D-FW: Communication Efficient Distributed Algorithms for High-dimensional Sparse Optimization

Jean Lafond[†], **Hoi-To Wai**[‡] and Eric Moulines[#]

[†]Institut Mines-Telecom, Telecom ParisTech, CNRS LTCI, Paris, France.
[‡]School of ECEE, Arizona State University, USA. [#]CMAP, Ecole Polytechnique, Palaiseau, France.

Acknowledgement: Direction Générale de l'Armement and the labex LMH (ANR-11-LABX-0056-LMH), NSF CCF-1011811.

[⋆]Equal contribution from J. Lafond and H.-T. Wai.

High-dimensional, distributed sparse optimization

What do we need?

- Lots of data scattered around in the network $\Longrightarrow$ need *scalable and distributed algorithms*

Problem of Interest

- Consider optimization problems of the form:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^n} \; F(\boldsymbol{\theta}) := \frac{1}{T} \sum_{s=1}^{T} f_s(\boldsymbol{\theta}) \;\; \text{s.t.} \;\; \|\boldsymbol{\theta}\|_1 \leq r \tag{1}$$

  - $f_s : \mathbb{R}^n \to \mathbb{R}$ – strongly convex, continuously differentiable objective fct. of agent s.
  - T – number of *agents* cooperating, moderately sized, $T \approx 10$ to 100.
  - n – dimension of parameter to be estimated, $n \approx 10^4$ to $10^6 \gg 0$.
  - Optimal solution to (1) is *sparse*, $\|\boldsymbol{\theta}^\star\|_0 \ll n$.
- *Applications*: sparse recovery, high-dimensional regression, etc.

This work:

- distributed, computation & *communication efficient* algorithms for (1).
- convergence rate analysis of the proposed algorithms.

Prior Work

- Focuses on improving the scalability, e.g., distributed proximal/projected gradient (D-PG) [RNV10, RNV12]. Let $t \in \mathbb{N}$ be the iteration number, the sth agent does:

$$\boldsymbol{\theta}_{t+1}^{s} = \mathcal{P}_{\mathcal{C}}\big(\underbrace{\sum_{s'=1}^{T} W_{ss'} \boldsymbol{\theta}_t^{s'} - \alpha_t \nabla f_s \big(\sum_{s'=1}^{T} W_{ss'} \boldsymbol{\theta}_t^{s'} \big)}_{\text{in-network parameter exchange}} \big), \tag{2}$$

  - While $\boldsymbol{\theta}^{\star}$ is sparse, intermediate iterates $\boldsymbol{\theta}_t^{s}$ in D-PG is *not sparse!*
  - Per-iteration communication cost for D-PG (and its variants) is high.
- Related works for different types of problems [JST$^+$14, BLG$^+$14].

Agenda

1 Frank-Wolfe algorithm
   Recent results on stochastic FW

2 Distributed FW algorithms for sparse optimization
   DistFW algorithm for star networks
   DeFW algorithm for general networks
   Convergence Analysis

3 Numerical Experiment

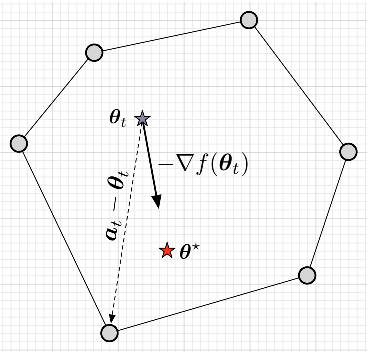
4 Conclusions & Future Work

Frank-Wolfe (FW) algorithm

(a.k.a. conditional gradient, projection-free optimization, etc.)

- A classical, first order algorithm with recent interests [FW56].
- Applications in machine learning and solving high-dimensional problems, e.g., matrix completion, sparse optimization [Jag13].
- Believed to be slow with sublinear convergence $\mathcal{O}(1/t)$ [CC68].
- Recent results demonstrated cases where linear convergence rate $\mathcal{O}((1-\rho)^t)$ can be achieved [LJJ13].
- Analysis of its stochastic variants [LWM15, LZ14].

