D-FW: COMMUNICATION EFFICIENT DISTRIBUTED ALGORITHMS FOR HIGH-DIMENSIONAL SPARSE OPTIMIZATION

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

[†] Institut Mines-Telecom, Telecom ParisTech, CNRS LTCI, Paris, France.
[‡]School of ECEE, Arizona State Univ., AZ, USA. [#]CMAP, Ecole Polytechnique, Palaiseau, France.
Emails: jean.lafond@telecom-paristech.fr, htwai@asu.edu, eric.moulines@polytechnique.edu

ABSTRACT

We propose distributed algorithms for high-dimensional sparse optimization. In many applications, the parameter is sparse but high-dimensional. This is pathological for existing distributed algorithms as the latter require an information exchange stage involving transmission of the full parameter, which may not be sparse during the *intermediate* steps of optimization. The novelty of this work is to develop communication efficient algorithms using the stochastic Frank-Wolfe (sFW) algorithm, where the gradient computation is inexact but controllable. For star network topology, we propose an algorithm with low communication cost and establishes its convergence. The proposed algorithm is then extended to perform decentralized optimization on general network topology. Numerical experiments are conducted to verify our findings.

***Index Terms*—** decentralized algorithm, sparse optimization, large-scale optimization, communication efficient algorithm

1. INTRODUCTION

Consider the following constrained optimization problem:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^n} \; F(\boldsymbol{\theta}) := \tfrac{1}{T}\sum_{s=1}^{T} f_s(\boldsymbol{\theta}), \text{ s.t. } \boldsymbol{\theta}\in\mathcal{C}. \tag{1}$$

We assume that $F(\boldsymbol{\theta})$ is convex. While the techniques developed in this paper are applicable to general problems, we focus on the sparsity constrained case where the constraint set $\mathcal{C}$ is an ℓ_1-ball, i.e.,

$$\mathcal{C} := \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_1 \le r\}. \tag{2}$$

We are interested in a distributed setting where the sth loss function $f_s(\boldsymbol{\theta})$ is kept privately by the sth agent. We focus on high-dimensional sparse optimization problems where the parameter to be estimated is sparse, i.e., $\|\boldsymbol{\theta}^\star\|_0 = k \ll n$, but lives in a high dimensional space, i.e., $n \gg 1$. As an application example, we consider $f_s(\boldsymbol{\theta})$ to be the squared loss function $f_s(\boldsymbol{\theta}) = (1/2)\|\boldsymbol{y}_s - \boldsymbol{A}_s\boldsymbol{\theta}\|_2^2$. Here, $\boldsymbol{y}_s$ is the observation obtained at agent s through its measurement matrix $\boldsymbol{A}_s$. Both $\boldsymbol{y}_s, \boldsymbol{A}_s$ are private data that the agents are not willing to share. In this case, problem (1) reduces to a distributed LASSO problem.

Our work is motivated by a growing number of problems in machine learning and signal processing, e.g., big-data optimization [1] and large-scale sparse learning [2, 3]. Though Problem (1) is convex, its high dimensionality has driven recent works to consider solving it using first-order optimization algorithms. Some examples such as stochastic projected/proximal-gradient methods can be found in [4–6] and their convergence rates are shown to be as fast as $\mathcal{O}(1/t^2)$ [7], where t is the iteration number.

⋆J. Lafond and H.-T. Wai have contributed equally. The work of J. Lafond is supported by Direction Générale de l'Armement and the labex LMH (ANR-11-LABX-0056-LMH) and H.-T. Wai by NSF CCF-1011811.

The aforementioned works focus on the case when $F(\boldsymbol{\theta})$ is known completely at a central processor. In a distributed setting where $f_s(\cdot)$ is kept privately, recent advances have demonstrated that the convergence to $\boldsymbol{\theta}^\star$ can be established using an algorithm with only local communication steps that are constrained by the topology of the communication network [8–11]. However, existing works typically require the full parameter $\boldsymbol{\theta}$ (and/or the local gradient $\nabla f_s(\boldsymbol{\theta})$) to be exchanged during each iteration. While the optimal solution $\boldsymbol{\theta}^\star$ to (1) is sparse, the intermediate solution during the algorithm may be dense. In high-dimensional problems, the incurred communication overhead maybe overwhelming.

This work proposes distributed algorithms for (1) that are *communication efficient*. Our plan is to exploit the structures in Frank-Wolfe algorithm [12] to develop distributed methods that only transmit a small number of scalar variables in the network at every iteration. We propose two algorithms: one for the star network topology and one for the general network topology. Several communication cost reduction techniques are developed. The proposed schemes select a small number of agents and coordinates for information exchange and provide estimates of the gradient with controllable accuracies. Finally, we show that the distributed algorithms converge as fast as $\mathcal{O}(\log(t)/t)$, while the communication cost at the tth iteration is $\mathcal{O}(t)$ for star network; and is $\mathcal{O}(t\log t)$ for general networks. The proposed schemes are suitable when the target solution accuracy is moderate.

