

# Extensions de l'algorithme de Frank-Wolfe pour la recherche de point-selle

Gauthier Gidel,  
École Normale Supérieure,  
INRIA Paris, Équipe SIERRA

26 Octobre 2016



# Sommaire

- ① Introduction.
- ② Motivations.
- ③ Optimisation convexe.
- ④ Optimisation de points-selles.

# Introduction

De nombreux problèmes d'optimisation ont été formulé au cours du 20<sup>ème</sup> siècle.

- Problème de transport de masse<sup>1</sup>.
- Recherche Opérationnelle<sup>2</sup>.
- Plus récemment, le débruitage d'image<sup>3</sup>.

Initialement on a cherché à résoudre des problèmes quadratiques

$$\min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{2} \mathbf{x}^\top Q \mathbf{x} + c^\top \mathbf{x} \tag{Q}$$

contraint à  $A\mathbf{x} \leq \mathbf{b}$ .

---

1. Leonid Vitalievich KANTOROVICH. “On the translocation of masses”. In : *Dokl. Akad. Nauk SSSR*. T. 37. 1942, p. 199–201.

2. George Bernard DANTZIG. *Linear programming and extensions*. Princeton university press, 1963.

3. Hidetoshi NISHIMORI. *Statistical physics of spin glasses and information processing : an introduction*. T. 111. Clarendon Press, 2001.

# Introduction

- Algorithme de Frank-Wolfe<sup>4</sup> proposé pour résoudre ce problème.
- Originellement, il proposait de résoudre (Q) à l'aide de résolutions successives de *programmes linéaires*<sup>5</sup>,

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^d} c^\top \mathbf{x} \\ \text{contraint à } \mathbf{Ax} \leq \mathbf{b}. \end{aligned} \tag{L}$$

---

4. Marguerite FRANK et Philip WOLFE. “An Algorithm for Quadratic Programming”. In : *Nav. Res. Logist. Q.* (1956).

5. George Bernard DANTZIG. *Linear programming and extensions*. Princeton university press, 1963, David G LUENBERGER. *Introduction to linear and nonlinear programming*. T. 28. Addison-Wesley Reading, MA, 1973.

# Motivation : Optimisation sous contraintes

- Données  $\mathbf{x}_1, \dots, \mathbf{x}_N$  de  $\mathbb{R}^d$ .
- Classe associée  $y_1, \dots, y_N \in \{\pm 1\}$ .

**Objectif** : déduire un classificateur linéaire prédisant la classe de  $\mathbf{x}$ ,

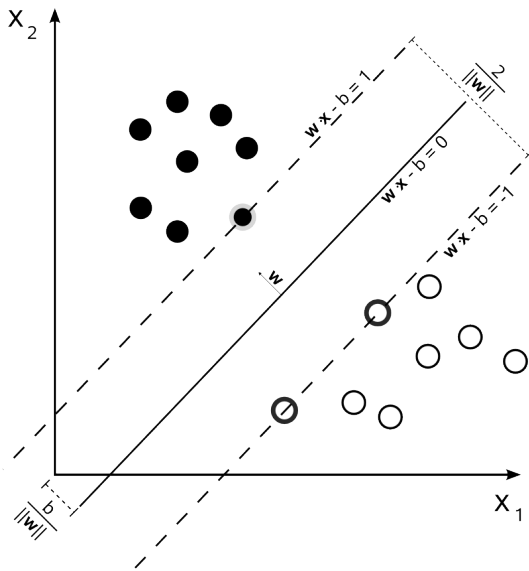
$$\tilde{y}(\mathbf{x}) = \boldsymbol{\omega}^\top \mathbf{x} - \mathbf{b} \quad (1)$$

Classe prédite :  $\text{sign}(\tilde{y}(\mathbf{x}))$

Dérivation du problème d'optimisation au tableau :

$$\begin{aligned} \arg \min_{\boldsymbol{\omega}, \mathbf{b}} \frac{1}{2} \|\boldsymbol{\omega}\|^2 \\ \mathbf{y}(\mathbf{x}_n) \left( \boldsymbol{\omega}^\top \mathbf{x}_n - \mathbf{b} \right) \geq 1 \quad \forall n \in \{1, \dots, N\}. \end{aligned} \quad (2)$$

