Frank-Wolfe Algorithms for Saddle Point problems



Gauthier Gidel[1,3] Tony Jebara[2] Simon Lacoste-Julien[3]

[1]INRIA Paris, Sierra Team [2]Department of CS, Columbia University

[3]Department of CS & OR (DIRO) Université de Montréal

12th July 2017

Overview

- Frank-Wolfe algorithm (FW) gained in popularity in the last couple of years.
- Main advantage: FW only needs LMO.
- Extend FW properties to solve saddle point problems[1].
- **Straightforward** extension but **Non trivial** analysis.

[1] Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. "Frank-Wolfe Algorithms for Saddle Point Problems". In: *AISTATS*. 2017.

Overview

- Frank-Wolfe algorithm (FW) gained in popularity in the last couple of years.
- Main advantage: FW only needs LMO.
- Extend FW properties to solve saddle point problems[1].
- **Straightforward** extension but **Non trivial** analysis.

Question for the audience: Call for application

[1] Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. "Frank-Wolfe Algorithms for Saddle Point Problems". In: *AISTATS*. 2017.

Saddle point and link with variational inequalities

Let $\mathcal{L} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{X}$ and $\mathcal{Y}$ are convex and compact.

$$\boxed{\text{Saddle point problem:} \quad \text{solve } \min_{\boldsymbol{x} \in \mathcal{X}} \max_{\boldsymbol{y} \in \mathcal{Y}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})}$$

A solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is called a ***Saddle Point***.

Saddle point and link with variational inequalities

Let $\mathcal{L} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{X}$ and $\mathcal{Y}$ are convex and compact.

$$\boxed{\text{Saddle point problem:} \quad \text{solve } \min_{\boldsymbol{x} \in \mathcal{X}} \max_{\boldsymbol{y} \in \mathcal{Y}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})}$$

A solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is called a ***Saddle Point***.

- **Necessary *stationary conditions:***

$$\langle \boldsymbol{x} - \boldsymbol{x}^*, \quad \nabla_x \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}^*) \rangle \geq 0$$

Saddle point and link with variational inequalities

Let $\mathcal{L} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{X}$ and $\mathcal{Y}$ are convex and compact.

$$\boxed{\text{Saddle point problem:} \quad \text{solve } \min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{y}\in\mathcal{Y}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})}$$

A solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is called a ***Saddle Point***.

- **Necessary *stationary conditions:***

$$\begin{aligned} \langle \boldsymbol{x} - \boldsymbol{x}^*, \quad \nabla_x \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}^*) \rangle &\geq 0 \\ \langle \boldsymbol{y} - \boldsymbol{y}^*, -\nabla_y \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}^*) \rangle &\geq 0 \end{aligned}$$

Saddle point and link with variational inequalities

Let $\mathcal{L} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{X}$ and $\mathcal{Y}$ are convex and compact.

$$\boxed{\text{Saddle point problem:} \quad \text{solve } \min_{\boldsymbol{x} \in \mathcal{X}} \max_{\boldsymbol{y} \in \mathcal{Y}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})}$$

A solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is called a ***Saddle Point***.

- **Necessary *stationary conditions:***

$$\langle \boldsymbol{x} - \boldsymbol{x}^*, \quad \nabla_x \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}^*) \rangle \geq 0$$
$$\langle \boldsymbol{y} - \boldsymbol{y}^*, -\nabla_y \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}^*) \rangle \geq 0$$

- ***Variational inequality:***

$$\forall \boldsymbol{z} \in \mathcal{X} \times \mathcal{Y} \quad \langle \boldsymbol{z} - \boldsymbol{z}^*, g(\boldsymbol{z}^*) \rangle \geq 0$$

where $(\boldsymbol{x}^*, \boldsymbol{y}^*) = \boldsymbol{z}^*$ and $g(\boldsymbol{z}) = (\nabla_x \mathcal{L}(\boldsymbol{z}), -\nabla_y \mathcal{L}(\boldsymbol{z}))$

Saddle point and link with variational inequalities

Let $\mathcal{L} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{X}$ and $\mathcal{Y}$ are convex and compact.

$$\boxed{\text{Saddle point problem:} \quad \text{solve } \min_{\boldsymbol{x} \in \mathcal{X}} \max_{\boldsymbol{y} \in \mathcal{Y}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})}$$

A solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is called a ***Saddle Point***.

- **Necessary *stationary conditions:***

$$\langle \boldsymbol{x} - \boldsymbol{x}^*, \quad \nabla_x \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}^*) \rangle \geq 0$$
$$\langle \boldsymbol{y} - \boldsymbol{y}^*, -\nabla_y \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}^*) \rangle \geq 0$$

- ***Variational inequality:***

$$\forall \boldsymbol{z} \in \mathcal{X} \times \mathcal{Y} \quad \langle \boldsymbol{z} - \boldsymbol{z}^*, g(\boldsymbol{z}^*) \rangle \geq 0$$

  where $(\boldsymbol{x}^*, \boldsymbol{y}^*) = \boldsymbol{z}^*$ and $g(\boldsymbol{z}) = (\nabla_x \mathcal{L}(\boldsymbol{z}), -\nabla_y \mathcal{L}(\boldsymbol{z}))$

- **Sufficient *condition*:** ***Global solution*** if $\mathcal{L}$ *convex-concave.* $\forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$

$$\boldsymbol{x}' \mapsto \mathcal{L}(\boldsymbol{x}', \boldsymbol{y}) \text{ is convex} \quad \text{and} \quad \boldsymbol{y}' \mapsto \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}') \text{ is concave.}$$

Motivations: games and robust learning

- ***Zero-sum games with two players:***

$$\min_{\boldsymbol{x}\in\Delta(I)} \max_{\boldsymbol{y}\in\Delta(J)} \boldsymbol{x}^\top M \boldsymbol{y}$$

[2] J. Wen, C. Yu, and R. Greiner. "Robust Learning under Uncertain Test Distributions: Relating Covariate Shift to Model Misspecification." In: *ICML*. 2014.