1.1. Relation to Prior Work

The Frank-Wolfe algorithm [12] is recently rediscovered as an appealing tool to handle large-scale convex optimization and it has been applied to a number of practical problems [13–16]; see the surveys [17, 18]. In fact, a communication-efficient distributed algorithm based on Frank-Wolfe algorithm has been studied in [19]. However, a different distributed data model is considered here. This paper is also related to the CoCoA method in [20]. Under the assumption of a star network topology, [20] focuses on solving the dual problem of (1) using a dual decomposition like technique [21]. The convergence rate is geometric. Their method, however, involves a step that transmits the full parameter in the network.

2. STOCHASTIC FRANK-WOLFE ALGORITHM

This section describes the stochastic Frank-Wolfe (sFW) algorithm for (1) and reports on a recent convergence result. We suppose that at iteration t, we have access to a noisy estimate of gradient:

$$\hat{\nabla}_t F(\boldsymbol{\theta}) = \nabla F(\boldsymbol{\theta}) + \boldsymbol{\epsilon}_t(\boldsymbol{\theta}), \tag{3}$$

where $\boldsymbol{\epsilon}_t(\boldsymbol{\theta})$ denotes the gradient estimation error. The sFW algorithm is described by the following recursion:

$$\boldsymbol{a}_t \leftarrow \arg\min_{\boldsymbol{a}\in\mathcal{C}} \langle \boldsymbol{a}, \hat{\nabla}_t F(\boldsymbol{\theta}_t)\rangle, \tag{4a}$$

$$\boldsymbol{\theta}_{t+1} \leftarrow (1-1/t)\boldsymbol{\theta}_t + (1/t)\boldsymbol{a}_t, \tag{4b}$$

and the algorithm is repeated with $t = t+1$. For the constraint set of interest, i.e., $\mathcal{C} := \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_1 \leq r\}$, the linear optimization step (4a) can be carried out as:

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

where $i_t = \arg\max_{j\in[n]} |[\hat{\nabla}_t F(\boldsymbol{\theta}_t)]_j|$ corresponds to the maximum magnitude coordinate in $\hat{\nabla}_t F(\boldsymbol{\theta}_t)$. We have used $[\boldsymbol{x}]_j$ to denote the jth element in the vector $\boldsymbol{x}$. Moreover, $\boldsymbol{e}_i$ is the ith canonical basis vector in $\mathbb{R}^n$.

We remark that in the above sFW algorithm, the update direction $\boldsymbol{a}_t$ is a sparse vector that merely adds a single new coordinate to $\boldsymbol{\theta}_t$. In particular, if $\boldsymbol{\theta}_0 = \mathbf{0}$, then $\boldsymbol{\theta}_t$ is at most t-sparse. As we shall see later, this is an important property that enables us to develop communication-efficient distributed algorithms for (1).

To study the convergence of the sFW algorithm, we introduce the following assumptions.

Assumption 1 *With probability at least* $1-\Delta$*,*

$$\|\boldsymbol{\epsilon}_t\| \leq \sigma\sqrt{\eta_t^{\Delta}/t},\ \forall\, t \geq 1\ , \tag{6}$$

for some $\sigma > 0$*,* η_t^{Δ} *is non-decreasing,* $\eta_t^{\Delta} \geq 1$ *and* $\Delta \in [0,1]$*. Furthermore,* η_t^{Δ}/t *is non-increasing and* $\lim_{t\to\infty} \eta_t^{\Delta}/t = 0$*.*

The diameter of the constraint set is denoted by $\rho := \sup_{\theta,\theta'\in\mathcal{C}} \|\theta - \theta'\|_1$. As $\mathcal{C}$ is bounded, ρ is finite. The objective function F is L-gradient Lipchitz and μ strongly convex. Finally, when $\mu > 0$, there is a unique optimal solution $\boldsymbol{\theta}^\star$ whose distance to the boundary is denoted by

$$\delta := \inf_{\boldsymbol{s}\in\partial\mathcal{C}} \|\boldsymbol{s} - \boldsymbol{\theta}^\star\|\ , \tag{7}$$

with $\partial\mathcal{C}$ the boundary set of $\mathcal{C}$. Notice that potentially we can have $L = \infty$ or $\mu = 0$ or $\delta = 0$. We have the following Theorem that is borrowed from [22]:

Theorem 1 *Consider the sFW algorithm with the gradient estimate given by* (3)*. Under Assumption 1, the following holds with probability at least* $1-\Delta$ *for all* $t \geq 2$*:*

$$F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^\star) \leq \min\left\{ D'\sqrt{\tfrac{\eta_t^{\Delta}}{t}}, D\tfrac{\eta_t^{\Delta}}{t}\right\}\ , \tag{8}$$

where $D' = 2(\rho\sigma + L\rho^2/2)$ *and* $D = 6(\rho\sigma + L\rho^2/2)^2/(\delta^2\mu)$*.*

From Theorem 1, the convergence rate of the sFW algorithm is at least $\mathcal{O}(\sqrt{\eta_t^{\Delta}/t})$ (since $D' < \infty$ for standard convex problems), while the rate can be accelerated to $\mathcal{O}(\eta_t^{\Delta}/t)$ if $F(\cdot)$ is strongly convex and $\boldsymbol{\theta}^\star$ lies in the interior of $\mathcal{C}$.