Suppose that $\mathcal{C}$ is a polytope, $\mathcal{C} = \mathrm{conv}\{\boldsymbol{a}^1, \boldsymbol{a}^2, ..., \boldsymbol{a}^d\}$.



Frank-Wolfe Algorithm [FW56]:

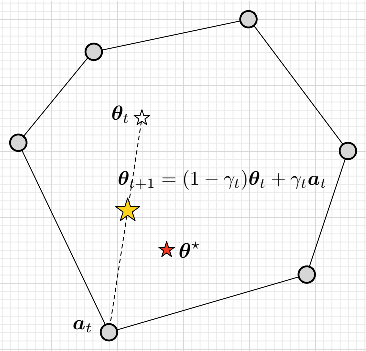
1. For iteration $t = 0, 1, 2, ...$
2. Linear optimization (LO):
   $\boldsymbol{a}_t \leftarrow \arg\min_{\boldsymbol{a} \in \mathcal{C}} \langle \nabla F(\boldsymbol{\theta}_t), \boldsymbol{a} - \boldsymbol{\theta}_t \rangle$.
3. Update the iterate:
   $\boldsymbol{\theta}_{t+1} \leftarrow (1-\gamma_t)\boldsymbol{\theta}_t + \gamma_t \boldsymbol{a}_t$, where $\gamma_t = 2/(t+2)$.
4. Repeat Step 2 to 3.

Convergence of FW algorithm [FW56]

If $F(\boldsymbol{\theta})$ is convex and continuously differentiable, then

$$F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^\star) = \mathcal{O}(1/t). \quad (3)$$

Suppose that $\mathcal{C}$ is a polytope, $\mathcal{C} = \mathrm{conv}\{\boldsymbol{a}^1, \boldsymbol{a}^2, ..., \boldsymbol{a}^d\}$.



Frank-Wolfe Algorithm [FW56]:

1. For iteration $t = 0, 1, 2, ...$
2. Linear optimization (LO): $\boldsymbol{a}_t \leftarrow \arg\min_{\boldsymbol{a} \in \mathcal{C}} \langle \nabla F(\boldsymbol{\theta}_t), \boldsymbol{a} - \boldsymbol{\theta}_t \rangle$.
3. Update the iterate: $\boldsymbol{\theta}_{t+1} \leftarrow (1-\gamma_t)\boldsymbol{\theta}_t + \gamma_t \boldsymbol{a}_t$, where $\gamma_t = 2/(t+2)$.
4. Repeat Step 2 to 3.

Convergence of FW algorithm [FW56]

If $F(\boldsymbol{\theta})$ is convex and continuously differentiable, then

$$F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^\star) = \mathcal{O}(1/t). \tag{3}$$

Case of stochastic gradient – stochastic FW

- Suppose that an *inexact/stochastic* gradient $\hat{\nabla}_t F(\boldsymbol{\theta}_t)$ is used in the LO in lieu of $\nabla F(\boldsymbol{\theta}_t) \Longrightarrow$ stochastic FW (sFW) algorithm.
- **Assumption**: with high probability (w.h.p.) the following holds,

$$\|\hat{\nabla}_t F(\boldsymbol{\theta}_t) - \nabla F(\boldsymbol{\theta}_t)\|_\infty = \mathcal{O}(\sqrt{1/t}),\ \forall\ t \geq 1, \tag{H1}$$

Convergence of sFW algorithm [LWM15]

Under (H1), we have w.h.p. $F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^\star) = \mathcal{O}(\sqrt{1/t})$. Furthermore, if F is strongly convex and $\boldsymbol{\theta}^\star \in \mathrm{int}(\mathcal{C})$, we have w.h.p.

$$F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^\star) = \mathcal{O}(1/t). \tag{4}$$

Linear Optimization Oracle

- In the case of ℓ_1 ball, $\mathcal{C} = \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_1 \leq r\}$, we have

$$\boldsymbol{a}_t = -r \cdot \mathrm{sign}([\nabla F(\boldsymbol{\theta}_t)]_{i_t}) \cdot \boldsymbol{e}_{i_t}, \tag{5}$$

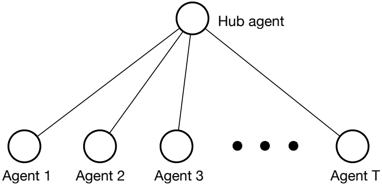
where $i_t = \arg\max_{j \in [n]} |[\nabla F(\boldsymbol{\theta}_t)]_j|$.

