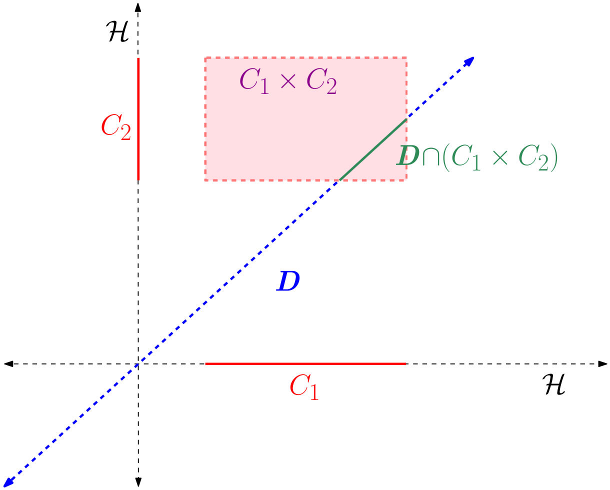
Projecting onto $\boldsymbol{D}$ amounts to computing an average

$$\mathrm{proj}_{\boldsymbol{D}}\, x = A^*A = A^* \sum_{i\in I} \omega_i \boldsymbol{x}^i.$$

Tools from the projection literature

Product space construction

- $\boldsymbol{\mathcal{H}} = \bigtimes_{i \in I} \mathcal{H}$
- $\boldsymbol{D} = \{(x, \ldots, x) \mid x \in \mathcal{H}\} \subset \boldsymbol{\mathcal{H}}$



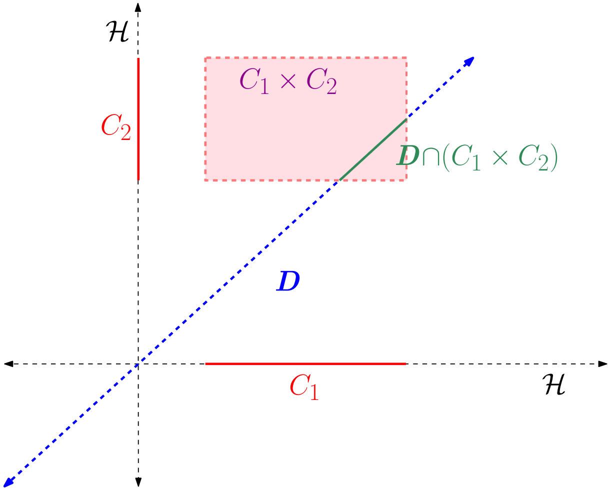
Tools from the projection literature

Product space construction

- $\boldsymbol{\mathcal{H}} = \bigtimes_{i \in I} \mathcal{H}$
- $\boldsymbol{D} = \{(x, \dots, x) \,|\, x \in \mathcal{H}\} \subset \boldsymbol{\mathcal{H}}$

Proposition (Reformulation of $\bigcap_{i \in I} C_i$)

$\boldsymbol{x} \in \boldsymbol{D} \cap \bigtimes_{i \in I} C_i$ if and only if
$\boldsymbol{x} = (x, \dots, x)$ and $x \in \bigcap_{i \in I} C_i$



Product space relaxation

$$\text{minimize}\quad f(x) \quad \text{subject to} \quad x \in \bigcap_{i\in I} C_i, \qquad (\star)$$

$$\underset{\boldsymbol{x}\in \times_{i\in I} C_i}{\text{minimize}} \quad \underbrace{f(A\boldsymbol{x}) + \lambda_t \operatorname{dist}^2_{\boldsymbol{D}}(\boldsymbol{x})}_{F_{\lambda_t}(\boldsymbol{x})}. \quad (*)$$