A consequence of Theorem 1 is that the sFW algorithm converges even when the gradient estimate is noisy. To develop a communication efficient distributed algorithm, our goal is to repeat the calculations in sFW using the suitable steps for the in-network operations to produce *local* gradient estimates satisfying Assumption 1.

3. DISTRIBUTED FW ALGORITHM

The first proposed distributed algorithm assumes that the agents are connected through star network topology. In particular, there exists a *hub* agent which is connected to all T agents in the network.

Under this setting, at iteration t, it is possible for the hub to evaluate $\nabla F(\boldsymbol{\theta}_t)$ by requesting *local gradient* $\nabla f_s(\boldsymbol{\theta}_t)$ from each agent. Initializing with $\boldsymbol{\theta}_0 = \mathbf{0}$, we have the Distributed Frank-Wolfe (DistFW) algorithm. At the tth iteration, we do:

DistFW Algorithm:

1. Compute $\nabla F(\boldsymbol{\theta}_t) = (1/T)\sum_{s=1}^T \nabla f_s(\boldsymbol{\theta}_t)$ by aggregating the gradient vectors from all agents.
2. Find $\boldsymbol{a}_t$ using (5) at the hub and broadcast $\boldsymbol{a}_t$ to the agents. Each agent computes (4b) to update the parameter.

It is easy to check that the above DistFW algorithm converges to an optimum solution of (1) at rate $\mathcal{O}(1/t)$ since the operations are identical to the classical Frank-Wolfe algorithm [12].

The Step 1 in DistFW requires the transmission of an n-dimensional vector from each agent. Moreover, when broadcasting $\boldsymbol{a}_t$ to the agents, one only need to broadcast the coordinate number i_t together with the associated sign of $[\nabla F(\boldsymbol{\theta}_t)]_{i_t}$. The communication cost per iteration is nT real numbers and T integers.

3.1. Communication Efficient DistFW

To reduce the communication cost per iteration in Step 1 of DistFW, we perform sampling on both the number of agents involved and the coordinates of the gradient vector. Specifically, at iteration t, we select t agents (while allowing overlaps) and aggregate their corresponding gradient vectors. The selected agents are denoted by $i_t(1), ..., i_t(t)$. For the sampling of coordinates, we consider the following two schemes for a selected agent s:

- *Random Coordinate Selection* — Agent s selects the coordinate $i \in [n] := \{1, ..., n\}$ with probability p/n.
- *Extremal Coordinate Selection* — Agent s sorts $\nabla f_s(\boldsymbol{\theta}_t)$ and selects the $p/2$ coordinates that correspond to the maximum and minimum elements in the vector, respectively.

The second scheme is motivated by the fact that in the sFW algorithm, we only select the coordinate i_t that has the maximum magnitude in $\hat{\nabla} F(\boldsymbol{\theta}_t)$. Under the above schemes, computing $\hat{\nabla}_t F(\boldsymbol{\theta}_t)$ requires an expected communication cost of only pt.

We analyze the case when random coordinate selection is used and demonstrate that a convergence rate as fast as $\mathcal{O}(\log(t)/t)$ can be obtained. The kth coordinate of the resultant $\hat{\nabla}_t F(\boldsymbol{\theta}_t)$ can be modeled by the following random variable:

$$[\hat{\nabla}_t F(\boldsymbol{\theta}_t)]_k = \tfrac{1}{t}\sum_{s=1}^t \nabla f_{i_t(s)}(\boldsymbol{\theta}_t)\boldsymbol{\xi}_{i_t(s)}^k, \tag{9}$$

where $\boldsymbol{\xi}_{i_t(s)}^k$ is a Bernoulli random variable with $\mathbb{P}(\boldsymbol{\xi}_{i_t(s)}^k = 1) = p/n$. Assuming that $\|\nabla f_i(\boldsymbol{\theta})\|_\infty \leq B_1$ for all $\boldsymbol{\theta} \in \mathcal{C}$ and $i \in [T]$, the gradient estimate error can be bounded as:

Proposition 1 *With probability at least* $1 - \pi^2\Delta/6$*, the following holds for all* $t \geq 1$ *and* $\boldsymbol{\theta} \in \mathcal{C}$*:*

$$\|\boldsymbol{\epsilon}_t(\boldsymbol{\theta})\|_\infty \leq \tfrac{n}{p}\sqrt{\tfrac{2B_1^2(\log(2nt^2) - \log\Delta)}{t}}, \tag{10}$$

where $\boldsymbol{\epsilon}_t(\boldsymbol{\theta}) := (n/p)\hat{\nabla}_t F(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta})$*.*