Properties —

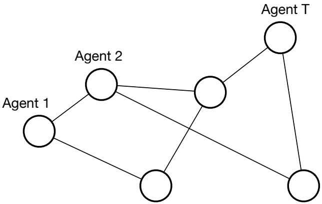
1. The update performed at iteration t, $\boldsymbol{a}_t$, is **1-sparse**!
2. Finding $\boldsymbol{a}_t$ needs only **maximum magnitude coordinate** in $\nabla F(\boldsymbol{\theta}_t)$ and the corresponding **sign**.

Distributed FW algorithms

- **Main idea**: to mimic the FW (or sFW) algorithm via in-network computations.
- We propose two schemes for different network topologies:



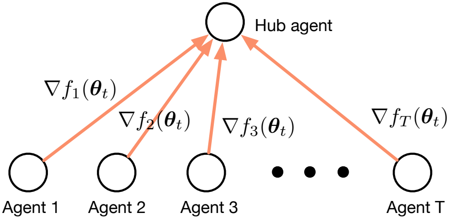
Distributed FW (DistFW)



Decentralized FW (DeFW)

Distributed FW (DistFW) algorithm

- **Setting**: $\exists$ *hub agent* all T agents can communicate with.



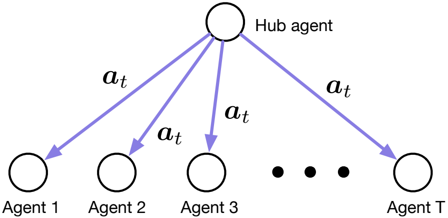
- **Aggregating phase:** the hub agent computes $\hat{\nabla}_t F(\boldsymbol{\theta}_t)$ by:

$$\hat{\nabla}_t F(\boldsymbol{\theta}_t) = (1/T) \sum_{s=1}^{T} \nabla f_s(\boldsymbol{\theta}_t). \quad (6)$$

- **Broadcasting phase:** based on $\hat{\nabla}_t F(\theta_t)$, the hub agent computes $\boldsymbol{a}_t$ from (5) and broadcast $\boldsymbol{a}_t$ to agents. The agents perform the individual updates by $\theta_{t+1} = (1 - \gamma_t)\theta_t + \gamma_t \boldsymbol{a}_t$.

Distributed FW (DistFW) algorithm

- **Setting**: $\exists$ *hub agent* all T agents can communicate with.



- **Aggregating phase:** the hub agent computes $\hat{\nabla}_t F(\boldsymbol{\theta}_t)$ by:

$$\hat{\nabla}_t F(\boldsymbol{\theta}_t) = (1/T) \sum_{s=1}^{T} \nabla f_s(\boldsymbol{\theta}_t). \tag{6}$$

- **Broadcasting phase:** based on $\hat{\nabla}_t F(\boldsymbol{\theta}_t)$, the hub agent computes $\boldsymbol{a}_t$ from (5) and broadcast $\boldsymbol{a}_t$ to agents. The agents perform the individual updates by $\boldsymbol{\theta}_{t+1} = (1 - \gamma_t)\boldsymbol{\theta}_t + \gamma_t \boldsymbol{a}_t$.