# Motivation : Support Vector Machine



# Oracles

Les algorithmes proposés doivent être utilisables en pratique :

- Impossible de connaître la fonction globalement.
- Seul l'accès à certaines propriétés locales est possible.
- Concept d'*oracle local*<sup>6</sup> résumant cela.
- Brique de base la plus coûteuse de l'algorithme

## Définition (Oracle d'ordre 0,1)

Soit  $f : \mathcal{X} \rightarrow \mathbb{R}$  une fonction.

- Un oracle d'ordre zéro prend comme entrée un point  $x \in \mathcal{X}$  et renvoie la valeur de  $f$  au point  $x$ .
- Un oracle d'ordre un prend en entrée un point  $x \in \mathcal{X}$  et renvoie un sous gradient de  $f$ .

---

6. Sébastien BUBECK. “Convex optimization : Algorithms and complexity”. In : *arXiv preprint arXiv :1405.4980* (2014).

# Définitions

## Définition (Ensemble convexe)

Soit  $\mathcal{X} \subset \mathbb{R}^d$ .  $\mathcal{X}$  est convexe si pour tout  $\mathbf{x}, \mathbf{y} \in \mathcal{X}$  et  $\lambda \in [0, 1]$ ,

$$\lambda \mathbf{x} + (1 - \lambda) \mathbf{y} \in \mathcal{X}. \quad (3)$$

## Définition (Fonction convexe)

Soit  $f : \mathcal{X} \subset \mathbb{R}^d \rightarrow \mathbb{R}$  où  $\mathcal{X}$  est convexe. On dit que  $f$  est une fonction convexe si pour tout  $\mathbf{x}, \mathbf{y} \in \mathcal{X}$  et  $\lambda \in [0, 1]$ ,

$$f(\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}). \quad (4)$$

On dit que  $f$  est une fonction  $\beta$ -fortement convexe si pour tout  $\mathbf{x}, \mathbf{y} \in \mathcal{X}$  et  $\lambda \in [0, 1]$ ,

$$f(\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}) - \frac{\beta}{2} \lambda (1 - \lambda) \|\mathbf{x} - \mathbf{y}\|_2^2. \quad (5)$$



# Optimisation convexe sous contraintes

Soit  $\mathcal{X}$  un convexe de  $\mathbb{R}^d$  on cherche à résoudre le problème,

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \quad \text{avec uniquement un accès à un oracle } \mathcal{O}_f(\cdot). \quad (6)$$

- Si un minimum appartient à l'intérieur de  $\mathcal{X}$  alors les contraintes sont inutiles et

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) = \min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) \quad (7)$$

- Intéressant lorsque  $\mathbf{x}^*$  appartient au bord du domaine  $\mathcal{X}$ .
- La contrainte change la valeur de l'optimum.

# Descente de gradient projeté : l'algorithme

Au point  $\mathbf{x}$  de l'ensemble  $\mathcal{X}$ , un sous-gradient  $\mathbf{g}$  de la fonction  $f$  vérifie pour tout  $\mathbf{y} \in \mathcal{X}$ ,

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{y} - \mathbf{x}, \mathbf{g} \rangle. \quad (8)$$

On note  $\partial f(\mathbf{x})$  l'ensemble des sous-gradient de  $f$  au point  $\mathbf{x}$ .

---

**Algorithm 1** Descente de gradient projetée

---

```
1: Soit  $\mathbf{x}^{(0)} \in \mathcal{X}$ 
2: for  $t = 0 \dots T$  do
3:    $\mathbf{r}^{(t)} \in \partial f(\mathbf{x}^{(t)})$ 
4:    $\mathbf{x}^{(t+1)} := P_{\mathcal{X}}(\mathbf{x}^{(t)} - \eta \mathbf{r}_t)$ 
5: end for
```

---

- Approche standard dans la littérature.
- La direction d'un sous-gradient optimise localement la descente mais peut faire sortir l'itérée  $\mathbf{x}^{(t)}$  du domaine  $\mathcal{X}$ .
- Projection sur  $\mathcal{X}$  après chaque descente de gradient.