Proof. First observe that $\mathbb{E}[\nabla f_{i_t(s)}(\boldsymbol{\theta}_t)\boldsymbol{\xi}_{i_t(s)}^k] = (p/n)[\nabla F(\boldsymbol{\theta})]_k$. The kth coordinate of $\boldsymbol{\epsilon}_t(\boldsymbol{\theta})$ can be written as

$$[\boldsymbol{\epsilon}_t(\boldsymbol{\theta})]_k = \frac{n}{tp}\sum_{s=1}^t \big(\nabla f_{i_t(s)}(\boldsymbol{\theta}_t)\boldsymbol{\xi}_{i_t(s)}^k - \mathbb{E}[\nabla f_{i_t(s)}(\boldsymbol{\theta}_t)\boldsymbol{\xi}_{i_t(s)}^k]\big),$$

consequently $\boldsymbol{\epsilon}_t(\boldsymbol{\theta})$ is zero mean. As $\|\nabla f_i(\boldsymbol{\theta})\|_\infty \leq B_1$, we have $|\nabla f_{i_t(s)}(\boldsymbol{\theta}_t)\boldsymbol{\xi}_{i_t(s)}^k - (p/n)[\nabla F(\boldsymbol{\theta})]_k| \leq B_1$ with probability one for all k. Applying the Hoefding's inequality [23] gives:

$$\mathbb{P}\big(|[\boldsymbol{\epsilon}_t(\boldsymbol{\theta})]_k| \geq x,\ \forall\, k \in [n]\big) \leq 2ne^{-tx^2(p/n)^2/(2B_1^2)}, \tag{11}$$

for all $x \geq 0$. Now, setting $x = \frac{n}{p}\sqrt{\frac{2B_1^2(\log(2nt^2)-\log\Delta)}{t}}$ gives Δ/t^2 on the right hand side of (11). Applying a union bound argument gives our desired result. **Q.E.D.**

Notice that the bound (10) applies on the scaled gradient estimate $(n/p)\hat{\nabla}_t F(\boldsymbol{\theta})$ instead of $\hat{\nabla}_t F(\boldsymbol{\theta})$. It remains relevant to the DistFW algorithm as the linear optimization (4a) is scale invariant, i.e., $\arg\min_{\boldsymbol{a}\in\mathcal{C}}\langle \boldsymbol{a}, \hat{\nabla}_t F(\boldsymbol{\theta}_t)\rangle = \arg\min_{\boldsymbol{a}\in\mathcal{C}}\langle \boldsymbol{a}, \alpha\hat{\nabla}_t F(\boldsymbol{\theta}_t)\rangle$ for any $\alpha > 0$. Finally, Assumption 1 can be satisfied by the proposed scheme with $\|\boldsymbol{\epsilon}_t(\boldsymbol{\theta}_t)\|_\infty = \mathcal{O}(\sqrt{\log(t)/t})$ and a convergence rate as fast as $\mathcal{O}(\log t/t)$ can be guaranteed by applying Theorem 1. The communication cost per iteration is pt real numbers and T integers.

4. DECENTRALIZED FW ALGORITHM

In this section, we assume the agents are connected in a network described by a connected, undirected simple graph $G = (V, E)$, where $V = [T] = \{1, \dots T\}$ and $E \subseteq V \times V$.

Our goal is to mimic the sFW algorithm by restricting to local communication. In particular, we apply the gossip average consensus (GAC) routine [24] for computing averages over the network. To describe the routine, let $\boldsymbol{x}_{s,0}$ be a vector stored at agent s at initialization and ℓ be the index of recursion, we have:

$$\texttt{GAC}: \boldsymbol{x}_{s,\ell+1} = \textstyle\sum_{s'=1}^T W_{ss'}\boldsymbol{x}_{s',\ell},\ \forall s \in [T], \tag{12}$$

where $\boldsymbol{W}$ with $[\boldsymbol{W}]_{ss'} = W_{ss'}$ is a non-negative, doubly stochastic matrix that respects the structure of G, and we design $\boldsymbol{W}$ such that $|\lambda_2(\boldsymbol{W})| < 1$ as G is connected; see [25] for examples of the construction algorithms. The GAC routine converges geometrically to $(1/T)\sum_{s=1}^T \boldsymbol{x}^{s,0}$ [26]:

Fact 1 For all $\ell \geq 1$ and $s \in [T]$ and using linear algebra, we have

$$\left\|\boldsymbol{x}^{s,\ell} - \tfrac{1}{T}\textstyle\sum_{s=1}^T \boldsymbol{x}^{s,0}\right\|_\infty \leq B_2 \cdot \lambda_2(\boldsymbol{W})^\ell, \tag{13}$$

where $B_2 = \max_{k\in[n]}\sqrt{\sum_{s'=1}^T([\boldsymbol{x}^{s',0}]_k)^2}$ and $\lambda_2(\boldsymbol{W}) < 1$ is the second largest eigenvalue of $\boldsymbol{W}$.

Notice if we set $\ell = C_1\log(t)$ for $C_1 \geq -1/(2\log(\lambda_2(\boldsymbol{W}))$, then the above upper bound becomes $\mathcal{O}(1/\sqrt{t})$.

