This is the supplementary document to the paper: "Parallel and Distributed Block-Coordinate Frank-Wolfe Algorithms".

A. Convergence analysis

We provide a self-contained convergence proof in this section. The skeleton of our convergence proof follow closely from Lacoste-Julien et al. (2013) and Jaggi (2013). There are a few subtle modification and improvements that we need to add due to our weaker definition of approximate oracle call that is nearly correct only in expectation. The delayed convergence is new and interesting for the best of our knowledge, which uses a simple result in "load balancing" (Mitzenmacher, 2001).

Note that for the cleanness of the presentation, we focus on the primal and primal-dual convergence of the version of the algorithms with pre-defined step sizes and additive approximate subroutine, it is simple to extend the same analysis for line-search variant and multiplicative approximation.

A.1. Primal Convergence

Lemma 8. *Denote the gap between current $f(x^{(k)})$ and the optimal $f(x^*)$ to be $h(x^{(k)})$. The iterative updates in Algorithm 1(with arbitrary fixed stepsize γ or by the line search) obey*

$$\mathbb{E}h(x^{(k+1)}) \leq (1 - \frac{\gamma\tau}{n})\mathbb{E}h(x^{(k)}) + \frac{\gamma^2(1+\delta)}{2}C_f^\tau.$$

where the expectation is taken over the joint randomness all the way to iteration $k+1$.

Proof. Let $x := x^{(k)}$ for notational convenience. We prove the result for Algorithm 1 first. Apply the definition of $C_f^{(S)}$ and then apply the definition of the additive approximation in (5), to get

$$\begin{aligned} f(x^{(k+1)}_{\mathrm{line-search}}) \leq f(x_\gamma^{(k+1)}) &= f(x + \gamma\sum_{i\in S}(s_{[i]} - x_{[i]})) \\ &\leq f(x) + \gamma\sum_{i\in S}\langle s_{[i]} - x_{[i]}, \nabla_{[i]}f(x)\rangle + \frac{\gamma^2}{2}C_f^{(S)} \\ &= f(x) + \gamma\langle s_{[S]} - x_{[S]}, \nabla_{[S]}f(x)\rangle + \frac{\gamma^2}{2}C_f^{(S)} \end{aligned}$$

Subtract $f(x^*)$ on both sides we get:

$$h(x^{(k+1)}) \leq h(x^{(k)}) + \gamma\langle s_{[S]} - x_{[S]}^{(k)}, \nabla_{[S]}f(x^{(k)})\rangle + \frac{\gamma^2}{2}C_f^{(S)}$$

Now take the expectation over the entire history then apply (5) and definition of the surrogate duality gap (6), we obtain

$$\begin{aligned} \mathbb{E}h(x^{(k+1)}) \leq& \mathbb{E}h(x^{(k)}) + \mathbb{E}\left\{\gamma\langle s_{[S]} - x_{[S]}^{(k)}, \nabla_{[S]}f(x^{(k)})\rangle\right\} + \mathbb{E}\frac{\gamma^2}{2}C_f^{(S)} \\ =& \mathbb{E}h(x^{(k)}) + \gamma\mathbb{E}\left\{\langle s_{[S]}, \nabla_{[S]}f(x^{(k)})\rangle - \min_{s\in\mathcal{M}^{(S)}}\langle s, \nabla_{[S]}f(x^{(k)})\rangle\right\} \\ &- \gamma\mathbb{E}\left\{\langle x_{[S]}^{(k)}, \nabla_{[S]}f(x^{(k)})\rangle - \min_{s\in\mathcal{M}^{(S)}}\langle s, \nabla_{[S]}f(x^{(k)})\right\} + \frac{\gamma^2}{2}C_f^\tau \\ \leq& \mathbb{E}h(x^{(k)}) + \frac{\gamma^2\delta}{2}C_f^\tau - \gamma\mathbb{E}_{x^k}\mathbb{E}_{S|x^k}\sum_{i\in S}g^{(i)}(x^{(k)}) + \frac{\gamma^2}{2}C_f^\tau \\ =& \mathbb{E}h(x^{(k)}) + \frac{\gamma^2\delta}{2}C_f^\tau - \gamma\mathbb{E}_{x^k}\frac{\tau}{n}g(x^{(k)}) + \frac{\gamma^2}{2}C_f^\tau \qquad (10) \\ \leq& (1 - \frac{\gamma\tau}{n})\mathbb{E}h(x^{(k)}) + \frac{\gamma^2(1+\delta)}{2}C_f^\tau. \end{aligned}$$

The last inequality follows from the property of the surrogate duality gap $g(x^{(k)}) \geq h(x^{(k)})$ due to the fact that $g(x) = f(x) - f^*(\cdot)$. This completes the proof of the descent lemma. □

Now we are ready to state the proof for Theorem 2.

Proof of Theorem 2. We follow the proof in Theorem C.1 in (Lacoste-Julien et al., 2013) to prove the statement for Algorithm 1. The difference is that we use a different and carefully chosen sequence of step size.