# Descente de gradient projeté : vitesse de convergence

L'algorithme présenté dépend cruciallement de la valeur du pas  $\eta$ . Il ne converge que si  $\eta$  est assez petit,

## Théorème

Soit  $f$  une fonction sous-différentiable et  $\mathcal{X}$  un compact convexe. Pour tout entier  $T$  et tout  $\eta > 0$  les points calculés par l'Algorithme 1 vérifient,

$$\min_{0 \leq t \leq T} f(\mathbf{x}^{(t)}) - f^* \leq \frac{\|\mathbf{x}_0 - \mathbf{x}\|_2^2}{2\eta T} + \frac{\eta C^2}{2} \quad (9)$$

en particulier pour  $\eta = \frac{\|\mathbf{x}_0 - \mathbf{x}\|_2}{C\sqrt{T}}$ ,

$$\min_{0 \leq t \leq T} f(\mathbf{x}^{(t)}) - f^* \leq \frac{\|\mathbf{x}_0 - \mathbf{x}\|_2}{C\sqrt{T}}. \quad (10)$$

**Problème** : l'oracle utilisé ici comprend une étape de projection sur un convexe. Cette projection peut être très coûteuse.

# Optimisation convexe sous contraintes : l'oracle de Frank-Wolfe

Oracle utilisé pour la projection :

$$\mathcal{O}_P(\mathbf{x}) \in \arg \min_{\mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|^2. \quad (11)$$

- Il est de la forme (Q).
- Oracle quadratique parfois très coûteux.
- Surtout lorsque le domaine  $\mathcal{X}$  possède une structure complexe.

Oracle de Frank-Wolfe est beaucoup plus simple à calculer car c'est un oracle linéaire :

$$\mathcal{O}_{FW}(\mathbf{g}) \in \arg \min_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{x}, \mathbf{g} \rangle \quad (12)$$

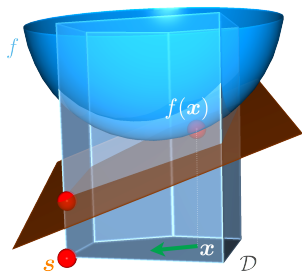
# Optimisation convexe sous contraintes : l'algorithme de Frank-Wolfe

---

**Algorithm 2** Algorithme de Frank-Wolfe

---

- 1: Soit  $\mathbf{x}^{(0)} \in \mathcal{X}$
  - 2: **for**  $t = 0 \dots T$  **do**
  - 3:  $\mathbf{s}^{(t)} := \operatorname{argmin}_{\mathbf{s} \in \mathcal{X}} \langle \mathbf{s}, \nabla f(\mathbf{x}^{(t)}) \rangle$ .
  - 4:  $g_t := \langle \mathbf{x}^{(t)} - \mathbf{s}^{(t)}, \mathbf{r}^{(t)} \rangle$
  - 5: **if**  $g_t \leq \epsilon$  **then return**  $\mathbf{x}^{(t)}$
  - 6:  $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \gamma_t(\mathbf{s}^{(t)} - \mathbf{x}^{(t)})$
  - 7: **end for**
- 



**FIGURE** – One step of the FW algorithm

Le pas  $\gamma_t$  est souvent égal à  $\frac{2}{2+t}$  ou alors il est choisi tel que

$$\gamma_t := \operatorname{argmin}_{\gamma \in [0,1]} f(\mathbf{x}^{(t)} + \gamma(\mathbf{s}^{(t)} - \mathbf{x}^{(t)})) \quad (13)$$

# Optimisation convexe sous contraintes : l'algorithme de Frank-Wolfe

## Théorème

Soit  $f$  une fonction convexe dont le gradient est  $L$ -Lipschitz et  $\mathcal{X}$  un compact convexe de diamètre  $D_{\mathcal{X}}$ . Pour tout  $t \geq 1$ , les itérés de l'Algorithme 2 appliqués à la minimisation de  $f$  dans  $\mathcal{X}$  vérifient,

$$f(\mathbf{x}^{(t)}) - f(\mathbf{x}^*) \leq \frac{2LD_{\mathcal{X}}^2}{t+2}, \quad (14)$$

où  $\mathbf{x}^* \in \mathcal{X}$  est un point minimisant  $f$  sur  $\mathcal{X}$ .