Product space relaxation

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \bigcap_{i \in I} C_i, \qquad (\star)$$

admits the equivalent reformulation (via the 0-∞ indicator function ι_D)

$$\underset{\boldsymbol{x} \in \boldsymbol{D} \cap \times_{i \in I} C_i}{\text{minimize}} \; f(A\boldsymbol{x}) = \underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \; f(A\boldsymbol{x}) + \iota_{\boldsymbol{D}}(\boldsymbol{x}).$$

$$\underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \; \underbrace{f(A\boldsymbol{x}) + \lambda_t \operatorname{dist}^2_{\boldsymbol{D}}(\boldsymbol{x})}_{F_{\lambda_t}(\boldsymbol{x})}. \quad (*)$$

Product space relaxation

$$\text{minimize} \ f(x) \quad \text{subject to} \quad x \in \bigcap_{i \in I} C_i, \tag{$\star$}$$

admits the equivalent reformulation (via the 0-∞ indicator function ι_D)

$$\underset{\boldsymbol{x} \in \boldsymbol{D} \cap \times_{i \in I} C_i}{\text{minimize}} \ f(A\boldsymbol{x}) = \underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \ f(A\boldsymbol{x}) + \iota_{\boldsymbol{D}}(\boldsymbol{x}).$$

Relaxation (for $\lambda_t \geqslant 0$)

$$\underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \ \underbrace{f(A\boldsymbol{x}) + \lambda_t \mathrm{dist}^2_{\boldsymbol{D}}(\boldsymbol{x})}_{F_{\lambda_t}(\boldsymbol{x})}. \quad (*)$$

Product space relaxation

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \bigcap_{i \in I} C_i, \qquad (\star)$$

admits the equivalent reformulation (via the 0-∞ indicator function ι_D)

$$\underset{\boldsymbol{x} \in \boldsymbol{D} \cap \times_{i \in I} C_i}{\text{minimize}} \; f(A\boldsymbol{x}) = \underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \; f(A\boldsymbol{x}) + \iota_{\boldsymbol{D}}(\boldsymbol{x}).$$

Relaxation (for $\lambda_t \geqslant 0$)

$$\underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \; \underbrace{f(A\boldsymbol{x}) + \lambda_t \mathrm{dist}^2_{\boldsymbol{D}}(\boldsymbol{x})}_{F_{\lambda_t}(\boldsymbol{x})}. \quad (*)$$

$\rightarrow$ Relaxation is tractable with vanilla CG!

$$\nabla F_{\lambda_t}(\boldsymbol{x}) = A^* \nabla f(A\boldsymbol{x}) + \lambda_t(\boldsymbol{x} - \mathrm{proj}_{\boldsymbol{D}} \boldsymbol{x})$$

$$\mathrm{LMO}_{\times_{i \in I} C_i}(\boldsymbol{x}) = (\mathrm{LMO}_{C_1}(\boldsymbol{x}^1), \ldots, \mathrm{LMO}_{C_m}(\boldsymbol{x}^m))$$

Product space relaxation

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \bigcap_{i \in I} C_i, \tag{$\star$}$$

admits the equivalent reformulation (via the 0-∞ indicator function ι_D)

$$\underset{\boldsymbol{x} \in \boldsymbol{D} \cap \times_{i \in I} C_i}{\text{minimize}} \ f(A\boldsymbol{x}) = \underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \ f(A\boldsymbol{x}) + \iota_{\boldsymbol{D}}(\boldsymbol{x}).$$

Relaxation (for $\lambda_t \geqslant 0$)

$$\underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \quad \underbrace{f(A\boldsymbol{x}) + \lambda_t \text{dist}^2_{\boldsymbol{D}}(\boldsymbol{x})}_{F_{\lambda_t}(\boldsymbol{x})}. \quad (*)$$

$\rightarrow$ Relaxation is tractable with vanilla CG!

$$\nabla F_{\lambda_t}(\boldsymbol{x}) = A^* \nabla f(A\boldsymbol{x}) + \lambda_t(\boldsymbol{x} - \text{proj}_{\boldsymbol{D}} \boldsymbol{x})$$

$$\text{LMO}_{\times_{i \in I} C_i}(\boldsymbol{x}) = (\text{LMO}_{C_1}(\boldsymbol{x}^1), \ldots, \text{LMO}_{C_m}(\boldsymbol{x}^m))$$

Pseudocode:

Product space relaxation

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \bigcap_{i \in I} C_i, \tag{$\star$}$$

admits the equivalent reformulation (via the 0-∞ indicator function ι_D)

$$\underset{\boldsymbol{x} \in \boldsymbol{D} \cap \times_{i \in I} C_i}{\text{minimize}} \quad f(A\boldsymbol{x}) = \underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \quad f(A\boldsymbol{x}) + \iota_{\boldsymbol{D}}(\boldsymbol{x}).$$

Relaxation (for $\lambda_t \geqslant 0$)

$$\underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \quad \underbrace{f(A\boldsymbol{x}) + \lambda_t \operatorname{dist}^2_{\boldsymbol{D}}(\boldsymbol{x})}_{F_{\lambda_t}(\boldsymbol{x})}. \quad (*)$$

$\to$ Relaxation is tractable with vanilla CG!

$$\nabla F_{\lambda_t}(\boldsymbol{x}) = A^* \nabla f(A\boldsymbol{x}) + \lambda_t(\boldsymbol{x} - \operatorname{proj}_{\boldsymbol{D}} \boldsymbol{x})$$

$$\operatorname{LMO}_{\times_{i \in I} C_i}(\boldsymbol{x}) = (\operatorname{LMO}_{C_1}(\boldsymbol{x}^1), \ldots, \operatorname{LMO}_{C_m}(\boldsymbol{x}^m))$$

Pseudocode:(A) Perform one CG step on $(*)$;

Product space relaxation

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \bigcap_{i \in I} C_i, \tag{$\star$}$$

admits the equivalent reformulation (via the 0-∞ indicator function ι_D)

$$\underset{\boldsymbol{x} \in \boldsymbol{D} \cap \times_{i \in I} C_i}{\text{minimize}} \quad f(A\boldsymbol{x}) = \underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \quad f(A\boldsymbol{x}) + \iota_{\boldsymbol{D}}(\boldsymbol{x}).$$

Relaxation (for $\lambda_t \geqslant 0$)

$$\underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \quad \underbrace{f(A\boldsymbol{x}) + \lambda_t \operatorname{dist}^2_{\boldsymbol{D}}(\boldsymbol{x})}_{F_{\lambda_t}(\boldsymbol{x})}. \quad (*)$$

$\to$ Relaxation is tractable with vanilla CG!

$$\nabla F_{\lambda_t}(\boldsymbol{x}) = A^* \nabla f(A\boldsymbol{x}) + \lambda_t(\boldsymbol{x} - \operatorname{proj}_{\boldsymbol{D}} \boldsymbol{x})$$

$$\operatorname{LMO}_{\times_{i \in I} C_i}(\boldsymbol{x}) = (\operatorname{LMO}_{C_1}(\boldsymbol{x}^1), \ldots, \operatorname{LMO}_{C_m}(\boldsymbol{x}^m))$$

Pseudocode:(A) Perform one CG step on $(*)$; (B) Update λ_t;

Product space relaxation

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \bigcap_{i \in I} C_i, \qquad (\star)$$

admits the equivalent reformulation (via the 0-∞ indicator function ι_D)

$$\underset{\boldsymbol{x} \in \boldsymbol{D} \cap \times_{i \in I} C_i}{\text{minimize}} \; f(A\boldsymbol{x}) = \underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \; f(A\boldsymbol{x}) + \iota_{\boldsymbol{D}}(\boldsymbol{x}).$$

Relaxation (for $\lambda_t \geqslant 0$)

$$\underset{\boldsymbol{x} \in \times_{i \in I} C_i}{\text{minimize}} \; \underbrace{f(A\boldsymbol{x}) + \lambda_t \operatorname{dist}^2_{\boldsymbol{D}}(\boldsymbol{x})}_{F_{\lambda_t}(\boldsymbol{x})}. \quad (*)$$

$\rightarrow$ Relaxation is tractable with vanilla CG!

$$\nabla F_{\lambda_t}(\boldsymbol{x}) = A^* \nabla f(A\boldsymbol{x}) + \lambda_t(\boldsymbol{x} - \operatorname{proj}_{\boldsymbol{D}} \boldsymbol{x})$$

$$\mathrm{LMO}_{\times_{i \in I} C_i}(\boldsymbol{x}) = (\mathrm{LMO}_{C_1}(\boldsymbol{x}^1), \ldots, \mathrm{LMO}_{C_m}(\boldsymbol{x}^m))$$

Pseudocode:(A) Perform one CG step on $(*)$; (B) Update λ_t; (C) $t \leftarrow t + 1$.