Motivations: games and robust learning

- ***Zero-sum games with two players:***

$$\min_{\boldsymbol{x}\in\Delta(I)} \max_{\boldsymbol{y}\in\Delta(J)} \boldsymbol{x}^\top M \boldsymbol{y}$$

- ***Robust learning:***[2] We want to learn

$$\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \ell\left(f_\theta(x_i), y_i\right) + \lambda\Omega(\theta)$$

with an uncertainty regarding the data:

$$\min_{\theta\in\Theta} \max_{w\in\Delta_n} \sum_{i=1}^{n} \omega_i \ell\left(f_\theta(x_i), y_i\right) + \lambda\Omega(\theta)$$

$$\boxed{\text{Minimize the } \textbf{worst case} \rightarrow \text{gives } \textbf{robustness}}$$

[2] J. Wen, C. Yu, and R. Greiner. "Robust Learning under Uncertain Test Distributions: Relating Covariate Shift to Model Misspecification." In: *ICML*. 2014.

Problem with Hard projection

The ***structured SVM:***

$$\min_{\omega \in \mathbb{R}^d} \lambda \Omega(\omega) + \frac{1}{n} \sum_{i=1}^{n} \underbrace{\max_{y \in \mathcal{Y}_i} \left(L_i(y) - \langle \omega, \phi_i(y) \rangle \right)}_{\text{structured empirical loss}}$$

Problem with Hard projection

The ***structured SVM:***

$$\min_{\omega \in \mathbb{R}^d} \lambda \Omega(\omega) + \frac{1}{n} \sum_{i=1}^{n} \underbrace{\max_{y \in \mathcal{Y}_i} \left(L_i(y) - \langle \omega, \phi_i(y) \rangle \right)}_{\text{structured empirical loss}}$$

Regularization: penalized $\rightarrow$ constrained.

$$\min_{\Omega(\omega) \leq \beta} \max_{\alpha \in \Delta(|\mathcal{Y}|)} b^T \alpha - \omega^T M \alpha$$

Problem with Hard projection

The ***structured SVM:***

$$\min_{\omega \in \mathbb{R}^d} \lambda \Omega(\omega) + \frac{1}{n} \sum_{i=1}^{n} \underbrace{\max_{y \in \mathcal{Y}_i} \left(L_i(y) - \langle \omega, \phi_i(y) \rangle \right)}_{\text{structured empirical loss}}$$

Regularization: penalized $\rightarrow$ constrained.

$$\min_{\Omega(\omega) \leq \beta} \max_{\alpha \in \Delta(|\mathcal{Y}|)} \; b^T \alpha - \omega^T M \alpha$$

Problem with Hard projection

The ***structured SVM:***

$$\min_{\omega \in \mathbb{R}^d} \lambda\Omega(\omega) + \frac{1}{n}\sum_{i=1}^{n} \underbrace{\max_{y \in \mathcal{Y}_i} \left(L_i(y) - \langle \omega, \phi_i(y)\rangle\right)}_{\text{structured empirical loss}}$$

Regularization: penalized → constrained.

$$\min_{\Omega(\omega)\leq\beta} \max_{\alpha\in\Delta(|\mathcal{Y}|)} b^T\alpha - \omega^T M\alpha$$

Problem with Hard projection

The ***structured SVM:***

$$\min_{\omega \in \mathbb{R}^d} \lambda \Omega(\omega) + \frac{1}{n} \sum_{i=1}^{n} \underbrace{\max_{y \in \mathcal{Y}_i} \left(L_i(y) - \langle \omega, \phi_i(y) \rangle \right)}_{\text{structured empirical loss}}$$

Regularization: penalized $\to$ constrained.

$$\min_{\Omega(\omega) \leq \beta} \max_{\alpha \in \Delta(|\mathcal{Y}|)} \; b^T \alpha - \omega^T M \alpha$$

Difficult to project when:

Problem with Hard projection

The ***structured SVM:***

$$\min_{\omega \in \mathbb{R}^d} \lambda \Omega(\omega) + \frac{1}{n} \sum_{i=1}^{n} \underbrace{\max_{y \in \mathcal{Y}_i} \left(L_i(y) - \langle \omega, \phi_i(y) \rangle \right)}_{\text{structured empirical loss}}$$

Regularization: penalized $\to$ constrained.

$$\min_{\Omega(\omega) \leq \beta} \max_{\alpha \in \Delta(|\mathcal{Y}|)} b^T \alpha - \omega^T M \alpha$$

Difficult to project when:

- ***Structured sparsity*** norm (group lasso norm).

Problem with Hard projection

The ***structured SVM:***

$$\min_{\omega\in\mathbb{R}^d} \lambda\Omega(\omega) + \frac{1}{n}\sum_{i=1}^{n} \underbrace{\max_{y\in\mathcal{Y}_i}\left(L_i(y) - \langle \omega, \phi_i(y)\rangle\right)}_{\text{structured empirical loss}}$$

Regularization: penalized $\rightarrow$ constrained.

$$\min_{\Omega(\omega)\leq\beta}\ \max_{\alpha\in\Delta(|\mathcal{Y}|)}\ b^T\alpha - \omega^T M\alpha$$

Difficult to project when:

- ***Structured sparsity*** norm (group lasso norm).
- The output $\mathcal{Y}$ is structured: ***exponential*** size.

Standard approaches in literature

- Projected gradient algorithm.

$$\boldsymbol{x}^{(t+1)} = P_{\mathcal{X}}(\boldsymbol{x}^{(t)} - \eta \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}))$$
$$\boldsymbol{y}^{(t+1)} = P_{\mathcal{Y}}(\boldsymbol{y}^{(t)} + \eta \nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}))$$

[3] GM Korpelevich. "The extragradient method for finding saddle points and other problems". In: *Matecon* (1976).