Take $C = h_0 + n(1+\delta)C_f^\tau$, and denote $\mathbb{E}h(x^{(k)})$ as h_k for short hands. The inequality in Lemma 8 simplifies to

$$h_{k+1} \leq \left(1 - \frac{\gamma\tau}{n}\right) h_k + \frac{\gamma^2}{2n}C.$$

Now we will prove $h_k \leq \frac{2nC}{\tau^2 k+2n}$ for $\gamma_k = \frac{2n\tau}{\tau^2 k+2n}$ by induction. The base case $k=0$ is trivially true since $C > h_0$. Assuming that the claim holds for k, we apply the induction hypothesis and the above inequality is reduced to

$$\begin{aligned} h_{k+1} &\leq (1 - \frac{\gamma\tau}{n})h_k + \frac{\gamma^2}{2n}C \leq \frac{2nC}{\tau^2 k + 2n}\left[1 - \frac{\gamma\tau}{n} + \frac{\tau^2 k + 2n}{2n}\frac{\gamma^2}{2n}\right] \\ &= \frac{2nC}{\tau^2 k + 2n}\left[\frac{\tau^2 k + 2n}{\tau^2 k + 2n} - \frac{2n\tau}{\tau^2 k + 2n}\cdot\frac{\tau}{n} + \frac{(2n\tau)^2}{4n^2(\tau^2 k + 2n)}\right] \\ &= \frac{2nC}{\tau^2 k + 2n}\cdot\frac{\tau^2 k + 2n - \tau^2}{\tau^2 k + 2n} \leq \frac{2nC}{\tau^2 k + 2n}\cdot\frac{\tau^2 k + 2n - \tau^2 + \tau^2}{\tau^2 k + 2n + \tau^2} \\ &= \frac{2nC}{\tau^2(k+1) + 2n}. \end{aligned}$$

This completes the induction and hence the proof for the primal convergence for Algorithm 1. □

A.2. Convergence of the surrogate duality gap

Proof of Theorem 3. We mimic the proof in (Lacoste-Julien et al., 2013, Section C.3) for the analogous result closely, and we will use the same notation for $h_k$and C as in the proof for primal convergence, moreover denote $g_k = \mathbb{E}g(x^{(k)})$ First from (10) in the proof of Lemma 8, we have

$$h_{k+1} \leq h_k - \frac{\gamma\tau}{n}g_k + \frac{\gamma^2}{2n}C.$$

Rearrange the terms, we get

$$g_k \leq \frac{n}{\gamma\tau}(h_k - h_{k+1}) + \frac{\gamma C}{2\tau}. \tag{11}$$

The idea is that if we take an arbitrary convex combination of $\{g_1, ..., g_K\}$, the result will be within the convex hull, namely between the minimum and the maximum, hence proven the existence claim in the theorem. By choosing weight $\rho_k := k/S_K$ where normalization constant $S_K = \frac{K(K+1)}{2}$ and taking the convex combination of both side of (11), we have

$$\begin{aligned} \mathbb{E}(\min_{k\in[K]} g_k) &\leq \sum_{k=0}^{K}\rho_k g_k \leq \frac{n}{\tau}\sum_{k=1}^{K}\rho_k(\frac{h_k}{\gamma_k} - \frac{h_{k+1}}{\gamma_k}) + \sum_{k=0}^{K}\rho_k\gamma_k\frac{C}{2\tau} \\ &= \frac{n}{\tau}(\frac{h_0\rho_0}{\gamma_0} - h_{K+1}\frac{\rho_k}{\gamma_k}) + \frac{n}{\tau}\sum_{k=0}^{K-1}h_{k+1}(\frac{\rho_{k+1}}{\gamma_{k+1}} - \frac{\rho_k}{\gamma_k}) + \sum_{k=0}^{K}\rho_k\gamma_k\frac{C}{2\tau} \\ &\leq \frac{n}{\tau}\sum_{k=0}^{K-1}h_{k+1}(\frac{\rho_{k+1}}{\gamma_{k+1}} - \frac{\rho_k}{\gamma_k}) + \sum_{k=0}^{K}\rho_k\gamma_k\frac{C}{2\tau} \end{aligned} \tag{12}$$

Note that $\rho_0 = 0$, so we simply dropped a negative term in last line. Applying the step size $\gamma_k = 2n\tau/(\tau^2 k + 2n)$, we get

$$\begin{aligned} \frac{\rho_{k+1}}{\gamma_{k+1}} - \frac{\rho_k}{\gamma_k} &= \frac{k+1}{S_K}\frac{\tau^2(k+1)2n}{2n\tau} - \frac{k}{S_K}\frac{\tau^2 k + 2n}{2n\tau} \\ &= \frac{1}{2nS_K\tau}\left[\tau^2(k+1)^2 + 2n(k+1) - \tau^2 k^2 - 2nk\right] \\ &= \frac{\tau^2(2k+1) + 2n}{2nS_K\tau}. \end{aligned}$$

Plug the above back into (12) and use the bound $h_{k+1} \leq 2nC/(\tau^2(k+1)+2n)$, we get

$$\begin{aligned}
\mathbb{E}(\min_{k\in[K]} g_k) &\leq \sum_{k=0}^{K} \rho_k g_k \leq \frac{nC}{\tau^2 S_K} \sum_{k=0}^{K-1} \frac{\tau^2(2k+2)+2n}{2n} \frac{2n}{\tau^(2k+1)+2n} + \sum_{k=0}^{K} \frac{k}{S_K} \frac{2n\tau}{\tau^2 k+2n} \frac{C}{2\tau} \\
&= \frac{nC}{\tau^2 S_K} \left[\sum_{k=0}^{K-1} (1 + \frac{\tau^2}{\tau^2(k+1)+2n}) + \sum_{k=1}^{K} \frac{k\tau^2}{(\tau^2 k+2n)} \right] \\
&\leq \frac{nC}{\tau^2 S_K} [2K+K] = \frac{2nC}{\tau^2(K+1)} \cdot 3.
\end{aligned}$$

This completes the proof for $K \geq 1$. □

Proof of Convergence with Delayed Gradient The idea is that we are going to treat the updates calculated from the delayed gradients as an additive error and then invoke our convergence results that allow the oracle to be approximate. We will first present a lemma that we will use for the proof of Theorem 6.