The algorithm

Split Conditional Gradient (SCG) Algorithm

Require: Point $x_0 \in \sum_{i \in I} \omega_i C_i$, smooth function f, weights $\{\omega_i\}_{i \in I} \subset]0,1]$ such that $\sum_{i \in I} \omega_i = 1$

1: **for** $t = 0, 1$ **to** $\ldots$ **do**

2: Choose penalty parameter $\lambda_t \in]0, +\infty[$

3: Choose step size $\gamma_t \in]0, 1]$

4: $g_t \leftarrow \nabla f(x_t)$

5: **for** $i = 1$ **to** m **do**

6: $\boldsymbol{v}_t^i \leftarrow \mathrm{LMO}_i(g_t + \lambda_t(\boldsymbol{x}_t^i - x_t))$

7: $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i + \gamma_t(\boldsymbol{v}_t^i - \boldsymbol{x}_t^i)$

8: **end for**

9: $x_{t+1} \leftarrow \sum_{i \in I} \omega_i \boldsymbol{x}_{t+1}^i$

10: **end for**

Practical advantages:

$\rightarrow$ Uses individual LMOs

$\rightarrow$ m LMO calls per iteration.

$\rightarrow$ Line 9: speeds up feasibility.

The algorithm

Split Conditional Gradient (SCG) Algorithm

Require: Point $x_0 \in \sum_{i\in I} \omega_i C_i$, smooth function f, weights $\{\omega_i\}_{i\in I} \subset]0,1]$ such that $\sum_{i\in I} \omega_i = 1$

1: **for** $t = 0, 1$ **to** $\ldots$ **do**
2: Choose penalty parameter $\lambda_t \in]0, +\infty[$
3: Choose step size $\gamma_t \in]0, 1]$
4: $g_t \leftarrow \nabla f(x_t)$
5: **for** $i = 1$ **to** m **do**
6: $\boldsymbol{v}_t^i \leftarrow \mathrm{LMO}_i(g_t + \lambda_t(\boldsymbol{x}_t^i - x_t))$
7: $\boldsymbol{x}_{t+1}^i \leftarrow \boldsymbol{x}_t^i + \gamma_t(\boldsymbol{v}_t^i - \boldsymbol{x}_t^i)$
8: **end for**
9: $x_{t+1} \leftarrow \sum_{i\in I} \omega_i \boldsymbol{x}_{t+1}^i$
10: **end for**

Practical advantages:

→ Uses individual LMOs

→ m LMO calls per iteration.

→ Line 9: speeds up feasibility.

Question:

→ Does it actually solve $(\star)$?

TL;DR: Yes.
$\gamma_t = \mathcal{O}(1/\sqrt{t})$ and $\lambda_t = \mathcal{O}(\ln t)$ work.

Why does averaging help?

$$\boldsymbol{x} \in \boldsymbol{D} \cap \bigtimes_{i \in I} C_i \Rightarrow A\boldsymbol{x} \in \bigcap_{i \in I} C_i,$$

so a feasible average is easier to satisfy than a feasible component!

Why does averaging help?

$$\boldsymbol{x} \in \boldsymbol{D} \cap \bigtimes_{i \in I} C_i \Rightarrow A\boldsymbol{x} \in \bigcap_{i \in I} C_i,$$

so a feasible average is easier to satisfy than a feasible component!

Proposition

$A\boldsymbol{x}_t \in \bigcap_{i \in I} C_i$ if and only if $\mathrm{proj}_D(\boldsymbol{x}) \in \bigtimes_{i \in I} C_i$.