Standard approaches in literature

- Projected gradient algorithm.

$$\boldsymbol{x}^{(t+1)} = P_{\mathcal{X}}(\boldsymbol{x}^{(t)} - \eta \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}))$$
$$\boldsymbol{y}^{(t+1)} = P_{\mathcal{Y}}(\boldsymbol{y}^{(t)} + \eta \nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}))$$

- Projected extra-gradient[3].

$$\bar{\boldsymbol{x}}^{(t+1)} = P_{\mathcal{X}}(\boldsymbol{x}^{(t)} - \eta \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}))$$
$$\bar{\boldsymbol{y}}^{(t+1)} = P_{\mathcal{Y}}(\boldsymbol{y}^{(t)} + \eta \nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}))$$

Intuition: ***lookahead*** *move: look at what your opponent* ***would*** *do before deciding your move.*

$$\boldsymbol{x}^{(t+1)} = P_{\mathcal{X}}(\boldsymbol{x}^{(t)} - \eta \nabla_x \mathcal{L}(\bar{\boldsymbol{x}}^{(t+1)}, \bar{\boldsymbol{y}}^{(t+1)}))$$
$$\boldsymbol{y}^{(t+1)} = P_{\mathcal{Y}}(\boldsymbol{y}^{(t)} + \eta \nabla_y \mathcal{L}(\bar{\boldsymbol{x}}^{(t+1)}, \bar{\boldsymbol{y}}^{(t+1)}))$$

Prevents oscillations for non strongly convex objective.

[3] GM Korpelevich. "The extragradient method for finding saddle points and other problems". In: *Matecon* (1976).

Standard approaches in literature

- Gradient method works for non-smooth optimization, but

$$\frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}\right) \underset{T\to\infty}{\longrightarrow} (\boldsymbol{x}^*, \boldsymbol{y}^*)$$

[4] N. He and Z. Harchaoui. "Semi-proximal Mirror-Prox for Nonsmooth Composite Minimization". In: *NIPS*. 2015.

Standard approaches in literature

- Gradient method works for non-smooth optimization, but

$$\frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}\right) \underset{T\to\infty}{\longrightarrow} (\boldsymbol{x}^*, \boldsymbol{y}^*)$$

- Extragradient method works for smooth optimization,

$$(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \to (\boldsymbol{x}^*, \boldsymbol{y}^*)$$

[4] N. He and Z. Harchaoui. “Semi-proximal Mirror-Prox for Nonsmooth Composite Minimization”. In: *NIPS*. 2015.

Standard approaches in literature

- Gradient method works for non-smooth optimization, but

$$\frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{x}^{(t)},\boldsymbol{y}^{(t)}\right)\underset{T\to\infty}{\longrightarrow}(\boldsymbol{x}^*,\boldsymbol{y}^*)$$

- Extragradient method works for smooth optimization,

$$(\boldsymbol{x}^{(t)},\boldsymbol{y}^{(t)})\to(\boldsymbol{x}^*,\boldsymbol{y}^*)$$

  Even when projections are expensive:

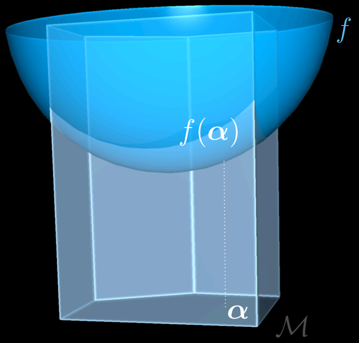
  Can use LMO to compute approximate projections[4].

[4] N. He and Z. Harchaoui. “Semi-proximal Mirror-Prox for Nonsmooth Composite Minimization”. In: *NIPS*. 2015.

The FW algorithm

Algorithm Frank-Wolfe algorithm

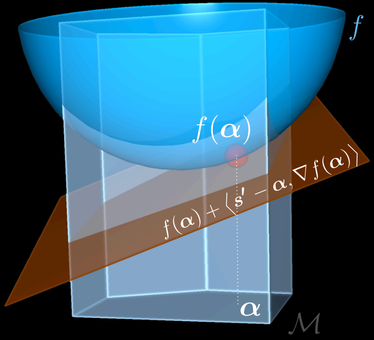
Let $\boldsymbol{x}^{(0)} \in \mathcal{X}$
for $t = 0 \dots T$ **do**
  Compute $\boldsymbol{r}^{(t)} = \nabla f(\boldsymbol{x}^{(t)})$
  Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{s} \in \mathcal{X}}{\operatorname{argmin}} \left\langle \boldsymbol{s}, \boldsymbol{r}^{(t)} \right\rangle$
  Compute $g_t := \left\langle \boldsymbol{x}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
  if $g_t \leq \epsilon$ **then return** $\boldsymbol{x}^{(t)}$
  Let $\gamma = \frac{2}{2+t}$ (or do line-search)
  Update $\boldsymbol{x}^{(t+1)} := (1-\gamma)\boldsymbol{x}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
end for



The FW algorithm

Algorithm Frank-Wolfe algorithm

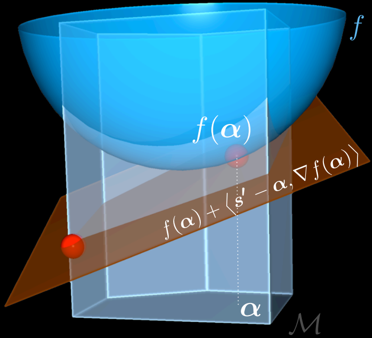
Let $\boldsymbol{x}^{(0)} \in \mathcal{X}$
for $t = 0 \dots T$ **do**
  Compute $\boldsymbol{r}^{(t)} = \nabla f(\boldsymbol{x}^{(t)})$
  Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{s} \in \mathcal{X}}{\operatorname{argmin}} \left\langle \boldsymbol{s}, \boldsymbol{r}^{(t)} \right\rangle$
  Compute $g_t := \left\langle \boldsymbol{x}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
  if $g_t \leq \epsilon$ **then return** $\boldsymbol{x}^{(t)}$
  Let $\gamma = \frac{2}{2+t}$ (or do line-search)
  Update $\boldsymbol{x}^{(t+1)} := (1-\gamma)\boldsymbol{x}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
end for