Lemma 9. *Let $x \in \mathcal{M}$, $\|\cdot\|$ be a norm, $\mathrm{Diam}(\mathcal{M})_{\|\cdot\|} \leq D$, L be the gradient Lipschitz constant of f with respect to the given norm $\|\cdot\|$. Moreover, let x' be at most κ steps away from x and the largest stepsize in the past κ steps, and*

$$\begin{aligned}
x^* &:= \operatorname*{argmin}_{s\in\mathcal{M}} \langle s, \nabla f(x) \rangle \\
\tilde{x} &:= \operatorname*{argmin}_{s\in\mathcal{M}} \langle s, \nabla f(x') \rangle
\end{aligned}$$

Then, we have

$$\langle \tilde{s} - x, \nabla f(x) \rangle \leq \langle s^* - x, \nabla f(x) \rangle + \gamma\kappa D^2 L$$

Proof. Because $\tilde{s}$ minimizes $\langle s, \nabla f(\tilde{x}) \rangle$ over $s \in \mathcal{M}$ and s^* is feasible, we can write

$$\langle s^* - \tilde{s}, \nabla f(\tilde{x}) \rangle \geq 0.$$

Using this and Hölder's inequality, we can write

$$\langle \tilde{s} - x, \nabla f(x) \rangle - \langle s^* - x, \nabla f(x) \rangle \leq \langle \tilde{s} - s^*, \nabla f(x) - \nabla f(\tilde{x}) \rangle \leq \|\tilde{s} - s^*\| \|\nabla f(\tilde{x}) - \nabla f(x)\|_* \leq DL\|\tilde{x} - x\|.$$

It remains to bound $\|\tilde{x} - x\|$.

$$\|\tilde{x} - x\| = \left\| \tilde{x} - \tilde{x} - \sum_{i=1}^{\kappa} \gamma_{-i}(s_{-i} - x_{-i}) \right\| \leq \gamma\kappa \max_i \|s_{-i} - x_{-i}\| \leq \gamma\kappa D,$$

where we used the fact that x is at most κ steps away from $\tilde{x}$. Assume γ_{-i} is the stepsize used and $\langle s_{-i}, x_{-i} \rangle$ are the actual updates that had been performed in the nearest ith parameter update before we get to x. □

The second lemma that we need is the following.

Lemma 10. *Let $\mathcal{M}$ be a convex set. Let $x_0 \in \mathcal{M}$. Let m be any positive integer. For $i = 1, ..., m$, let $x_i = x_{i-1} + \gamma_i(s_i - x_{i-1})$ for some $0 \leq \gamma_i \leq 1$ and $s_i \in \mathcal{M}$. Then there exists an $s \in \mathcal{M}$ and $\gamma \leq \sum_{i=1}^m \gamma_i$, such that $x_m = \gamma(s - x_0) + x_0$.*

Proof. We prove by induction. When $m = 1$, $s = s_1$ and $\gamma = \gamma_1$. Assume for any $m = k-1$, that the claim holds assume the condition is true, then by the recursive formula,

$$\begin{aligned}
x_k &= x_{k-1} + \gamma_k(s_k - x_{k-1}) \\
&= x_0 + \gamma(s - x_0) + \gamma_k[s_k - x_0 - \gamma(s - x_0)] \\
&= x_0 - (\gamma + \gamma_k - \gamma_k\gamma)x_0 + (\gamma - \gamma_k\gamma)s + \gamma_k s_k \\
&= x_0 + (\gamma + \gamma_k - \gamma_k\gamma) \left[\frac{\gamma - \gamma_k\gamma}{\gamma + \gamma_k - \gamma_k\gamma} s + \frac{\gamma_k}{\gamma + \gamma_k - \gamma_k\gamma} s_k - x_0 \right] \\
&= x_0 + (\gamma + \gamma_k - \gamma_k\gamma)(s' - x_0)
\end{aligned}$$

Note that s' is a convex combination of s_k and s therefore by convexity $s' \in \mathcal{M}$. Substitute $\gamma \leq \sum_{i=1}^{k-1} \gamma_i$, we get

$$\gamma + \gamma_k - \gamma_k \gamma \leq \sum_{i=1}^{k} \gamma_i.$$

This completes the inductive proof for all m. □

The third Lemma that we will need is the following characterization of the expected "max load" in randomized load balancing.

Lemma 11 ((Mitzenmacher, 2001; Raab & Steger, 1998)). *Suppose m balls are thrown independently and uniformly at random into n bins. Then, the maximum number of balls in a bin Y satisfies*

$$\mathbb{E}Y \leq \begin{cases} \frac{3\log n}{\log(n/m)} & \textit{if } m < n/\log n, \\ c' \log n & \textit{if } m < cn\log n, \\ \frac{m}{n} + O(\sqrt{\frac{2m}{n}} \log n) & \textit{if } m \gg n \log n. \end{cases}$$

where c' is a constant that depends only on c.