We propose the following decentralized Frank-Wolfe (DeFW) algorithm. At iteration t, the local parameter stored at the sth agent is denoted as $\boldsymbol{\theta}_t^s$. We define $\bar{\boldsymbol{\theta}}_t := (1/T)\sum_{s=1}^T \boldsymbol{\theta}_t^s$ as the instantaneous average. Initializing with $\boldsymbol{\theta}_0^s = \mathbf{0}$ for all s, the tth iteration of DeFW algorithm can be described as:

DeFW Algorithm:

1. Compute $\boldsymbol{g}_t^{s,\ell_t} \approx \nabla F(\bar{\boldsymbol{\theta}}_t)$ decentralizedly using $(\boldsymbol{g}_t^{s,0} := \nabla f_s(\tilde{\boldsymbol{\theta}}_t^s))_{s=1}^T$ as the initialization to GAC routine (12) with $\ell_t = C_1\log(t)$ recursions.
2. Find $\boldsymbol{a}_t^s$ using (5) and the gradient estimate $\boldsymbol{g}_t^{s,\ell_t}$ at agent s for all $s \in [T]$ locally.
3. Update $\boldsymbol{\theta}_{t+1}^s = (1-1/t)\tilde{\boldsymbol{\theta}}_t^s + (1/t)\boldsymbol{a}_t^s$.
4. Compute $\tilde{\boldsymbol{\theta}}_{t+1}^s \approx \bar{\boldsymbol{\theta}}_{t+1}$ decentralizedly using $(\boldsymbol{\theta}_{t+1}^s)_{s=1}^T$ as the initialization to GAC routine (12) with $\ell_t = C_1\log(t)$ recursions.

Suppose that the gradient $\nabla f_s(\cdot)$ is L-Lipschitz continuous and is bounded, the following corollary is easy to show using Fact 1.

Corollary 1 *The gradient estimate $\boldsymbol{g}_{s,\ell_t}$ satisfies:*

$$\|\boldsymbol{g}_t^{s,\ell_t} - \nabla F(\bar{\boldsymbol{\theta}}_t)\|_\infty = \mathcal{O}(1/\sqrt{t}) \tag{14}$$

Proof. Using the triangular inequality,

$$\begin{aligned}\|\boldsymbol{g}_t^{s,\ell_t} - \nabla F(\bar{\boldsymbol{\theta}}_t)\|_\infty &\leq \|\boldsymbol{g}_t^{s,\ell_t} - \tfrac{1}{T}\textstyle\sum_{s'=1}^T \nabla f_{s'}(\tilde{\boldsymbol{\theta}}_t^{s'})\|_\infty + \\ &\quad \|\tfrac{1}{T}\textstyle\sum_{s'=1}^T(\nabla f_{s'}(\tilde{\boldsymbol{\theta}}_t^{s'}) - \nabla f_s(\bar{\boldsymbol{\theta}}_t)\|_\infty \\ &\leq \|\boldsymbol{g}_t^{s,\ell_t} - \tfrac{1}{T}\textstyle\sum_{s'=1}^T \nabla f_{s'}(\tilde{\boldsymbol{\theta}}_t^{s'})\|_\infty + L\max_{s'\in[T]}\|\tilde{\boldsymbol{\theta}}_t^{s'} - \bar{\boldsymbol{\theta}}_t\|_\infty\end{aligned} \tag{15}$$

As $\ell_t = C_1\log(t)$, applying Fact 1 shows that the right hand side of (15) can be upper bounded by $\mathcal{O}(1/\sqrt{t})$. **Q.E.D.**

Consequently, the DeFW algorithm can be analyzed as an sFW algorithm operated on $\bar{\boldsymbol{\theta}}_t$ where $\boldsymbol{g}_t^{s,\ell_t}$ serves as an estimate of $\hat{\nabla}_t F(\bar{\boldsymbol{\theta}}_t)$. Theorem 1 shows that the objective value converges at a rate that can be as fast as $\mathcal{O}(1/t)$.

4.1. Communication Efficient DeFW

We first show that the GAC routine applied in Step 4 of DeFW can be implemented with low communication cost. This can be seen by noting that $\tilde{\boldsymbol{\theta}}_t$ is at most $t \cdot T$ sparse, since each DeFW iteration will add at most T new coordinates into $\tilde{\boldsymbol{\theta}}_t$. As such, this step can be completed by requiring each agent to exchange $C_1\log(t)\cdot t\cdot T$ real numbers at iteration t.

On the other hand, the GAC routine applied in Step 1 of DeFW requires exchanging of the gradient vector $\nabla f_s(\boldsymbol{\theta}_t)$, which may be dense in general. We consider the same coordinate selection schemes in Section 3.1 to reduce communication cost by sampling a subset of the coordinates. Notice that we now consider sampling (an average of) $\min\{p_t, n\}$ coordinates at each agent and the GAC routine communicates on the *union* of the selected coordinates. This can be realized in a decentralized algorithm as an agent will notice the selected coordinates of its neighbors once they begin to communicate, after a fixed number of GAC recursions C_2 (proportional to the diameter of G) the union of the selected coordinates will be acquired at all agents.