Why does averaging help?

$$\boldsymbol{x} \in \boldsymbol{D} \cap \bigtimes_{i \in I} C_i \Rightarrow A\boldsymbol{x} \in \bigcap_{i \in I} C_i,$$

so a feasible average is easier to satisfy than a feasible component!

Proposition

$A\boldsymbol{x}_t \in \bigcap_{i \in I} C_i$ if and only if $\mathrm{proj}_D(\boldsymbol{x}) \in \bigtimes_{i \in I} C_i$.

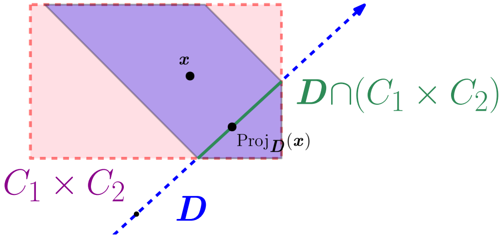


Figure: Darker shaded region $\{\boldsymbol{x} \in \boldsymbol{\mathcal{H}} \mid A\boldsymbol{x} \in \bigcap_{i \in I} C_i\}$ contains the segment $\boldsymbol{D} \cap \bigtimes_{i \in I} C_i$.

Splitting the Conditional Gradient Algorithm

1. Motivation: History of splitting and CG / “Frank-Wolfe” algorithms

2. Algorithm design

3. Convergence guarantees

Convergence of the subproblems

Proposition (Convergence of relaxed problems)

Let $(\lambda_t)_{t\in\mathbb{N}} \to +\infty$. For every $t \in \mathbb{N}$, set $F_t = f \circ A + \lambda_t \operatorname{dist}^2_{\boldsymbol{D}} /2 + \iota_{\times_{i\in I} C_i}$. Then

1. F_t converges pointwise to $f \circ A + \iota_{\boldsymbol{D}\cap\times_{i\in I} C_i}$.
2. F_t converges epigraphicallly to $f \circ A + \iota_{\boldsymbol{D}\cap\times_{i\in I} C_i}$.
3. ∂F_n converges graphically to $\partial(f \circ A + \iota_{\boldsymbol{D}\cap\times_{i\in I} C_i})$.

where epigraphical and graphical convergence are in, e.g., [Rockafellar/Wets, '09].

Proposition (Convergence of optimal values for $\lambda_t \nearrow +\infty$)

$$\lim_{t\to+\infty}\left(\inf_{\boldsymbol{x}\in\times_{i\in I} C_i} F_{\lambda_t}(\boldsymbol{x})\right) \to \inf_{\boldsymbol{x}\in\times_{i\in I} C_i}\left(\lim_{t\to\infty} F_{\lambda_t}(\boldsymbol{x})\right) = \inf_{x\in\bigcap_{i\in I} C_i} f(x).$$

Convex case

Theorem (Convex convergence)

Let f be convex and L_f-smooth, let $(C_i)_{i\in I}$ be nonempty compact convex subsets of $\mathcal{H}$ with diameters $\{R_i\}_{i\in I} \subset [0, +\infty[$ such that $\bigcap_{i\in I} C_i \neq \varnothing$, and for every $\lambda \geqslant 0$, set $F_\lambda \colon \boldsymbol{x} \mapsto f(A\boldsymbol{x}) + \frac{\lambda}{2}\operatorname{dist}^2_{\boldsymbol{D}}(\boldsymbol{x})$. Let $\lambda_0 > 0$ and $\lambda_{t+1} = \lambda_t + (\sqrt{t}+2)^{-2}$ and $\gamma_t = 2/(\sqrt{t}+2)$. Then

$$0 \leqslant F_{\lambda_t}(\boldsymbol{x}_t) - F_{\lambda_t}(\boldsymbol{x}_t^*) \leqslant \mathcal{O}\left(\frac{\ln t}{\sqrt{t}}\right)$$

In particular,

1. $F_{\lambda_t}(\boldsymbol{x}_t) \to \inf_{x\in\bigcap_{i\in I} C_i} f(x)$ and $\operatorname{dist}_{\boldsymbol{D}}(\boldsymbol{x}_t) \to 0$.
2. Every accumulation point $\boldsymbol{x}_\infty$ of $(\boldsymbol{x}_t)_{t\in\mathbb{N}}$ produces a solution $A\boldsymbol{x}_\infty \in \bigcap_{i\in I} C_i$ such that $f(A\boldsymbol{x}_\infty) = \inf_{x\in\bigcap_{i\in I} C_i} f(x)$.