Proof of Theorem 6. The proof involves a sharpening of the Lemma 9 for the BCFW and minibatch setting, where $x \in \mathcal{M} = \mathcal{M}^{(1)} \times ... \times \mathcal{M}^{(n)}$ is a product domain. The proof idea is to exploit this property. Let the current update be on coordinate block index subset S. For each $j \in S$, let the corresponding worker be delayed by $\varkappa_j$ steps, and the corresponding parameter vector be $\tilde{x}$. $\varkappa_j$ is a random variable.

As in the proof of Lemma 9, we can bound the suboptimality of the approximate subroutine for solving problem j:

$$\begin{aligned} \text{Suboptimality}(\tilde{s}_j) &\leq \langle \tilde{s}_j - s_j^*, \nabla_j f(\tilde{x}) - \nabla_j f(x) \rangle \leq \|\tilde{s}_j - s_j^*\| \|\nabla_j f(\tilde{x}) - \nabla_j f(x)\|_* \\ &\leq D^{(j)}_{\|\cdot\|} L^{(j)}_{\|\cdot\|} \|\tilde{x} - x\| = D^{(j)}_{\|\cdot\|} L^{(j)}_{\|\cdot\|} \left\| \sum_{i=1}^{\varkappa_j} \gamma_{-i}(s_{-i} - x_{-i}) \right\| \\ &\leq D^{1}_{\|\cdot\|} L^{1}_{\|\cdot\|} \sum_{i=1}^{\varkappa_j} \gamma_{-i} \left\| (s_{-i} - x_{-i}) \right\| \\ &\leq D^{1}_{\|\cdot\|} L^{1}_{\|\cdot\|} \sum_{i=1}^{\varkappa_j} \gamma_{-i} D^{\tau}_{\|\cdot\|} \leq \varkappa_j \gamma_{-\varkappa_j} D^{1}_{\|\cdot\|} L^{1}_{\|\cdot\|} D^{\tau}_{\|\cdot\|}. \end{aligned} \tag{13}$$

Let $\kappa := \mathbb{E}\varkappa_j$, take expectation on both sides we get

$$\mathbb{E} \text{ Suboptimality}(\tilde{s}_j) \leq \mathbb{E}(\varkappa_j \gamma_{-\varkappa_j}) D^{1}_{\|\cdot\|} L^{1}_{\|\cdot\|} D^{\tau}_{\|\cdot\|}$$

Repeat the same argument for each $i \in S$, we get

$$\mathbb{E} \text{ Suboptimality}(\tilde{s}) \leq \mathbb{E}(\varkappa_j \gamma_{-\varkappa_j}) \tau D^{1}_{\|\cdot\|} L^{1}_{\|\cdot\|} D^{\tau}_{\|\cdot\|}.$$

To put it into the desired format in (5), we solve the following inequality for δ

$$\frac{\gamma \delta C_f^{\tau}}{2} \geq \mathbb{E}(\varkappa_j \gamma_{-\varkappa_j}) \tau D^{\tau}_{\|\cdot\|} D^{1}_{\|\cdot\|} L^{1}_{\|\cdot\|}$$

we get

$$\delta \geq \frac{2\tau}{C_f^{\tau}} \mathbb{E}\left(\frac{\varkappa_j \gamma_{-\varkappa_j}}{\gamma} \right) D^{\tau}_{\|\cdot\|} D^{1}_{\|\cdot\|} L^{1}_{\|\cdot\|}.$$

By the specification of the stepsizes, we can calculate for each k,

$$\frac{\gamma_{-\varkappa_j}}{\gamma} = \frac{\tau^2 k + 2n}{\tau^2(\max(k - \varkappa_j, 0)) + 2n}.$$

Note that we always enforce $\varkappa_j$ to be smaller than $\frac{k}{2}$ (otherwise the update is dropped), we can therefore upper bound $\mathbb{E}(\frac{\varkappa_j \gamma_{-\varkappa_j}}{\gamma})$ by 2κ. This gives us the the first bound (9) on δ in Theorem 6.

To get the second bound on δ, we start from (13) and bound $\|\tilde{x} - x\|$ differently. Let S be the set of $\tau\varkappa_j$ coordinate blocks that were updated in the past $\varkappa_j$ iterations. In the cases where fewer than $\tau\varkappa_j$ blocks were updated, just arbitrarily pick among the coordinate blocks that were updated 0 times so that $|S| = \tau\varkappa_j$. $\tilde{x} - x$ is supported only on S. Suppose coordinate block $i \in S$ is updated by m times, as below

$$\tilde{x}_{(i)} = \sum_{j=1}^{m} \gamma_j (s_j - [x_j]_{(i)})$$

for some sequence of $0 \leq \gamma_1, ..., \gamma_m \leq 1$ and $s_1, ..., s_m \in \mathcal{M}_i$ and recursively $[x_j]_{(i)} = [x_{j-1}]_{(i)} + \gamma_j(s_j - [x_{j-1}]_{(i)})$ ($x_0 = x$). Apply Lemma 10 for each coordinate block, we know that there exist $s_{(i)} \in \mathcal{M}_i$ in each block $i \in S$ such that

$$\tilde{x}_{(i)} = x_{(i)} + \gamma_{(i)}(s_{(i)} - x_{(i)})$$

with

$$\gamma_{(i)} \leq \sum_{j \in \text{ iterations where } i \text{ is updated}} \gamma_j \leq m\gamma_{\max}. \tag{14}$$