Let the set of coordinates selected by agent s be $\mathcal{S}_t^s$. The following sparse vector will serve as an input to the GAC routine (12):

$$\hat{\boldsymbol{g}}_t^{s,0} = \nabla f_s(\tilde{\boldsymbol{\theta}}_t^s) \odot \boldsymbol{\xi}_t, \tag{16}$$

where $\odot$ denotes the Hadamard's product, $[\boldsymbol{\xi}_t]_k = 1$ if $k \in \cup_{s=1}^T \mathcal{S}_t^s$ and is zero otherwise. Note that $\boldsymbol{\xi}_t$ is a Bernoulli random vector. The (expected) sparsity of $\hat{\boldsymbol{g}}_t^{s,0}$ is upper bounded by $\min\{p_t T, n\}$. This step can be completed with a communication cost of at most $(C_1\log(t) + C_2)\cdot\min\{p_t T, n\}$.

We now analyze the case with randomized coordinate selection. Notice that $\mathbb{E}[\boldsymbol{\xi}_t] = \xi_{mean}\mathbf{1}$, $\xi_{mean} = (1-(1-\min\{p_t,n\}/n)^T)$ and $\min\{p_t,n\}/n \leq \xi_{mean} \leq \min\{p_t T, n\}/n$. We obtain the following error bound:

Proposition 2 *With probability at least $1 - \pi^2\Delta/6$, the following holds for all $t \geq 1$ and $\boldsymbol{\theta} \in \mathcal{C}$:*

$$\|\xi_{mean}^{-1}\hat{\boldsymbol{g}}_t^{s,\ell_t} - \nabla F(\bar{\boldsymbol{\theta}}_t)\|_\infty = \mathcal{O}\Big(\frac{n\sqrt{\log(2nt^2) - \log\Delta}}{p_t}\Big) \tag{17}$$

if $p_t < n$, and $\|\xi_{mean}^{-1}\hat{\boldsymbol{g}}_t^{s,\ell_t} - \nabla F(\bar{\boldsymbol{\theta}}_t)\|_\infty = \mathcal{O}(1/\sqrt{t})$ if $p_t \geq n$.

Proof. We begin by applying the triangular inequality:

$$\begin{aligned}\|\xi_{mean}^{-1}\hat{\boldsymbol{g}}_t^{s,\ell_t} - \nabla F(\bar{\boldsymbol{\theta}}_t)\|_\infty &\leq \xi_{mean}^{-1}\|\hat{\boldsymbol{g}}_t^{s,\ell_t} - \nabla F(\bar{\boldsymbol{\theta}}_t)\odot\boldsymbol{\xi}_t\|_\infty \\ &\quad + \|\nabla F(\bar{\boldsymbol{\theta}}_t)\odot(\xi_{mean}^{-1}\boldsymbol{\xi}_t - \mathbf{1})\|_\infty.\end{aligned}$$

For the former term, from Corollary 1 it can be upper bounded by $\mathcal{O}(\xi_{mean}^{-1}/\sqrt{t})$. For the latter term, we first apply the inequality

$\|\nabla F(\bar{\boldsymbol{\theta}}_t) \odot (\xi_{mean}^{-1}\boldsymbol{\xi}_t - \mathbf{1})\|_\infty \leq \|\nabla F(\bar{\boldsymbol{\theta}}_t)\|_\infty \|(\xi_{mean}^{-1}\boldsymbol{\xi}_t - \mathbf{1})\|_\infty$ of Hadamard's product [27]. Then, applying the Hoefding's inequality [23] and a union bound argument show that with probability at least $1 - (\pi^2\Delta/6)$, we have:

$$\begin{aligned} &\|\nabla F(\bar{\boldsymbol{\theta}}_t) \odot (\xi_{mean}^{-1}\boldsymbol{\xi}_t - \mathbf{1})\|_\infty \\ &\quad \leq \xi_{mean}^{-1}\|\nabla F(\bar{\boldsymbol{\theta}}_t)\|_\infty \sqrt{(\log(2nt^2) - \log\Delta)/2}, \end{aligned} \tag{18}$$

for all $t \geq 1$. Using the upper bound $\xi_{mean}^{-1} \leq n/\min\{p_t, n\}$ yields the desired result. As a remark, we notice that when $p_t = n$, the error bound can be improved to $\mathcal{O}(1/\sqrt{t})$ since $\xi_{mean}^{-1}\boldsymbol{\xi}_t - \mathbf{1} = \mathbf{0}$ in the latter case. **Q.E.D.**

As a consequence of Proposition 2, the error of gradient estimate converges as $\mathcal{O}(\sqrt{\log(t)/p_t})$. Setting $p_t = C_3\sqrt{t}$ suffices to let us apply Theorem 1. In particular, the DeFW algorithm can converge at rate as fast as $\mathcal{O}(\log(t)/t)$. Meanwhile, the communication cost per iteration is $\mathcal{O}(t\log t)$.

5. NUMERICAL RESULTS

This section performs numerical experiments to verify our findings on the proposed Frank-Wolfe based algorithms. We focus on the distributed LASSO problem where $f_s(\boldsymbol{\theta}) = (1/2)\|\boldsymbol{y}_s - \boldsymbol{A}_s\boldsymbol{\theta}\|_2^2$ is the squared loss function.