Éléments de preuve au tableau <sup>7</sup>.

**NB** : Descente de gradient en  $O(\frac{1}{t})$  avec des hypothèses similaires.

---

7. [Martin JAGGI](#). "Revisiting Frank-Wolfe : Projection-free sparse convex optimization". In : *ICML, 2013*.

## Optimisation de point selle : motivation

**Jeux à somme nulle** : Jeu à deux joueurs (J1 et J2) :

- J1 choisit une action  $i \in \{1, \dots, I\}$  et J2 choisit  $j \in \{1, \dots, J\}$ .
- La perte de J1 suite à ces deux choix est  $a_{ij}$ , celui de J2 est  $-a_{ij}$ .
- Chaque joueur joue selon une distribution de probabilité sur les actions (respectivement  $\mathbf{x}$  et  $\mathbf{y}$  pour les joueurs 1 et 2).
- Ainsi l'espérance du profit du joueur 1 est

$$\mathbb{E}[a_{ij}] = \mathbf{x}^\top A \mathbf{y}. \quad (15)$$

- But : minimiser sa perte. Ainsi le problème pour le joueur 1 est

$$\min_{\mathbf{x} \in \Delta_I} \max_{\mathbf{y} \in \Delta_J} \mathbf{x}^\top A \mathbf{y}. \quad (16)$$

$$\text{où } \Delta_I := \left\{ \mathbf{x} \in \mathbb{R}^I \mid \sum_{i=1}^I x_i = 1, \mathbf{x} \geq 0 \right\}.$$

# Optimisation de point selle : motivation

## Apprentissage robuste.

Modèle classique de minimisation de la moyenne des pertes empiriques plus une régularisation,

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{k=1}^n \ell(f(\mathbf{x}_k), y_k) + \lambda \Omega(f), \quad (17)$$

**Hypothèse** : il existe une incertitude concernant les données fournies.

- Meilleures prédictions seraient faites sur les données de test en résolvant le problème,

$$\min_{f \in \mathcal{F}} \max_{\substack{\tilde{\mathbf{x}}_i \in B_\delta(\mathbf{x}_i) \\ i \in \{1, \dots, n\}}} \frac{1}{n} \sum_{k=1}^n \ell(f(\tilde{\mathbf{x}}_k), y_k) + \lambda \Omega(f) \quad (18)$$

$$\text{où } B_\delta(\mathbf{x}) := \{\mathbf{y} \mid \|\mathbf{y} - \mathbf{x}\| \leq \delta\}.$$

- Problème de point selle.



# Définition d'un point selle

## Définition (Point selle)

Soit  $\mathcal{X}$  et  $\mathcal{Y}$  deux ensembles non vides et  $\mathcal{L} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  une fonction. Un couple  $(\mathbf{x}^*, \mathbf{y}^*)$  est appelé *point selle* si

$$\mathcal{L}(\mathbf{x}^*, \mathbf{y}) \leq \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*) \leq \mathcal{L}(\mathbf{x}, \mathbf{y}^*) \quad \forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{y} \in \mathcal{Y}. \quad (19)$$

Cadre plus restreint :

- Pour tout  $\mathbf{y} \in \mathcal{Y}$   $\mathcal{L}(\cdot, \mathbf{y})$  convexe.
- Pour tout  $\mathbf{x} \in \mathcal{X}$   $\mathcal{L}(\mathbf{x}, \cdot)$  concave
- $\mathcal{X}$  et  $\mathcal{Y}$  compacts convexes.

Solutions de (19) coïncident avec les points optimaux :

$$\min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{y} \in \mathcal{Y}} \mathcal{L}(\mathbf{x}, \mathbf{y}) = \max_{\mathbf{y} \in \mathcal{Y}} \min_{\mathbf{x} \in \mathcal{X}} \mathcal{L}(\mathbf{x}, \mathbf{y}) \quad (20)$$

# Approche standard dans la littérature

Algorithmes standards pour résoudre un problème de point selle,

- **Sous gradient projeté.**

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

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

La *moyenne* des  $(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})$  converge vers un point selle.