Note that $s_{(i)} \in \mathcal{M}_i$ for each $i \in S$ implies that their concatenation $s_{(S)} \in \mathcal{M}_S$. Also $\gamma_{\max} \leq \gamma_{-\varkappa_j}$. Therefore

$$\|\tilde{x} - x\| = \left\| \sum_{i \in S} \gamma_{(i)}(s_{(i)} - x_{(i)}) \right\| \leq m\gamma_{\max} \|s_{(S)} - x_{(S)}\| \leq Y \gamma_{-\varkappa_j} D_{\|\cdot\|}^{\tau\varkappa_j}$$

where Y is a random variable that denotes the number of updates received by the most updated coordinate block (the maximum load). Apply a previously used argument to get $\gamma_{-\varkappa_j} < 2\gamma$, take expectation on both sides, to get the following by the law of total expectations and (14)

$$\begin{aligned} \mathbb{E}\|\tilde{x} - x\| &\leq \mathbb{E}\left(Y\gamma_{-\varkappa_j} D_{\|\cdot\|}^{\tau\varkappa_j}\right) = \mathbb{E}\left[\mathbb{E}\left(Y\gamma_{-\varkappa_j} D_{\|\cdot\|}^{\tau\varkappa_j} \middle| \varkappa_j\right)\right] = \mathbb{E}\left[\gamma_{-\varkappa_j} D_{\|\cdot\|}^{\tau\varkappa_j} \mathbb{E}\left(Y|\varkappa_j\right)\right] \\ &\leq 2\gamma \mathbb{E}\left[D_{\|\cdot\|}^{\tau\varkappa_j} \mathbb{E}\left(Y|\varkappa_j\right)\right] \end{aligned} \tag{15}$$

This expectation is taken over the entire history of minibatch choice and delay associated with each update. When we condition on $\varkappa_j$, the conditional expectation of Y becomes the load-balancing problem.

By Lemma 11 when $\kappa_{\max}\tau \leq \frac{n}{\log n}$, it follows from (15) that

$$\mathbb{E}\|\tilde{x} - x\| \leq 2\gamma \mathbb{E} D_{\|\cdot\|}^{\varkappa_j \tau} \frac{3\log n}{\log(n/\varkappa_j\tau)} \leq \frac{3\log n}{\log[n/(\tau\kappa_{\max})]} 2\gamma \mathbb{E} D_{\|\cdot\|}^{\varkappa_j\tau}.$$

When $\kappa_{\max}\tau < cn\log n$,

$$\mathbb{E}\|\tilde{x} - x\| \leq 2\gamma \mathbb{E} D_{\|\cdot\|}^{\varkappa_j\tau} O(\log n) \leq O(\log n) 2\gamma \mathbb{E} D_{\|\cdot\|}^{\varkappa_j\tau}.$$

When $\kappa_{\max}\tau \gg n\log n$, then

$$\mathbb{E}\|\tilde{x} - x\| \leq (1 + o(1)) \frac{\tau\kappa_{\max}}{n} 2\gamma \mathbb{E} D_{\|\cdot\|}^{\varkappa_j\tau}.$$

Repeating the above results for each block $j \in S$, and summing them up leads to an upper bound for $\frac{\gamma\delta C_f^{\tau}}{2}$ and the proof of (9) is complete by solving for δ. □

B. Proofs of other technical results

Relationship of the curvatures.

Proof of Lemma 1. $C_f^{(S)} \leq C_f$ follows from the fact that

$$\langle y_{(S)} - x_{(S)}, \nabla_{(S)} f(x) \rangle = \langle y_{[S]} - x_{[S]}, \nabla f(x) \rangle,$$

and $s_{[S]} \in \mathcal{M}$. In other words, the $\arg\sup$ of (3) is a feasible solution in the $\sup$ to compute the global C_f. Similar argument holds for the proof $C_f^{(i)} \leq C_f^{(S)}$ as $i \in S$.

In the second part,

$$C_f^\tau = \frac{1}{\binom{n}{\tau}} \sum_{T \subset [n], |T| = \tau} C_f^{(T)}.$$

We can evenly partition sets T in the summation into n parts P_j for $j \in [n]$, such that sets in P_j have the element j. Clearly each P_j has a size of $\binom{n}{\tau}/n$. We can use $C_f^{(S)} \geq C_f(j)$ from the first inequality of the lemma, to get the inequality below.

$$C_f^\tau = \frac{1}{\binom{n}{\tau}} \sum_{j \in [n]} \sum_{T \in P_j} C_f^{(T)} \geq \frac{1}{\binom{n}{\tau}} \sum_{j \in [n]} \sum_{T \in P_j} C_f^{(j)} = \frac{1}{\binom{n}{\tau}} \sum_{j \in [n]} \binom{n}{\tau} \frac{1}{n} C_f^{(j)} = \frac{1}{n} C_f^{\otimes}$$

The relaxation of C_f^τ to C_f is trivial since $C_f^{(T)} \leq C_f$ holds for any $T \subseteq [n]$ from the first part of the lemma. □

Bounding C_f^τ using expected boundedness and expected incoherence

Proof of Theorem 4. By Definition of H, for any $x, z \in \mathcal{M}$, $\gamma \in [0, 1]$

$$f(x + \gamma(z - x)) \leq f(x) + \gamma(z - x)^T \nabla f(x) + \frac{\gamma^2}{2}(z - x)^T H (z - x).$$

Rearranging the terms we get

$$\frac{2}{\gamma^2}\left[f(x + \gamma(z - x)) - f(x) - \gamma(z - x)^T \nabla f(x)\right] \leq (z - x)^T H (z - x)$$

The definition of set curvature (3) is written in an equivalent notation with $z = x_{[S^c]} + s_{[S]}$ and $y = x + \gamma(z - x) = x + \gamma(s_{[S]} - x_{[S]})$. So we know the support of $z - x$ is constrained to be within the coordinate blocks S.