Communication efficiencies

- ✗ **Aggregating**: requires $\nabla f_s(\boldsymbol{\theta}_t)$ from the agents, maybe dense.
- ✓ **Broadcasting**: involves $\boldsymbol{a}_t$ that is only 1-sparse.
- ✓ *Our remedy*: agent s "sparsifies" its own $\nabla f_s(\boldsymbol{\theta}_t)$ to a p_t-sparse ($p_t \ll n$) vector before communicating:
  - **Random Coordinate Selection** — Agent s selects the coordinate $i \in [n] := \{1, ..., n\}$ with probability p_t/n.
  - **Extremal Coordinate Selection** — Agent s sorts $\nabla f_s(\boldsymbol{\theta}_t)$ and selects $p_t/2$ coordinates that correspond to the max. and min. elements in the vector.
  - Recall: the LO oracle only cares about the max. magnitude elements in $\nabla F(\boldsymbol{\theta}_t)$.

Communication efficiencies

- ✗ **Aggregating**: requires $\nabla f_s(\boldsymbol{\theta}_t)$ from the agents, maybe dense.
- ✓ **Broadcasting**: involves $\boldsymbol{a}_t$ that is only 1-sparse.
- ✓ *Our remedy*: agent s "sparsifies" its own $\nabla f_s(\boldsymbol{\theta}_t)$ to a p_t-sparse ($p_t \ll n$) vector before communicating:
  - **Random Coordinate Selection** — Agent s selects the coordinate $i \in [n] := \{1, \ldots, n\}$ with probability p_t / n.
  - **Extremal Coordinate Selection** — Agent s sorts $\nabla f_s(\boldsymbol{\theta}_t)$ and selects $p_t/2$ coordinates that correspond to the max. and min. elements in the vector.

  - Recall: the LO oracle only cares about the **max. magnitude elements** in $\nabla F(\boldsymbol{\theta}_t)$.

Decentralized FW (DeFW) algorithm

- **Setting**: agents are connected via a graph $G = (V, E)$.
- Let $\bar{\boldsymbol{\theta}}_t := (1/T) \sum_{s=1}^{T} \boldsymbol{\theta}_t^s$. Our challenges are:
  - *Aggregating* – computing $\hat{\nabla}_t F(\bar{\boldsymbol{\theta}}_t) \approx \nabla F(\bar{\boldsymbol{\theta}}_t) = (1/T) \sum_{s=1}^{T} \nabla f_s(\bar{\boldsymbol{\theta}}_t)$.
  - *Consensus* – the local parameters $\boldsymbol{\theta}_t^s$ should be close to $\bar{\boldsymbol{\theta}}_t$.
- *Gossip-based average consensus (G-AC) subroutine* [DKM+10] –

  input : $\{\boldsymbol{x}_{s,0}\}_{s \in [T]}$ – initial values held by the agents

  repeat for $\ell = 0, 1, ..., \ell_t$:

  $$\text{gossip upd} : \boldsymbol{x}^{s,\ell+1} = \sum_{s' \in \mathcal{N}_s} W_{ss'} \boldsymbol{x}^{s',\ell}, \ \forall s \in [T],$$

  output : $\boldsymbol{x}^{s,\ell_t} \approx (1/T) \sum_{s'=1}^{T} W_{ss'} \boldsymbol{x}^{s',0}$ – the average

  where $\boldsymbol{W} \in \mathbb{R}_{+}^{T \times T}$ is a doubly stochastic, weighted adj. matrix of G
- ✓ – Geometric convergence – $\|\boldsymbol{x}^{s,\ell_t} - (1/T) \sum_{s=1}^{T} \boldsymbol{x}^{s,0}\|_\infty = \mathcal{O}(\lambda_2(\boldsymbol{W})^{\ell_t})$.