The FW algorithm

Algorithm Frank-Wolfe algorithm

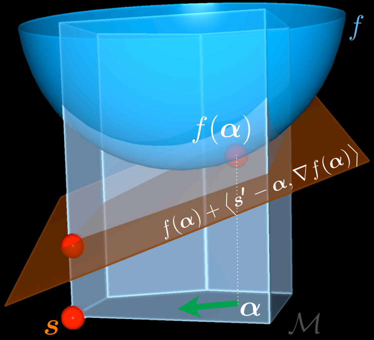
Let $\boldsymbol{x}^{(0)} \in \mathcal{X}$
for $t = 0 \dots T$ **do**
  Compute $\boldsymbol{r}^{(t)} = \nabla f(\boldsymbol{x}^{(t)})$
  Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{s} \in \mathcal{X}}{\operatorname{argmin}} \left\langle \boldsymbol{s}, \boldsymbol{r}^{(t)} \right\rangle$
  Compute $g_t := \left\langle \boldsymbol{x}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
  if $g_t \leq \epsilon$ **then return** $\boldsymbol{x}^{(t)}$
  Let $\gamma = \frac{2}{2+t}$ (or do line-search)
  Update $\boldsymbol{x}^{(t+1)} := (1-\gamma)\boldsymbol{x}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
end for



The FW algorithm

Algorithm Frank-Wolfe algorithm

Let $\boldsymbol{x}^{(0)} \in \mathcal{X}$
for $t = 0 \dots T$ **do**
  Compute $\boldsymbol{r}^{(t)} = \nabla f(\boldsymbol{x}^{(t)})$
  Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{s} \in \mathcal{X}}{\operatorname{argmin}} \left\langle \boldsymbol{s}, \boldsymbol{r}^{(t)} \right\rangle$
  Compute $g_t := \left\langle \boldsymbol{x}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
  if $g_t \leq \epsilon$ **then return** $\boldsymbol{x}^{(t)}$
  Let $\gamma = \frac{2}{2+t}$ (or do line-search)
  Update $\boldsymbol{x}^{(t+1)} := (1-\gamma)\boldsymbol{x}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
end for



SP-FW

Algorithm Saddle point FW algorithm

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

  Compute $\boldsymbol{r}^{(t)} := \begin{pmatrix} \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \\ -\nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \end{pmatrix}$

  Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{z} \in \mathcal{X} \times \mathcal{Y}}{\operatorname{argmin}} \left\langle \boldsymbol{z}, \boldsymbol{r}^{(t)} \right\rangle$

  Compute $g_t := \left\langle \boldsymbol{z}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$

  if $g_t \leq \epsilon$ **then return** $\boldsymbol{z}^{(t)}$

  Let $\gamma = \min\left(1, \frac{\nu}{C} g_t\right)$ **or** $\gamma = \frac{2}{2+t}$

  Update $\boldsymbol{z}^{(t+1)} := (1-\gamma)\boldsymbol{z}^{(t)} + \gamma \boldsymbol{s}^{(t)}$

end for

- Originally proposed by Hammond[4] with $\gamma_t = 1/(t+1)$.

[5] J. Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. MIT, 1984.

SP-FW

Algorithm Saddle point FW algorithm

1: **for** $t = 0 \dots T$ **do**
2: Compute $\boldsymbol{r}^{(t)} := \begin{pmatrix} \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \\ -\nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \end{pmatrix}$
3: Compute $\boldsymbol{s}^{(t)} \in \operatorname*{argmin}_{\boldsymbol{z} \in \mathcal{X} \times \mathcal{Y}} \left\langle \boldsymbol{z}, \boldsymbol{r}^{(t)} \right\rangle$
4: Compute $g_t := \left\langle \boldsymbol{z}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
5: **if** $g_t \leq \epsilon$ **then return** $\boldsymbol{z}^{(t)}$
6: Let $\gamma = \min\left(1, \frac{\nu}{C} g_t\right)$ **or** $\gamma = \frac{2}{2+t}$
7: Update $\boldsymbol{z}^{(t+1)} := (1-\gamma)\boldsymbol{z}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
8: **end for**

- Originally proposed by Hammond[4] with $\gamma_t = 1/(t+1)$.

[5] J. Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. MIT, 1984.

SP-FW

Algorithm Saddle point FW algorithm

1: **for** $t = 0 \dots T$ **do**
2: Compute $\boldsymbol{r}^{(t)} := \begin{pmatrix} \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \\ -\nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \end{pmatrix}$
3: Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{z} \in \mathcal{X} \times \mathcal{Y}}{\operatorname{argmin}} \left\langle \boldsymbol{z}, \boldsymbol{r}^{(t)} \right\rangle$
4: Compute $g_t := \left\langle \boldsymbol{z}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
5: **if** $g_t \leq \epsilon$ **then return** $\boldsymbol{z}^{(t)}$
6: Let $\gamma = \min\left(1, \frac{\nu}{C} g_t\right)$ **or** $\gamma = \frac{2}{2+t}$
7: Update $\boldsymbol{z}^{(t+1)} := (1-\gamma)\boldsymbol{z}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
8: **end for**

- Originally proposed by Hammond[4] with $\gamma_t = 1/(t+1)$.

[5] J. Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. MIT, 1984.

SP-FW

Algorithm Saddle point FW algorithm

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

  Compute $\boldsymbol{r}^{(t)} := \begin{pmatrix} \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \\ -\nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \end{pmatrix}$

  Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{z} \in \mathcal{X} \times \mathcal{Y}}{\operatorname{argmin}} \left\langle \boldsymbol{z}, \boldsymbol{r}^{(t)} \right\rangle$

  Compute $g_t := \left\langle \boldsymbol{z}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$

  if $g_t \leq \epsilon$ **then return** $\boldsymbol{z}^{(t)}$

  Let $\gamma = \min\left(1, \frac{\nu}{C} g_t\right)$ **or** $\gamma = \frac{2}{2+t}$

  Update $\boldsymbol{z}^{(t+1)} := (1-\gamma)\boldsymbol{z}^{(t)} + \gamma \boldsymbol{s}^{(t)}$

end for

- Originally proposed by Hammond[4] with $\gamma_t = 1/(t+1)$.