Plugging this into the definition of (3) we get an analog of Equation (2.12) in (Jaggi, 2011) for $C_f^{(S)}$.

$$\begin{aligned}
C_f^{(S)} &= \sup_{\substack{x, z \in \mathcal{M}, \gamma \in [0,1] \\ z_{(S^c)} - x_{(S^c)} = 0}} \frac{2}{\gamma^2}\left[f(x + \gamma(z - x)) - f(x) - \gamma(z - x)^T \nabla f(x)\right] \\
&\leq \sup_{\substack{x, z \in \mathcal{M}, \\ z_{(S^c)} - x_{(S^c)} = 0}} (z - x)^T H (z - x) = \sup_{\substack{x, z \in \mathcal{M}, \\ z_{(S^c)} - x_{(S^c)} = 0}} s_{(S)}^T H s_{(S)} \\
&\leq \sup_{w \in \mathcal{M}^{(S)}} (2w^T) H_S (2w) = 4 \left\{ \sup_{w_i \in \mathcal{M}^{(i)} \forall i \in S} \sum_{i \in S} w_i^T H_{ii} w_i + \sum_{i, j \in S, i \neq j} w_i^T H_{ii} w_j \right\} \\
&\leq 4 \left\{ \sum_{i \in S} \sup_z \sup_{w_i} w_i^T H_{ii}(z) w_i + \sum_{i, j \in S, i \neq j} \sup_z \sup_{w_i, w_j} w_i^T H_{ii}(z) w_j \right\} \\
&\leq 4(\sum_{i \in S} B_i + \sum_{i, j \in S, i \neq j} \mu_{ij}).
\end{aligned}$$

Take expectation for all possible S of size τ and we obtain the lemma statement. □

Proof of the example with sublinear dependence of κ

Proof of Lemma 7. We first show that a continuous extension of $D^{\theta}_{\|\cdot\|}$ is concave in θ

$$\begin{aligned} D^{\theta}_{\|\cdot\|} &= \max_{S\subset[n]||S|=\theta} \sup_{x,y\in\mathcal{M}^{(S)}} \|x-y\| \\ &= \max_{S\subset[n]||S|=\theta} \sup_{x,y\in\mathcal{M}^{(S)}} \sqrt{\sum_{i\in S}\|x_{(i)}-y_{(i)}\|^2} \\ &= \sqrt{\max_{S\subset[n]||S|=\theta} \sum_{i\in S} \sup_{x_{(i)},y_{(i)}\in\mathcal{M}^{(i)}} \|x_{(i)}-y_{(i)}\|^2} \end{aligned}$$

The supremum is obtained by sorting and the function in the square root is concave function of θ, when we extend the support of this function to $\mathbb{R}_+$ through linear interpolation. By the composition theorem, the square root of that is also a concave function in τ. We call this function $\tilde{D}^{\theta}_{\|\cdot\|}$. Note that $\tilde{D}^{\theta}_{\|\cdot\|} = D^{\theta}_{\|\cdot\|}$ when $\theta \in [n]$ such that if we take expectation over the any discrete distribution over θ, their expectations are the same. It follows from Jensen's inequality that

$$\begin{aligned} \mathbb{E}D^{\varkappa\tau}_{\|\cdot\|} &= \mathbb{E}\tilde{D}^{\varkappa\tau}_{\|\cdot\|} \leq \tilde{D}^{\mathbb{E}\varkappa\tau}_{\|\cdot\|} \\ &\leq D^{\lceil\mathbb{E}\varkappa\tau\rceil}_{\|\cdot\|} \\ &= \sqrt{\max_{S\subset[n]\big||S|=\lceil\mathbb{E}\varkappa\tau\rceil} \sum_{i\in S} \sup_{x_{(i)},y_{(i)}\in\mathcal{M}^{(i)}} \|x_{(i)}-y_{(i)}\|^2} \\ &\leq \sqrt{\lceil\mathbb{E}\varkappa\rceil} D^{\tau}_{\|\cdot\|}. \end{aligned}$$

□

Proof of specific examples

Proof of Example 1. First of all, $H = \lambda A^T A$. Since all columns of A have the same magnitude $\sqrt{2}/n$. By the Holder's inequality and the 1-norm constraint in every block, we know $B_i = \frac{2}{n^2\lambda}$ for any i therefore $B = \frac{2}{n^2\lambda}$. Secondly, by well-known upper bound for the area of the spherical cap, which says for any fixed vector z and random vector a on a unit sphere in $\mathbb{R}^d$,

$$\mathbb{P}(|\langle z,a\rangle| > \epsilon\|z\|) \leq 2e^{\frac{-d\epsilon^2}{2}},$$