Due to space limitation, we consider solving a small-scale problem of dimension $n = 5 \times 10^4$ using synthetic data and we set $T = 20$. For each s, the matrix $\boldsymbol{A}_s$ is generated with independent $\mathcal{N}(0,1)$ elements with dimension of 50×50000 and $\boldsymbol{y}_s \sim \mathcal{N}(\boldsymbol{A}_s\boldsymbol{\theta}_{true}, \sigma^2\boldsymbol{I})$. The parameter $\boldsymbol{\theta}_{true} \in \mathbb{R}^{50000}$ is generated as a sparse vector with sparsity 0.0005 and independent $\mathcal{N}(0,1)$ entries on the non-zeros. There are approximately 25 non-zeros elements in $\boldsymbol{\theta}_{true}$. The noise variance σ^2 is set to 0.01 and the constraint set is $\mathcal{C} = \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_1 \leq 1.5\|\boldsymbol{\theta}_{true}\|_1\}$. Notice that $\delta > 0$ in this case.

As the DistFW algorithm can be seen as a special case of the DeFW algorithm, the following discussion focuses on the DeFW algorithm. For benchmarking purpose, we compare the DeFW algorithm with the distributed projected gradient (DPG) method in [8] and the projected gradient exact first order algorithm (PG-EXTRA) in [9]. The iterates of the latter algorithm is proven to converge at an ergodic rate of $\mathcal{O}(1/t)$, where t is the iteration number. A drawback for these two algorithms is that they both require exchanging the local parameter estimate $\boldsymbol{\theta}_t^s$ during each iteration, which may not be sparse during the intermediate steps. For instance, at the tth iteration of the DPG algorithm, for all $s \in [T]$, we do

$$\boldsymbol{\theta}_{t+1}^s = \mathcal{P}_{\mathcal{C}}\big(\textstyle\sum_{s'=1}^T W_{ss'}\boldsymbol{\theta}_t^{s'} - \alpha_t \nabla f_s(\sum_{s'=1}^T W_{ss'}\boldsymbol{\theta}_t^{s'})\big), \tag{19}$$

where $\alpha_t > 0$ is a step size that satisfies $\sum_{t=1}^\infty \alpha_t = \infty$ and $\sum_{t=1}^\infty \alpha_t^2 < \infty$; and we repeat with $t = t+1$. As seen, the sparsity of operands in (19) cannot be controlled and the DPG algorithm may require a high communication cost.

We have implemented the tested algorithms in `MATLAB`. The communication network G is generated as an Erdos-Renyi graph with connectivity $p = 0.3$ and the doubly stochastic matrix $\boldsymbol{W}$ is generated according to the Metropolis-Hastings rule described in [25]. For the DeFW algorithms, we set $p_t = 2\lceil\sqrt{t}\rceil$ and $\ell_t = \lceil\log(t) + 5\rceil$. For the DPG algorithm, we set $\alpha_t = 0.8/t$. For the PG-EXTRA algorithm, we set a fixed step size $\alpha = 1/n \approx \mathcal{O}(1/L)$, where L is the Lipschitz constant for the gradient ∇f_s, and $\tilde{\boldsymbol{W}} = (\boldsymbol{I} + \boldsymbol{W})/2$[1].

[1]Notice that we considered a scenario with $n \gg 0$, which forces us to take a relatively small step size for PG-EXTRA.

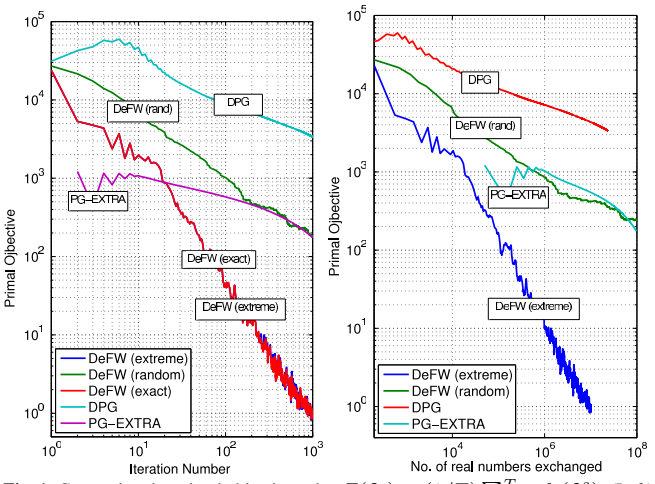

Fig. 1: Comparing the primal objective value $F(\boldsymbol{\theta}_t) = (1/T)\sum_{s=1}^T f_s(\boldsymbol{\theta}_t^s)$. (Left) against the iteration number. (Right) against the number of real numbers communicated.

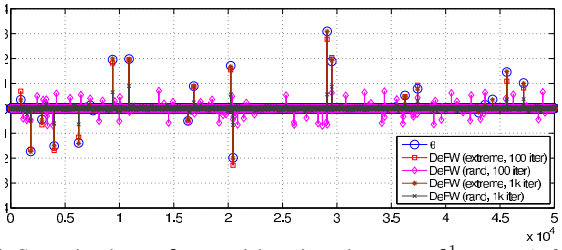

Fig. 2: Comparing the true $\boldsymbol{\theta}_{true}$ and the estimated parameter $\boldsymbol{\theta}_t^1$ at agent 1 after 100 and 1000 iterations from DeFW.