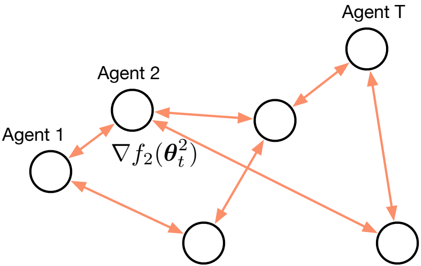
Decentralized FW (DeFW) algorithm

- **Setting**: agents are connected via a graph $G = (V, E)$.
- We want to compute **averages** over the network!
- *Gossip-based average consensus (G-AC) subroutine* [DKM+10] –

$$
\begin{aligned}
&\texttt{input}: \{\boldsymbol{x}_{s,0}\}_{s \in [T]} \text{ – initial values held by the agents} \\
&\texttt{repeat for } \ell = 0, 1, ..., \ell_t: \\
&\quad \texttt{gossip upd}: \boxed{\boldsymbol{x}^{s,\ell+1} = \sum_{s' \in \mathcal{N}_s} W_{ss'} \boldsymbol{x}^{s',\ell}, \ \forall s \in [T],} \\
&\texttt{output}: \boldsymbol{x}^{s,\ell_t} \approx (1/T) \textstyle\sum_{s'=1}^{T} W_{ss'} \boldsymbol{x}^{s',0} \text{ – the average}
\end{aligned}
$$

where $\boldsymbol{W} \in \mathbb{R}_+^{T \times T}$ is a doubly stochastic, weighted adj. matrix of G

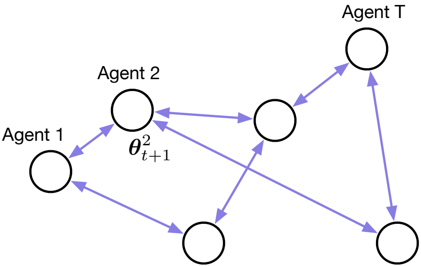
- ✓ – Geometric convergence – $\| \boldsymbol{x}^{s,\ell_t} - (1/T) \sum_{s=1}^{T} \boldsymbol{x}^{s,0} \|_\infty = \mathcal{O}(\lambda_2(\boldsymbol{W})^{\ell_t})$.



- **Aggregating**: apply the G-AC subroutine by setting $\boldsymbol{x}^{s,0} = \nabla f_s(\boldsymbol{\theta}_t^s)$ and $\ell_t = \Omega(\log t) \Longrightarrow$ each agent has an $\mathcal{O}(1/\sqrt{t})$-estimate of $\nabla F(\bar{\boldsymbol{\theta}}_t)$.
- Each agent computes $\boldsymbol{a}_t^s$ using the estimate of $\nabla F(\bar{\boldsymbol{\theta}}_t)$.
- **Consensus**: apply the G-AC subroutine by setting $\boldsymbol{x}^{s,0} = \boldsymbol{\theta}_{t+1}^s$ and $\ell_t = \Omega(\log t) \Longrightarrow$ each agent has an $\mathcal{O}(1/\sqrt{t})$-estimate of $\bar{\boldsymbol{\theta}}_{t+1}^s$

Communication Cost —

- ✓ — for *consensus step*, θ_t^s is at most $t \cdot T \ll n$ sparse
- ✓ — for *aggregating step*, we 'sparsify' $\nabla f_s(\theta_t^s)$ like in DistFW.



- **Aggregating**: apply the G-AC subroutine by setting $\boldsymbol{x}^{s,0} = \nabla f_s(\boldsymbol{\theta}_t^s)$ and $\ell_t = \Omega(\log t) \Longrightarrow$ each agent has an $\mathcal{O}(1/\sqrt{t})$-estimate of $\nabla F(\bar{\theta}_t)$.
- Each agent computes $\boldsymbol{a}_t^s$ using the estimate of $\nabla F(\bar{\theta}_t)$.
- **Consensus**: apply the G-AC subroutine by setting $\boldsymbol{x}^{s,0} = \boldsymbol{\theta}_{t+1}^s$ and $\ell_t = \Omega(\log t) \Longrightarrow$ each agent has an $\mathcal{O}(1/\sqrt{t})$-estimate of $\bar{\boldsymbol{\theta}}_{t+1}^s$

Communication Cost —

- ✓ — for *consensus step*, $\boldsymbol{\theta}_t^s$ is at most $t \cdot T \ll n$ sparse
- ✓ — for *aggregating step*, we 'sparsify' $\nabla f_s(\boldsymbol{\theta}_t^s)$ like in DistFW.