[5] J. Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. MIT, 1984.

SP-FW

Algorithm Saddle point FW algorithm

for $t = 0 \dots T$ **do**
  Compute $\boldsymbol{r}^{(t)} := \begin{pmatrix} \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \\ -\nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \end{pmatrix}$
  Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{z} \in \mathcal{X} \times \mathcal{Y}}{\operatorname{argmin}} \left\langle \boldsymbol{z}, \boldsymbol{r}^{(t)} \right\rangle$
  Compute $g_t := \left\langle \boldsymbol{z}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
  if $g_t \leq \epsilon$ **then return** $\boldsymbol{z}^{(t)}$
  Let $\gamma = \min\left(1, \frac{\nu}{C} g_t\right)$ **or** $\gamma = \frac{2}{2+t}$
  Update $\boldsymbol{z}^{(t+1)} := (1-\gamma)\boldsymbol{z}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
end for

- Originally proposed by Hammond[4] with $\gamma_t = 1/(t+1)$.

[5] J. Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. MIT, 1984.

SP-FW

Algorithm Saddle point FW algorithm

for $t = 0 \dots T$ **do**
  Compute $\boldsymbol{r}^{(t)} := \begin{pmatrix} \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \\ -\nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \end{pmatrix}$
  Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{z} \in \mathcal{X} \times \mathcal{Y}}{\operatorname{argmin}} \left\langle \boldsymbol{z}, \boldsymbol{r}^{(t)} \right\rangle$
  Compute $g_t := \left\langle \boldsymbol{z}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
  if $g_t \leq \epsilon$ **then return** $\boldsymbol{z}^{(t)}$
  Let $\gamma = \min\left(1, \frac{\nu}{C} g_t\right)$ **or** $\gamma = \frac{2}{2+t}$
  Update $\boldsymbol{z}^{(t+1)} := (1 - \gamma)\boldsymbol{z}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
end for

- Originally proposed by Hammond[4] with $\gamma_t = 1/(t+1)$.
- One can define FW extension with **away** step.

[5] J. Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. MIT, 1984.

SP-FW

Algorithm Saddle point FW algorithm

for $t = 0 \dots T$ **do**
  Compute $\boldsymbol{r}^{(t)} := \begin{pmatrix} \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \\ -\nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \end{pmatrix}$
  Compute $\boldsymbol{s}^{(t)} \in \operatorname*{argmin}_{\boldsymbol{z} \in \mathcal{X} \times \mathcal{Y}} \left\langle \boldsymbol{z}, \boldsymbol{r}^{(t)} \right\rangle$
  Compute $g_t := \left\langle \boldsymbol{z}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
  if $g_t \leq \epsilon$ **then return** $\boldsymbol{z}^{(t)}$
  Let $\gamma = \min\left(1, \frac{\nu}{C} g_t\right)$ **or** $\gamma = \frac{2}{2+t}$
  Update $\boldsymbol{z}^{(t+1)} := (1-\gamma)\boldsymbol{z}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
end for

- Originally proposed by Hammond[4] with $\gamma_t = 1/(t+1)$.
- One can define FW extension with **away** step.
- Crucial for our linear convergence results.

[5] J. Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. MIT, 1984.

SP-FW

Algorithm Saddle point FW algorithm

1: **for** $t = 0 \dots T$ **do**
2: Compute $\boldsymbol{r}^{(t)} := \begin{pmatrix} \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \\ -\nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \end{pmatrix}$
3: Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{z} \in \mathcal{X} \times \mathcal{Y}}{\operatorname{argmin}} \left\langle \boldsymbol{z}, \boldsymbol{r}^{(t)} \right\rangle$
4: Compute $g_t := \left\langle \boldsymbol{z}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
5: **if** $g_t \leq \epsilon$ **then return** $\boldsymbol{z}^{(t)}$
6: Let $\gamma = \min\left(1, \frac{\nu}{C} g_t\right)$ **or** $\gamma = \frac{2}{2+t}$
7: Update $\boldsymbol{z}^{(t+1)} := (1-\gamma)\boldsymbol{z}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
8: **end for**

- $\gamma_t = \frac{1}{1+t} \Rightarrow \boldsymbol{z}^{(t)} = \frac{1}{t} \sum_{i=0}^{t} \boldsymbol{s}^{(i)}$.

- Originally proposed by Hammond[4] with $\gamma_t = 1/(t+1)$.
- One can define FW extension with **away** step.
- Crucial for our linear convergence results.

[5] J. Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. MIT, 1984.

SP-FW

Algorithm Saddle point FW algorithm

1: **for** $t = 0 \dots T$ **do**
2: Compute $\boldsymbol{r}^{(t)} := \begin{pmatrix} \nabla_x \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \\ -\nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) \end{pmatrix}$
3: Compute $\boldsymbol{s}^{(t)} \in \underset{\boldsymbol{z} \in \mathcal{X} \times \mathcal{Y}}{\operatorname{argmin}} \left\langle \boldsymbol{z}, \boldsymbol{r}^{(t)} \right\rangle$
4: Compute $g_t := \left\langle \boldsymbol{z}^{(t)} - \boldsymbol{s}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle$
5: **if** $g_t \leq \epsilon$ **then return** $\boldsymbol{z}^{(t)}$
6: Let $\gamma = \min\left(1, \frac{\nu}{C} g_t\right)$ **or** $\gamma = \frac{2}{2+t}$
7: Update $\boldsymbol{z}^{(t+1)} := (1-\gamma)\boldsymbol{z}^{(t)} + \gamma \boldsymbol{s}^{(t)}$
8: **end for**

- Originally proposed by Hammond[4] with $\gamma_t = 1/(t+1)$.
- One can define FW extension with **away** step.
- Crucial for our linear convergence results.

- $\gamma_t = \frac{1}{1+t} \Rightarrow \boldsymbol{z}^{(t)} = \frac{1}{t} \sum_{i=0}^{t} \boldsymbol{s}^{(i)}$.
- $(\gamma_t = \frac{1}{1+t})$ + Bilinear objective $\leftrightarrow$ *fictitious play algorithm.*[5]

[5] J. Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. MIT, 1984.