Convex case

Theorem (Convex convergence)

Let f be convex and L_f-smooth, let $(C_i)_{i\in I}$ be nonempty compact convex subsets of $\mathcal{H}$ with diameters $\{R_i\}_{i\in I} \subset [0, +\infty[$ such that $\bigcap_{i\in I} C_i \neq \varnothing$, and for every $\lambda \geqslant 0$, set $F_\lambda \colon \boldsymbol{x} \mapsto f(A\boldsymbol{x}) + \frac{\lambda}{2}\operatorname{dist}^2_{\boldsymbol{D}}(\boldsymbol{x})$. Let $\lambda_0 > 0$ and $\lambda_{t+1} = \lambda_t + (\sqrt{t}+2)^{-2}$ and $\gamma_t = 2/(\sqrt{t}+2)$. Then

$$0 \leqslant F_{\lambda_t}(\boldsymbol{x}_t) - F_{\lambda_t}(\boldsymbol{x}_t^*) \leqslant \mathcal{O}\left(\frac{\ln t}{\sqrt{t}}\right)$$

In particular,

1. $F_{\lambda_t}(\boldsymbol{x}_t) \to \inf_{x\in\bigcap_{i\in I} C_i} f(x)$ and $\operatorname{dist}_{\boldsymbol{D}}(\boldsymbol{x}_t) \to 0$.
2. Every accumulation point $\boldsymbol{x}_\infty$ of $(\boldsymbol{x}_t)_{t\in\mathbb{N}}$ produces a solution $A\boldsymbol{x}_\infty \in \bigcap_{i\in I} C_i$ such that $f(A\boldsymbol{x}_\infty) = \inf_{x\in\bigcap_{i\in I} C_i} f(x)$.

We believe this rate can be improved!

Nonconvex case

Theorem (Nonconvex convergence)

Let f be L_f-smooth, let $(C_i)_{i\in I}$ be nonempty compact convex subsets of $\mathcal{H}$ with diameters $\{R_i\}_{i\in I} \subset [0,+\infty[$ such that $\bigcap_{i\in I} C_i \neq \varnothing$, and for every $\lambda \geqslant 0$, set $F_\lambda \colon \boldsymbol{x} \mapsto f(A\boldsymbol{x}) + \frac{\lambda}{2}\operatorname{dist}^2_{\boldsymbol{D}}(\boldsymbol{x})$. Let $\lambda_t = \sum_{k=0}^{t-1} 1/(k+1)$ and $\gamma_t = 1/\sqrt{t}$. Then,

$$0 \leqslant \frac{1}{t}\sum_{k=0}^{t-1} \langle \nabla F_{\lambda_k}(\boldsymbol{x}_k) \mid \boldsymbol{x}_k - \boldsymbol{v}_k \rangle \leqslant \mathcal{O}\left(\frac{\ln t}{\sqrt{t}} + \frac{1}{\sqrt{t}}\right).$$

In particular, there exists a subsequence $(t_k)_{k\in\mathbb{N}}$ such that

1. $(\langle \nabla F_{\lambda_{t_k}}(\boldsymbol{x}_{t_k}) \mid \boldsymbol{x}_{t_k} - \boldsymbol{v}_{t_k}\rangle)_{k\in\mathbb{N}} \to 0$ and $\operatorname{dist}_{\boldsymbol{D}}(\boldsymbol{x}_{t_k}) \to 0$.
2. Furthermore, every accumulation point $\boldsymbol{x}_\infty$ of $(\boldsymbol{x}_{t_k})_{k\in\mathbb{N}}$ yields a stationary point $A\boldsymbol{x}_\infty \in \bigcap_{i\in I} C_i$ of the problem $(\star)$.