- **Extra-sous-gradient projeté.**

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

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

**Intuition :** *Anticiper* le comportement de l'adversaire en effectuant un pas en avant pour voir quelle sera son action future.

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

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

# Extension de l'algorithme de Frank-Wolfe

## Idée :

- Étendre l'algorithme de Frank-Wolfe aux points-selles de la même manière que la descente de gradient projetée.
- Le vecteur  $\mathbf{r}^{(t)}$  qui était égal au gradient est remplacé par la quantité ligne 3 de l'Algorithme 3.
- Ligne 7 les constantes  $\nu$  et  $C_{\mathcal{L}}$  sont des constantes dépendant de  $\mathcal{L}$ ,  $\mathcal{X}$  et  $\mathcal{Y}$ .

---

## Algorithm 3 FW pour points selles (SP-FW)

---

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

---

## Avantages de SP-FW

- Il ne nécessite qu'un oracle linéaire.
- $g_t$  majore l'erreur  $f(\mathbf{x}^{(t)}) - f^*$  et est donc un critère d'arrêt.
- *Parcimonie* de la solution calculée,

$$\mathbf{x}^{(t)} = \sum_{\mathbf{s} \in \mathcal{S}^{(t)}} \alpha(\mathbf{s}) \mathbf{s}$$

avec  $\sum_{\mathbf{s} \in \mathcal{S}^{(t)}} \alpha(\mathbf{s}) = 1$ ,  $\mathcal{S}^{(t+1)} = \{\mathbf{s}^{(t)}\} \cup \mathcal{S}^{(t)}$  et  $\mathcal{S}^{(0)} = \emptyset$ . (21)

- Simple à programmer.
- Invariant par transformations affines.

## Théorème

Soit  $\mathcal{L}$  une fonction convexe-concave et  $\mathcal{X} \times \mathcal{Y}$  un compact convexe. Supposons que le gradient de  $\mathcal{L}$  est Lipschitz, que  $\mathbf{x} \mapsto \mathcal{L}(\mathbf{x}, \mathbf{y}) - \mu \|\mathbf{x}\|_2^2 + \mu \|\mathbf{y}\|_2^2$  est convexe-concave et que nous sommes dans l'un des deux cas suivant :

Le point selle appartient à l'intérieur de  $\mathcal{X} \times \mathcal{Y}$ . (I)

Les ensembles  $\mathcal{X}$  et  $\mathcal{Y}$  sont des polytopes. (P)

Si  $\mu$  est assez grande devant les dérivées croisées de  $\mathcal{L}$ , alors SP-AFW converge géométriquement et

$$\min_{s \leq t} g_s = O\left((1 - \rho)^{t/2}\right) \text{ pour (I),} \quad \min_{s \leq t} g_s = O\left((1 - \rho)^{t/3}\right) \text{ pour (P)}$$

où  $\rho := \frac{\nu\mu}{2C}$ . La constante  $\nu \in [-\infty, 1]$  représente à quel point  $\mu$  est suffisamment grand devant la norme des dérivées croisées de  $\nabla_y \nabla_x \mathcal{L}$  et  $\nabla_y \nabla_x \mathcal{L}$ . Notre preuve de convergence se restreint au cas où  $\nu \in ]0, 1]$ .

# Contributions

Trois contributions théoriques :

- 1 Introduction d'une nouvelle variante de SP-FW. Cette variante de l'algorithme (SP-AFW) peut diminuer le poids de certain atomes dans (21).
- 2 Preuve de la convergence de SP-FW en remplaçant l'hypothèse sur la forte convexité de la fonction en une hypothèse sur la forte convexité des ensemble  $\mathcal{X}$  et  $\mathcal{Y}$ .
- 3 Extension du lien fait par Hammond entre le *fictitious play algorithm* et SP-FW.