we get

$$\mathbb{P}(\mu_{ij} > 2\sqrt{\frac{20\log d}{d}}) \leq \frac{2}{d^{10}}.$$

Take union bound over all pairs of labels we get the probability as claimed. □

Proof of Example 2. The matrix $D^T D$ is tridiagonal with 2 on the diagonal and -1 on the off-diagonal. If we vectorize U by concatenating $u = [u_1; ...; u_{n-1}]$, the Hessian matrix for u will be $H = \Pi I_d \otimes (D^T D)\Pi^T$ where Π is some permutation matrix. Without calculating it explicitly, we can express

$$\begin{aligned} &u_S^T H_S u_S = u_S^T (D^T \otimes 1_d)(D^T \otimes 1_d)^T u_S \\ =&\sum_{i\in S} u_i^T \begin{bmatrix} D^T_{:,i} \\ D^T_{:,i} \\ \vdots \\ D^T_{:,i} \end{bmatrix} \begin{bmatrix} D_{:,i} & D_{:,i} & \dots & D_{:,i} \end{bmatrix} u_i + \sum_{i,j\in S, i\neq j} u_i^T \begin{bmatrix} D^T_{:,i} \\ D^T_{:,i} \\ \vdots \\ D^T_{:,i} \end{bmatrix} \begin{bmatrix} D_{:,j} & D_{:,j} & \dots & D_{:,j} \end{bmatrix} u_j. \end{aligned}$$

We note that for any $|i-j| \geq 2$, the second term is 0. Apply the constraint that $\|u_i\|_2 \leq \lambda$ and the fact that the ℓ_2 operator norm of $\begin{bmatrix} D_{:,j} & D_{:,j} & \dots & D_{:,j} \end{bmatrix}$ is $\sqrt{2d}$, we get $B_i = 2\lambda^2 d$. Similarly, $2(n-2)$ nonzero obeys $\mu_{ij} = \lambda^2 d$. This allows us to obtain an upper bound

$$C_f^{\tau} \leq 4\left[2\tau\lambda^2 d + \frac{2(n-2)\tau(\tau-1)}{(n-2)(n-1)}\lambda^2 d\right] \leq 16\tau\lambda^2 d.$$

which scales with τ. □

B.1. Pseudocode for the Multicore Shared Memory Architecture

We present pseudocode for the multicore shared memory setting here. It is the same except that each worker becomes a thread, the network buffer of servers become the a data structure, the workers' network buffer becomes the shared parameter vector and the workers can write to the data structure or the shared parameter vector directly.

Algorithm 2 AP-BCFW: Asynchronous Parallel Block-Coordinate Frank-Wolfe (Shared memory)

```
————————————SERVER THREAD———————————
Input: An initial feasible $x^{(0)}$, mini-batch size $\tau$, number of workers $T$.
0.  Write $x^{(0)}$ to shared memory. Declare a container (a queue or a stack).
for $k$ = 1,2,... ($k$ is the iteration number.) do
  1.  Keep popping the container until we have $\tau$ updates on $\tau$ disjoint blocks. Denote the index set by $S$.
  2.  Set step size $\gamma = \frac{2n\tau}{\tau^2 k+2n}$.
  3.  Write sparse updates $x^{(k)} = x^{(k-1)} + \gamma \sum_{i \in S}(s_{[i]} - x_{[i]}^{(k-1)})$ into the shared memory.
  if converged then
    Broadcast STOP signal to all threads and break.
  end if
end for
Output: $x^{(k)}$.
———————————WORKER THREADS———————————
while no STOP signal received do
  a. Randomly choose $i \in [n]$.
  b. Calculate partial gradient $\nabla_{(i)} f(x)$ using $x$ in the shared memory and solve (2).
  c. Push $\{i, s_{(i)}\}$ to the container.
end while
```

The above pseudo code can be further simplified when $\tau = 1$. In particular, we do not need a server any more. Each worker can simply write to the shared memory bus. The probability of two workers writing to the same block is small as we analyzed in Section D.2. The algorithm essentially lock-free as in (Niu et al., 2011) modulo requiring the updates of each coordinate block to be atomic. Niu et al. (2011) is stronger in that it allows each scalar addition to be atomic.

There is an additional restriction due to the fixed predefined sequence of step sizes, which in fact requires a centralized shared counter that is atomic, so that no two threads have simultaneously the same k. In practice, we can simply choose a fixed sequence of stepsize for each worker separately.

Algorithm 3 AP-BCFW: Asynchronous Parallel Block-Coordinate Frank-Wolfe (Lock-Free Shared-Memory)

```
Input: An initial feasible $x^{(0)}$, number of workers $T$, a centralized counter.
0.  Write $x^{(0)}$ to shared memory.
—————INDEPENDENTLY ON EACH THREAD—————
while not converged do
  a. Randomly choose $i \in [n]$.
  b. Calculate partial gradient $\nabla_{(i)} f(x)$ using $x$ in the shared memory and solve (2).
  c. Read centralized counter for $k$. Set step size $\gamma = \frac{2n}{k+2n}$.
  d. Add $\gamma(s_{(i)} - x_{(i)})$ to block $i$ of the shared memory.
  e. Increment the counter $k = k + 1$.
end while
——————————————————————————————————-
if converged then
  Output: $x^{(k)}$. and break.
end if
```

C. Application to Structural SVM

We briefly review structural SVMs and show how to solve the associated convex optimization problem using our AP-BCFW method.