Convergence Analysis

- With *randomized co-ord. selection*, DistFW & DeFW $\approx$ sp. cases of sFW.
- Analyzing the convergence (rate) requires verifying (H1).

Convergence of DistFW and DeFW algorithms (informal)

For DistFW and DeFW with *rand. coordinate selection scheme*, if $p_t = \Omega(\sqrt{t})$ and $\ell_t = \Omega(\log(t))$, then (H1) holds. The following holds w.h.p. if F is strongly convex and $\boldsymbol{\theta}^\star \in \mathrm{int}(\mathcal{C})$,

$$F(\bar{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star) = \mathcal{O}(1/t).$$

- To achieve $F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^\star) \leq \epsilon$, we need $\Omega(1/\epsilon)$ iterations and communicating $\sim (1/\epsilon)^{3/2}$ (for DistFW) and $\sim (1/\epsilon)^2 \cdot \log(1/\epsilon)$ (for DeFW) non-zero real numbers $\Longrightarrow$ **Independent of n!**

Convergence rate comparisons

	DeFW (proposed)	PG-EXTRA[1]	D-PG[2]
Primal opt.: $F(\bar{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star)$	$\mathcal{O}(1/t)$	$\mathcal{O}(1/t)$	$\mathcal{O}(1/t)$
Comm. cost at iter. t	$\sim t \cdot T$	$\sim n$	$\sim n$
Comp. complexity at iter. t	$\sim \sqrt{t}$	$\sim n$	$\sim n$
Comm. cost for ϵ-optimality	$\sim (1/\epsilon)^2 \log(1/\epsilon)$	$\sim (1/\epsilon) \cdot n$	$\sim (1/\epsilon) \cdot n$

In terms of the communication cost...

- Low accuracy (when ϵ is large), DeFW $>$ PG-EXTRA or D-PG.
- High accuracy (when ϵ is small), DeFW $<$ PG-EXTRA or D-PG.

[1] [SLWY15] W. Shi, Q. Ling, G. Wu, and W. Yin, "A Proximal Gradient Algorithm for Decentralized Composite Optimization," TSP, 2015.

[2] [RNV10] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization," J. Optim. Theory. Appl., Dec., 2010.

Numerical Experiment – Settings

We apply DeFW on a distributed LASSO problem:

$$\min_{\boldsymbol{\theta}} \; \frac{1}{20} \sum_{s=1}^{20} \| \boldsymbol{y}_s - \boldsymbol{A}_s \boldsymbol{\theta} \|_2^2 \;\; \text{s.t.} \; \|\boldsymbol{\theta}\|_1 \leq r, \tag{7}$$

- *Dimensions* — $n = 5 \times 10^4$, $T = 20$ and $\boldsymbol{A}_s \in \mathbb{R}^{50 \times 50000}$
- *Parameters* — $\boldsymbol{y}_s \sim \mathcal{N}(\boldsymbol{A}_s \boldsymbol{\theta}_{true}, 0.01 \boldsymbol{I})$, $\|\boldsymbol{\theta}_{true}\|_0 = 25$ and $r = 1.5 \|\boldsymbol{\theta}_{true}\|_1$.
- *Network* — $G = (V, E)$ is Erdos-Renyi graph with connectivity $p = 0.3$, weights on $\boldsymbol{W}$ follows the Metropolis-Hastings rule [XB04].
- *DeFW* — we set $p_t = 2\lceil \sqrt{t} \rceil$, $\ell_t = \lceil \log(t) + 5 \rceil$.
- *Benchmark* — D-PG [RNV10] with step size $\alpha_t = 0.8/t$, PG-EXTRA [SLWY15] with fixed step size $\alpha = 1/n \approx 1/L$.