Advantages of SP-FW

Same main property as FW:

$\boxed{\text{Only LMO (linear minimization oracle).}}$

Advantages of SP-FW

Same main property as FW:

$\boxed{\text{Only LMO (linear minimization oracle).}}$

Same other **advantages** as FW:

- Convergence certificate g_t for free.

Advantages of SP-FW

Same main property as FW:

$\boxed{\text{Only LMO (linear minimization oracle).}}$

Same other **advantages** as FW:

- Convergence certificate g_t for free.
- Affine invariance of the algorithm.

Advantages of SP-FW

Same main property as FW:

$$\boxed{\text{Only LMO (linear minimization oracle).}}$$

Same other **advantages** as FW:

- Convergence certificate g_t for free.
- Affine invariance of the algorithm.
- ***Sparsity*** of the iterates.

Advantages of SP-FW

Same main property as FW:

$$\boxed{\text{Only LMO (linear minimization oracle).}}$$

Same other **advantages** as FW:

- Convergence certificate g_t for free.
- Affine invariance of the algorithm.
- ***Sparsity*** of the iterates.
- Universal step size $\gamma_t := \frac{2}{2+t}$, adaptive step size $\gamma_t := \frac{\nu}{C} g_t$.

Advantages of SP-FW

Same main property as FW:

$$\boxed{\text{Only LMO (linear minimization oracle).}}$$

Same other **advantages** as FW:

- Convergence certificate g_t for free.
- Affine invariance of the algorithm.
- ***Sparsity*** of the iterates.
- Universal step size $\gamma_t := \frac{2}{2+t}$, adaptive step size $\gamma_t := \frac{\nu}{C} g_t$.

Main **difference** with FW:

- **No** line-search.

Advantages of SP-FW

Same main property as FW:

$$\boxed{\text{Only LMO (linear minimization oracle).}}$$

Same other **advantages** as FW:

- Convergence certificate g_t for free.
- Affine invariance of the algorithm.
- ***Sparsity*** of the iterates.
- Universal step size $\gamma_t := \frac{2}{2+t}$, adaptive step size $\gamma_t := \frac{\nu}{C} g_t$.

Main **difference** with FW:

- **No** line-search.

When constraint set is a "complicated" ***structured*** polytope:
projection is ***difficult*** whereas LMO is ***tractable***.

Hypothesis

Similar hypothesis as AFW:

- $\mathcal{L}$ is L-**smooth** and μ-**strongly convex-concave**.
- $\mathcal{X}$ and $\mathcal{Y}$ **polytopes**.

Hypothesis

Similar hypothesis as AFW:

- $\mathcal{L}$ is L-**smooth** and μ-**strongly convex-concave**.
- $\mathcal{X}$ and $\mathcal{Y}$ **polytopes**.
- Additional assumption on **bilinearity**:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{x}^\top M \boldsymbol{y} - h(\boldsymbol{y})$$

Roughly, $\|M\|$ smaller than the strong convexity constant.

$$\boxed{\nu := \frac{1}{2} - \frac{\sqrt{2}\|M\|}{\mu}\frac{D}{\delta} > 0}$$

$D := \max\{\operatorname{diam}(\mathcal{X}), \operatorname{diam}(\mathcal{Y})\}$, $\delta := \min\{PWidth(\mathcal{X}), PWidth(\mathcal{Y})\}$

Theoretical contribution

SP extension of FW with *away step*[6]:

Linear rate with ***adaptive*** step size $\gamma_t := \frac{\nu}{LD^2} g_t$.

$$\min_{s \leq t} g_s \leq O(1) \left(1 - \nu^2 \frac{\delta^2}{D^2} \frac{\mu}{2L}\right)^{k(t)}$$

Sublinear rate with ***universal*** step size $\gamma_t := \frac{2}{2+k(t)}$.

$$\min_{s \leq t} g_s \leq O\left(\frac{1}{t}\right)$$

- $k(t)$: number of non drop steps, $\boxed{k(t) \geq t/3}$.

[6] Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. "Frank-Wolfe Algorithms for Saddle Point Problems". In: *AISTATS*. 2017.

Theoretical contribution

SP extension of FW with *away step*[6]:

Linear rate with ***adaptive*** step size $\gamma_t := \frac{\nu}{LD^2} g_t$.

$$\min_{s \leq t} g_s \leq O(1) \left(1 - \nu^2 \frac{\delta^2}{D^2} \frac{\mu}{2L}\right)^{k(t)}$$

Sublinear rate with ***universal*** step size $\gamma_t := \frac{2}{2+k(t)}$.

$$\min_{s \leq t} g_s \leq O\left(\frac{1}{t}\right)$$

- $k(t)$: number of non drop steps, $\boxed{k(t) \geq t/3}$.
- Proof use recent advances on AFW $\rightarrow$ **growth condition.**

[6] Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. "Frank-Wolfe Algorithms for Saddle Point Problems". In: *AISTATS*. 2017.

Theoretical contribution

SP extension of FW with *away step*[7]:

Linear rate with ***adaptive*** step size $\gamma_t := \frac{\nu}{LD^2} g_t$.

$$\min_{s \leq t} g_s \leq O(1) \left(1 - \nu^2 \frac{\delta^2}{D^2} \frac{\mu}{2L}\right)^{k(t)}$$

Sublinear rate with ***universal*** step size $\gamma_t := \frac{2}{2+k(t)}$.

$$\min_{s \leq t} g_s \leq O\left(\frac{1}{t}\right)$$

- $k(t)$: number of non drop steps, $\boxed{k(t) \geq t/3}$.
- Proof use recent advances on AFW $\rightarrow$ **growth condition.**
- Partially answering a **30 years old conjecture**[8].
  - strongly monotone obj with step size $\frac{1}{t+1}$ over polytope.

[7] Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. "Frank-Wolfe Algorithms for Saddle Point Problems". In: *AISTATS*. 2017.