# Expériences

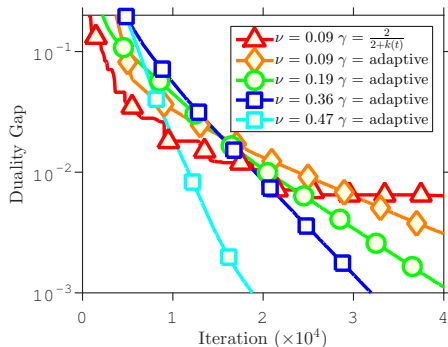


FIGURE – SP-AFW sur un exemple quadratique  $d = 30$

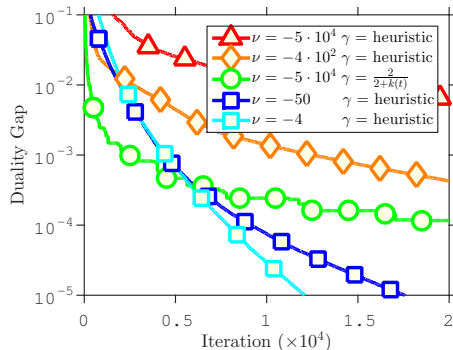


FIGURE – SP-AFW sur un exemple quadratique  $d = 30$  avec un pas heuristique

# Expériences

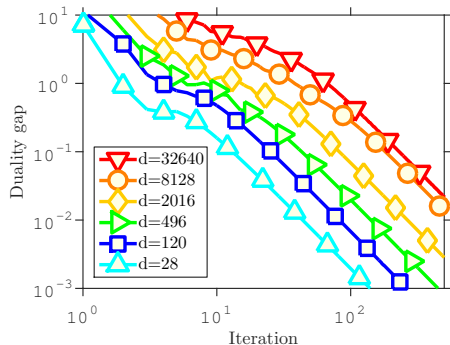


FIGURE – SP-FW sur un jeux à somme nulle.

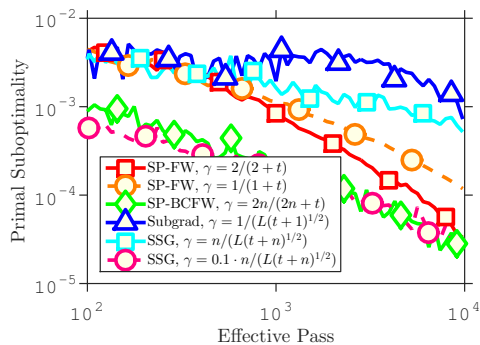


FIGURE – SP-FW sur un problème de SVM structuré



# Problèmes ouverts

## La conjecture de Hammond.

- Dans sa thèse Hammond <sup>8</sup> a conjecturé la convergence de l'algorithme de Frank-Wolfe pour les points selles.
- Réponse partielle à cette conjecture dans notre Théorème 7.
- Reste néanmoins ouverte lorsque  $\nu \leq 0$ .

## La conjecture de Karlin <sup>9</sup>.

- Relativement ancienne, elle provient de la théorie des jeux.
- Affirme que le *fictitious play algorithm* converge en  $O(1/\sqrt{t})$ .
- Il sert à la recherche d'équilibre de jeux à deux joueurs.
- Nous avons démontré des liens étroits entre le *fictitious play algorithm* et l'algorithme de Frank-Wolfe.

---

8. [Janice H HAMMOND](#). “Solving asymmetric variational inequality problems and systems of equations with generalized nonlinear programming algorithms”. [Thèse de doct. Massachusetts Institute of Technology, 1984](#).

9. [Samuel KARLIN](#). *Mathematical methods and theory in games, programming and economics*. 1960.

# Conclusion

- Les problèmes de points selles modélisent des problématiques nouvelles sont d'un grand intérêt dans de nombreux domaines nécessitant de l'optimisation.
- L'extension proposée possède les mêmes propriétés avantageuses que sont homologue convexe.
- Preuve de convergence de ce nouvel algorithme dans certain cas particuliers.
- Travail expérimental approfondi afin de vérifier nos affirmations théoriques et vérifier les conjectures encore en suspend.

Merci pour votre attention.

Des Questions ?