Best-known rates / Future work

	$m = 2$	$m > 2$	f convex	f nonconvex
[Pedregosa et al., '20]	✘	✘	$\mathcal{O}\left(\frac{1}{t}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$
[Gidel et al. '18]	✔	✘	$\mathcal{O}\left(\frac{1}{t}\right)$	✘
[Yurtsever et al. '19], [Lan et al., '21](✔)	✔	✔	$\mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$	✘
[ZW/Pokutta '23]	✔	✔	✔	$\mathcal{O}\left(\frac{\ln t}{\sqrt{t}}\right)$

Best-known rates / Future work

	$m = 2$	$m > 2$	f convex	f nonconvex	
[Pedregosa et al., '20]	✘	✘	$\mathcal{O}\left(\frac{1}{t}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$	$\longleftarrow$
[Gidel et al. '18]	✔	✘	$\mathcal{O}\left(\frac{1}{t}\right)$	✘	
[Yurtsever et al. '19], [Lan et al., '21](✔)	✔	✔	$\mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$	✘	
[ZW/Pokutta '23]	✔	✔	✔	$\mathcal{O}\left(\frac{\ln t}{\sqrt{t}}\right)$	$\longleftarrow$

Usually, there is a quadratic speed-up from nonconvex and convex rates.

Best-known rates / Future work

	$m = 2$	$m > 2$	f convex	f nonconvex	
[Pedregosa et al., '20]	✘	✘	$\mathcal{O}\left(\frac{1}{t}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$	
[Gidel et al. '18]	✔	✘	$\mathcal{O}\left(\frac{1}{t}\right)$	✘	⟵
[Yurtsever et al. '19], [Lan et al., '21](✔)	✔	✔	$\mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$	✘	⟵
[ZW/Pokutta '23]	✔	✔	✔	$\mathcal{O}\left(\frac{\ln t}{\sqrt{t}}\right)$	

Usually, there is a quadratic speed-up from nonconvex and convex rates.

Is the gap between $m = 2$ and $m > 2$ actually necessary?

Thank you for your attention!

References

G. Braun, S. Pokutta, and R. Weismantel, Alternating linear minimization: revisiting von Neumann's alternating projections
preprint, arXiv: 2212.02933

R. Díaz Millán and O. P. Ferreira and L. F. Prudente, Alternating conditional gradient method for convex feasibility problems
Comput. Optim. Appl., vol. 80, pp. 245–269, 2021

G. Gidel, F. Pedregosa, and S. Lacoste-Julien, Frank-Wolfe splitting via augmented Lagrangian Method
Proc. AISTATS, pp. 1456–1465, 2018.

N. He and Z. Harchaoui, Semi-proximal mirror-prox for nonsmooth composite minimization
Proc. NeurIPS, vol. 28, 2015

V. Kolmogorov and T. Pock, One-sided Frank-Wolfe algorithms for saddle problems
Proc. ICML, PMLR, vol. 139, pp. 5665–5675, 2021

G. Lan, E. Romeijn, and Z. Zhou, Conditional gradient methods for convex optimization with general affine and nonlinear constraints
SIAM J. Optim., vol. 31, no. 3, pp. 2307–2339, 2021.

Refrences

Y-F. Liu, X. Liu, and S. Ma, On the nonergodic convergence rate of an inexact augmented Lagrangian framework for composite convex programming
Math. Oper. Res., vol. 44, no. 2 pp. 632–650, 2019

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

R. T. Rockafellar, and R. J-B Wets, Variational Analysis
Springer, 2009

A. Silveti-Falls, C. Molinari, and J. Fadili, Linearly convergent Frank-Wolfe with backtracking line-search
SIAM J. Optim., vol. 30, no. 4, pp. 2687-2725, 2020.

A. Yurtsever, O. Fercoq, and V. Cevher, A conditional-gradient-based augmented Lagrangian framework
Proc. ICML, pp. 7272–7281, 2019.