[8] J. Hammond. "Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms". PhD thesis. MIT, 1984.

Difficulties for saddle point

Usual **descent Lemma**:

$$h_{t+1} \leq h_t - \underbrace{\gamma_t g_t}_{\geq 0} + \gamma_t^2 \frac{L\|\boldsymbol{d}^{(t)}\|^2}{2}$$

With γ_t small enough the sequence decreases.

Difficulties for saddle point

Usual **descent Lemma**:

$$h_{t+1} \leq h_t - \underbrace{\gamma_t g_t}_{\geq 0} + \gamma_t^2 \frac{L\|\boldsymbol{d}^{(t)}\|^2}{2}$$

With γ_t small enough the sequence decreases.

For saddle point problem the Lipschitz gradient property gives

$$\mathcal{L}_{t+1} - \mathcal{L}^* \leq \mathcal{L}_t - \mathcal{L}^* - \underbrace{\gamma_t \left(g_t^{(x)} - g_t^{(y)} \right)}_{\text{arbitrary sign}} + \gamma_t^2 \frac{L\|\boldsymbol{d}^{(t)}\|^2}{2}.$$

- Cannot control the oscillation of the sequence.
- Must introduce other quantities to establish convergence.

Difficulties for saddle point

Standard merit functions: *primal + dual gaps*

$$h_t := \max_{\boldsymbol{y} \in \mathcal{Y}} \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}) - \min_{\boldsymbol{x} \in \mathcal{X}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^{(t)}) \geq 0.$$

[9] Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. "Frank-Wolfe Algorithms for Saddle Point Problems". In: *AISTATS*. 2017.

Difficulties for saddle point

Standard merit functions: *primal + dual gaps*

$$h_t := \max_{\boldsymbol{y} \in \mathcal{Y}} \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}) - \min_{\boldsymbol{x} \in \mathcal{X}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^{(t)}) \geq 0.$$

Problem: $\hat{\boldsymbol{y}}^{(t)} := \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y})$ depends on t.

[9] Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. "Frank-Wolfe Algorithms for Saddle Point Problems". In: *AISTATS*. 2017.

Difficulties for saddle point

Standard merit functions: *primal + dual gaps*

$$h_t := \max_{\boldsymbol{y}\in\mathcal{Y}} \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}) - \min_{\boldsymbol{x}\in\mathcal{X}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^{(t)}) \geq 0.$$

Problem: $\hat{\boldsymbol{y}}^{(t)} := \arg\max_{\boldsymbol{y}\in\mathcal{Y}} \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y})$ depends on t.

$$w_t := \underbrace{\mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^*) - \mathcal{L}^*}_{:= w_t^{(x)}} + \underbrace{\mathcal{L}^* - \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{y}^{(t)})}_{:= w_t^{(y)}}.$$

We have,

$$0 \leq w_t \leq h_t \leq g_t$$

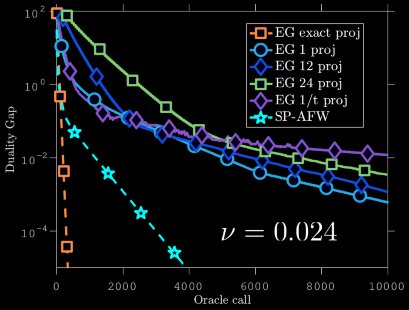
In general, w_t can be zero even if we have not reached a solution. But for strongly convex-concave function[9]

$$h_t \leq Cte\sqrt{w_t}$$

[9] Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. "Frank-Wolfe Algorithms for Saddle Point Problems". In: *AISTATS*. 2017.

Toy Experiments

- SP-AFW vs. Extragradient with approximate projection.



Theoretical step-size

$$\boxed{\gamma_t = \frac{\nu}{C} g_t.}$$

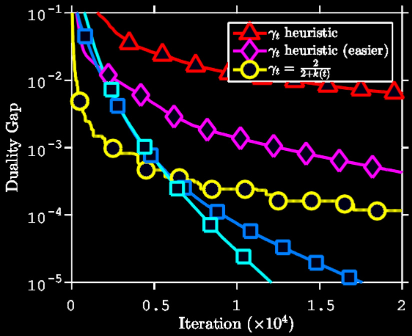
EG : [He & Harchaoui NIPS 2015]

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 + (\boldsymbol{x} - \boldsymbol{x}^*)^\top M(\boldsymbol{y} - \boldsymbol{y}^*) - \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{y}^*\|_2^2$$

- $\mathcal{X} = \mathcal{Y} := [0,1]^d$ • $d = 30$ • $C := 2LD^2$ • $L = \mu$

Toy Experiments

- SP-AFW with heuristic step-size. (When $\nu < 0$)



Heuristic step-size.

$$\boxed{\gamma_t = \frac{g_t}{C + 2\frac{\|M\|^2 D^2}{\mu}}}$$

Recall: theoretical step-size

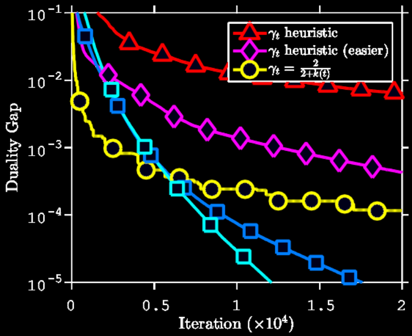
$$\boxed{\gamma_t = \frac{\nu}{C} g_t.}$$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 + (\boldsymbol{x} - \boldsymbol{x}^*)^\top M(\boldsymbol{y} - \boldsymbol{y}^*) - \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{y}^*\|_2^2$$

- $\mathcal{X} = \mathcal{Y} := [0,1]^d$
- $d = 30$
- $C := 2LD^2$
- $L = \mu$

Toy Experiments

- SP-AFW with heuristic step-size. (When $\nu < 0$)



Heuristic step-size.