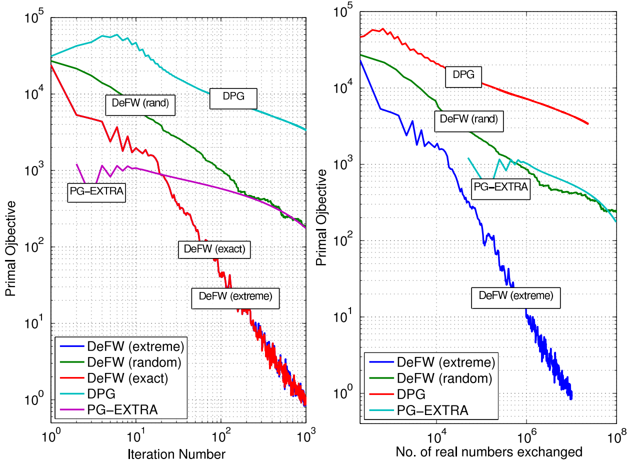


Fig. Comparing the primal objective value $F(\boldsymbol{\theta}_t) = (1/T)\sum_{s=1}^{T} f_s(\boldsymbol{\theta}_t^s)$. (Left) against the iteration number. (Right) against the number of real numbers communicated.

- PG-EXTRA outperforms DeFW (rand) at high accuracy.
- DeFW (extreme) outperforms the competing algorithms.

Conclusions, Future work

To conclude,

- We proposed two distributed FW-based algorithms for high-dimensional sparse optimization.
- Applied recent results on stochastic FW to analyze its performance.
- Proposed algorithms offer trade-offs between comm. cost and accuracy.

Future work —

- *Asynchronous* and fully *parallel* computations variants of D-FW.
- Analyze the performance with extreme coordinate selection.
- Extend D-FW to matrix completion problems.
- Implement and test D-FW on computer networks using real data set.

Questions?

[BLG+14] Aurélien Bellet, Yingyu Liang, Alireza Bagheri Garakani, Maria-Florina Balcan, and Fei Sha.
A Distributed Frank-Wolfe Algorithm for Communication-Efficient Sparse Learning.
pages 1–19, 2014.

[CC68] M. D. Canon and C. D. Cullum.
A tight upper bound on the rate of convergence of frank-wolfe algorithm.
SIAM Journal on Control, 6(4), 1968.

[DKM+10] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione.
Gossip Algorithms for Distributed Signal Processing.
Proc. IEEE, 98(11):1847–1864, November 2010.

[FW56] M. Frank and P. Wolfe.
An algorithm for quadratic programming.
Naval Res. Logis. Quart., 1956.

[Jag13] Martin Jaggi.
Revisiting frank-wolfe: Projection-free sparse convex optimization.
In *ICML*, volume 28, pages 427–435, June 2013.

[JST+14] Martin Jaggi, Virginia Smith, Martin Takac, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan.
Communication-efficient distributed dual coordinate ascent.
In *NIPS*, 2014.

[LJJ13] Simon Lacoste-Julien and Martin Jaggi.
An affine invariant linear convergence analysis for frank-wolfe algorithms.
In *NIPS*, 2013.

[LWM15] Jean Lafond, Hoi-To Wai, and Eric Moulines.
Convergence analysis of a stochastic projection-free algorithm.
ArXiv e-prints (1510.01171), 2015.

[LZ14] Guanghui Lan and Yi Zhou.
Conditional gradient sliding for convex optimization.
Technical Report, 2014.

[RNV10] S. Sundhar Ram, Angelia Nedic, and V. V. Veeravalli.
Distributed stochastic subgradient projection algorithms for convex optimization.
Journal of Optimzation Theory and Applications, 147(3):516–545, December 2010.

[RNV12] S. S. Ram, A. Nedic, and V. V. Veeravalli.
A new class of distributed optimization algorithms : application to regression of distributed data.
Optimization Methods and Software, (1):37–41, February 2012.

[SLWY15] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin.
A Proximal Gradient Algorithm for Decentralized Composite Optimization.
IEEE Trans. on Signal Process., pages 1–11, 2015.

[XB04] Lin Xiao and Stephen Boyd.
Fast linear iterations for distributed averaging.
Systems & Control Letters, 53(1):65–78, September 2004.