The results of our numerical example can be found in Figure 1 and Figure 2. In the legend, 'DeFW (extreme)', 'DeFW (random)' and 'DeFW (exact)' denote the DeFW algorithm with extremal, random and without coordinate selection, respectively. We first notice from Figure 1 (left) that the DeFW algorithms converge at a rate of $\mathcal{O}(1/t)$, corroborating with the analysis from Theorem 1. Secondly, Figure 1 (right) compares the primal objective against the number of real numbers exchanged during the algorithms, where the number of real numbers exchanged was obtained by precisely counting the number of non-zeros in the vectors exchanged between agents. We see that the proposed DeFW algorithms have outperformed DPG and PG-EXTRA. This is because the proposed algorithm exploits the sparse optimization structure in the sFW algorithm. Another interesting observation is that the extremal coordinate selection scheme has outperformed the random one for DeFW. This is possible as the local extremal coordinates are likely to be the maximum magnitude coordinate in $\nabla F(\boldsymbol{\theta}_t)$ selected at iteration t. Lastly, Figure 2 shows that the iterates in DeFW are sparse during the iteration.

6. CONCLUSIONS

In this paper, we have proposed two distributed algorithms for high-dimensional convex optimization problems. Our algorithms are suitable for cases when the parameter to be learnt is sparse. In particular, we explicitly exploit the structure in sFW algorithm to develop communication cost saving schemes. The algorithms with communication cost reduction schemes are proven analytically and empirically to converge at a comparable rate to existing algorithms, while requiring a significantly lower communication cost.

Our future work include applying the Frank-Wolfe based distributed algorithms to problems with general constraint set $\mathcal{C}$, and the analysis of the DistFW / DeFW algorithm with the extremal coordinate selection scheme.

7. REFERENCES

[1] V. Cevher, S. Becker, and M. Schmidt, "Convex Optimization for Big Data: Scalable, randomized, and parallel algorithms for big data analytics," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 32–43, Sep. 2014.

[2] E. Candes and T. Tao, "Decoding by Linear Programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[3] J. Langford, L. Li, and T. Zhang, "Sparse online learning via truncated gradient," *NIPS*, 2009.

[4] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[5] N. L. Roux, M. Schmidt, and F. R. Bach, "A stochastic gradient method with an exponential convergence rate for finite training sets," in *NIPS*, December 2012, pp. 2663–2671.

[6] L. Rosasco, S. Villa, and B. C. Vu, "Convergence of Stochastic Proximal Gradient Algorithm," *ArXiv e-prints*, 2014.

[7] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[8] 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*, no. 1, pp. 37–41, Feb. 2012.

[9] W. Shi, Q. Ling, G. Wu, and W. Yin, "A Proximal Gradient Algorithm for Decentralized Composite Optimization," *IEEE Trans. on Signal Process.*, pp. 1–11, 2015.

[10] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multiuser systems," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 641–656, February 2014.

[11] T.-H. Chang, A. Nedic, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1524–1538, June 2014.

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

[13] H. Ouyang and A. Gray, "Fast stochastic frank-wolfe algorithms for non-linear svms," in *SDM - SIAM International Confernce on Data Mining*, 2010.

[14] F. R. Bach, "Learning with submodular functions: A convex optimization perspective," *Foundations and Trends in Machine Learning*, vol. 6, no. 2-3, pp. 145–373, 2013.

[15] M. Fukushima, "A modified frank-wolfe algorithm for solving the traffic assignment problem," *Transportation Research Part B: Methodological*, vol. 18, no. 2, pp. 169–177, April 1984.

[16] M. Dudk, Z. Harchaoui, and J. Malick, "Lifted coordinate descent for learning with trace-norm regularization," in *AISTATS*, Mar 2012.

[17] M. Jaggi, "Revisiting frank-wolfe: Projection-free sparse convex optimization," in *ICML*, vol. 28, no. 1, June 2013, pp. 427–435.

[18] R. M. Freund and P. Grigas, "New analysis and results for the frank-wolfe method," *ArXiv e-prints*, July 2013.

[19] A. Bellet, Y. Liang, A. B. Garakani, M.-F. Balcan, and F. Sha, "A Distributed Frank-Wolfe Algorithm for Communication-Efficient Sparse Learning," pp. 1–19, 2014. [Online]. Available: http://arxiv.org/abs/1404.2644

[20] M. Jaggi, V. Smith, M. Takac, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan, "Communication-efficient distributed dual coordinate ascent," in *NIPS*, 2014.

[21] D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[22] J. Lafond, H.-T. Wai, and E. Moulines, "Convergence analysis of a stochastic projection-free algorithm," *ArXiv e-prints*, 2015. [Online]. Available: http://arxiv.org/abs/1510.01171

[23] P. Massart, *Concentration Inequalities and Model Selection*. Springer, 2003.

[24] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip Algorithms for Distributed Signal Processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[25] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, Sep. 2004.

[26] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[27] R. A. Horn and C. R. Johnson, *Topics in matrix analysis*. Cambridge: Cambridge University Press, 1994, corrected reprint of the 1991 original.