$$\boxed{\gamma_t = \frac{g_t}{C + 2\frac{\|M\|^2 D^2}{\mu}}}$$

Recall: theoretical step-size

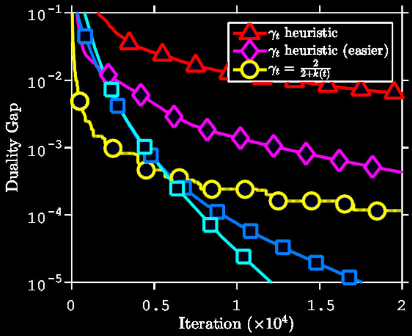
$$\boxed{\gamma_t = \frac{\nu}{C} g_t.}$$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 + (\boldsymbol{x} - \boldsymbol{x}^*)^\top M(\boldsymbol{y} - \boldsymbol{y}^*) - \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{y}^*\|_2^2$$

- $\mathcal{X} = \mathcal{Y} := [0,1]^d$ • $d = 30$ • $C := 2LD^2$ • $L = \mu$

Toy Experiments

- SP-AFW with heuristic step-size. (When $\nu < 0$)



Heuristic step-size.

$$\boxed{\gamma_t = \frac{g_t}{C + 2\frac{\|M\|^2 D^2}{\mu}}}$$

Recall: theoretical step-size

$$\boxed{\gamma_t = \frac{\nu}{C} g_t.}$$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) := \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 + (\boldsymbol{x} - \boldsymbol{x}^*)^\top M(\boldsymbol{y} - \boldsymbol{y}^*) - \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{y}^*\|_2^2$$

- $\mathcal{X} = \mathcal{Y} := [0,1]^d$
- $d = 30$
- $C := 2LD^2$
- $L = \mu$

Conclusion

- SP-FW one of the first SP solver only working with LMO.

Conclusion

- SP-FW one of the first SP solver only working with LMO.
- FW resurgence lead to new *structured* problems.

Conclusion

- SP-FW one of the first SP solver only working with LMO.
- FW resurgence lead to new *structured* problems.
- Same hope as FW for SP-FW

Conclusion

- SP-FW one of the first SP solver only working with LMO.
- FW resurgence lead to new *structured* problems.
- Same hope as FW for SP-FW ↴

$\boxed{\text{Call for applications !}}$

Conclusion

- SP-FW one of the first SP solver only working with LMO.
- FW resurgence lead to new *structured* problems.
- Same hope as FW for SP-FW $\urcorner$

$$\boxed{\text{Call for applications !}}$$

- With a bilinear objective this algorithm is ***highly related*** to the *fictitious play algorithm.*

Conclusion

- SP-FW one of the first SP solver only working with LMO.
- FW resurgence lead to new *structured* problems.
- Same hope as FW for SP-FW $\neg$

$$\boxed{\text{Call for applications !}}$$

- With a bilinear objective this algorithm is ***highly related*** to the *fictitious play algorithm.*
- Rich interplay tapping into this game theory literature.

Conclusion

- SP-FW one of the first SP solver only working with LMO.
- FW resurgence lead to new *structured* problems.
- Same hope as FW for SP-FW ↴

$\boxed{\text{Call for applications !}}$

- With a bilinear objective this algorithm is ***highly related*** to the *fictitious play algorithm.*
- Rich interplay tapping into this game theory literature.
- Still many theoretical opened questions.

  ↳ Karlin's conjecture.[10]

  ↳ Convergence without assumption on the bilinearity.

[10] Samuel Karlin. *Mathematical methods and theory in games, programming and economics.* 1960.

Thank You !

Slides available on `www.di.ens.fr/~gidel`.

Problems with difficult projection

University game:

1. Game between two universities (A and B).
2. Admitting d students and have to assign pairs of students into dorms.
3. The game has a payoff matrix M belonging to $\mathbb{R}^{(d(d-1)/2)^2}$.
4. $M_{ij,kl}$ is the expected tuition that B gets (or A gives up) if A pairs student i with j and B pairs student k with l.
5. Here the actions are both in the *marginal polytope* of all perfect *unipartite matchings*.

Hard to project on this polytope whereas the LMO can be solved efficiently with the blossom algorithm[11].

[11] J. Edmonds. "Paths, trees and flowers". In: *Canadian Journal of Mathematics* (1965).

Experiments

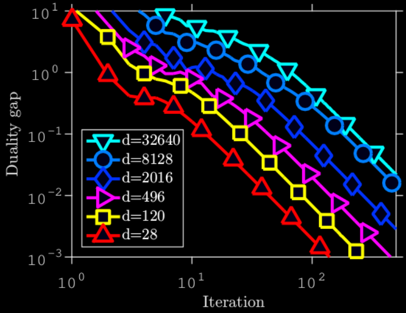


Figure: SP-FW on the University game.

- Sublinear convergence rate (faster than expected $O(t^{-2})$)

Experiments

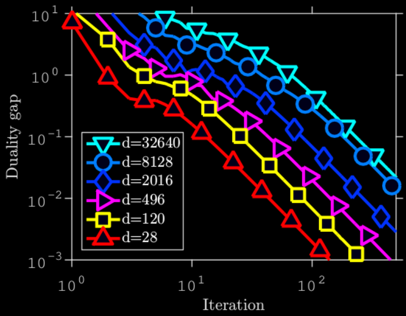


Figure: SP-FW on the University game.

- Sublinear convergence rate (faster than expected $O(t^{-2})$)
- Best theoretical rate proved: $O(t^{-1/d})$

Experiments

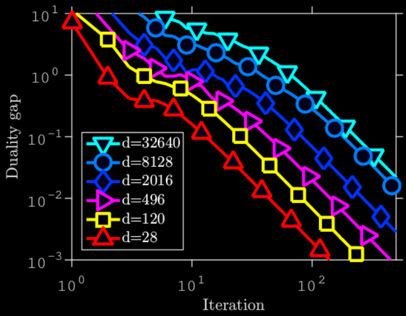


Figure: SP-FW on the University game.

- Sublinear convergence rate (faster than expected $O(t^{-2})$)
- Best theoretical rate proved: $O(t^{-1/d})$
- Scale well with dimension.