In structured prediction setting, the task is to predict a structured output $\mathbf{y} \in \mathcal{Y}$, given $\mathbf{x} \in \mathcal{X}$. For example, $\mathbf{x}$ could be the pixels in the picture of a word, $\mathbf{y}$ could be the sequence of characters in the word. A feature map $\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ encodes compatibility between inputs and outputs. A linear classifier parameter $\mathbf{w}$ is learned from data so that $\text{argmax}_{\mathbf{y}\in\mathcal{Y}}\langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y})\rangle$ gives the output for an input $\mathbf{x}$. Suppose we have the training data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ to learn $\mathbf{w}$. Define $\psi_i(\mathbf{y}) := \phi(\mathbf{x}_i, \mathbf{y}_i) - \phi(\mathbf{x}_i, y)$ and let $L_i(\mathbf{y}) := L(\mathbf{y}_i, \mathbf{y})$ denote the loss incurred by predicting $\mathbf{y}$ instead of the correct output $\mathbf{y}_i$. The classifier parameter $\mathbf{w}$ is learned by solving the optimization problem

$$\begin{aligned} &\min_{\mathbf{w},\xi} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i \\ &\text{s.t } \langle \mathbf{w}, \psi_i(\mathbf{y})\rangle \geq L(\mathbf{y}_i, \mathbf{y}) - \xi_i \qquad \forall i, \mathbf{y} \in \mathcal{Y}(\mathbf{x}_i). \end{aligned} \tag{16}$$

We solve the dual of this problem using our method. We introduce some more notation to formulate the dual. Denote $\mathcal{Y}_i := \mathcal{Y}(\mathbf{x}_i)$, the set of possible labels for $\mathbf{x}_i$. Note that $|\mathcal{Y}_i|$ is exponential in the length of label $\mathbf{y}_i$. Let $m = \sum_{i=1}|\mathcal{Y}_i|$. Let $A \in \mathbb{R}^{d\times m}$ denote a matrix whose m columns are given by $\{\frac{1}{\lambda n}\psi_i(\mathbf{y}) \mid i \in [n], \mathbf{y} \in \mathcal{Y}_i\}$. Let $b \in \mathbb{R}^m$ be a vector given by the entries $\{\frac{1}{n}L_i(\mathbf{y}) \mid i \in [n], \mathbf{y} \in \mathcal{Y}_i\}$. The dual of (16) is given by

$$\begin{aligned} &\min_{\alpha\in\mathbb{R}^m} f(\alpha) := \frac{\lambda}{2}\|A\alpha\|^2 - b^T\alpha \\ &\text{s.t} \sum_{\mathbf{y}\in\mathcal{Y}_i}\alpha_i(\mathbf{y}) = 1 \qquad \forall i \in [n], \alpha \geq 0 \end{aligned} \tag{17}$$

The primal solution $\mathbf{w}$ can be retrieved from the dual solution α from the relation $\mathbf{w} = A\alpha$ obtained from KKT conditions. Also note that the domain $\mathcal{M}$ of (17) is exactly the product of simplices $\mathcal{M} = \Delta_{|\mathcal{Y}_1|} \times \cdots \times \Delta_{|\mathcal{Y}_n|}$.

The subproblem in equation (2) takes a well-known form in the Frank-Wolfe setup for solving (17). The gradient is given by

$$\nabla f(\alpha) = \lambda A^T A\alpha - b = \lambda A^T \mathbf{w} - b$$

whose $(i, \mathbf{y})$-th component is given by $\frac{1}{n}\left(\langle \mathbf{w}, \psi_i(\mathbf{y})\rangle - L_i(\mathbf{y})\right)$. Define $H_i(\mathbf{y}; \mathbf{w}) := L_i(\mathbf{y}) - \langle \mathbf{w}, \psi_i(\mathbf{y})\rangle$ so that the $(i, \mathbf{y})$-th component of the gradient is $-\frac{1}{n}H_i(\mathbf{y}; \mathbf{w})$. In the subproblem (2), the domain $\mathcal{M}^{(i)}$ is the simplex $\Delta_{\mathcal{Y}_i}$ and the block gradient $\nabla_{(i)}f(\alpha)$ is linear. So, the objective is minimized at a corner of the simplex $\mathcal{M}^{(i)}$ and the optimum value is simply given by $\min_{\mathbf{y}} \nabla_{(i)}f(\alpha)$ which can be rewritten as $\max_{\mathbf{y}} H_i(\mathbf{y}; \mathbf{w})$. Further, the corner can be explicitly written as the indicator vector $e^{\mathbf{y}_i^*} \in \mathcal{M}^{(i)}$ where $\mathbf{y}_i^* = \text{argmax}_{\mathbf{y}} H_i(\mathbf{y}; \mathbf{w})$. It turns out that this maximization problem can be solved efficiently for several problems. For example, when the output is a sequence of labels, a dynamic programming algorithm like Viterbi can be used.

As mentioned before, m is too large to update the dual variable α directly. So, we make an update to the primal variable $\mathbf{w} = A\alpha$ instead. The Block-Coordinate Frank-Wolfe update for the i-th block maybe written as $\alpha_{(i)}^{k+1} = \alpha_i^k + \gamma(s_i - \alpha_{(i)}^k)$ where γ is the step-size. Recalling that the optimal s_i is $e^{\mathbf{y}_i^*}$, by multiplying the previous equation by A_i, we arrive at $\mathbf{w}_i^{(k+1)} = \mathbf{w}_i^k + \gamma(A_{i,y_i^*} - \mathbf{w}_i^{(k)})$ where $\mathbf{w}_i^{(k)} := A_i\alpha_{(i)}$. From this definition of $\mathbf{w}_i^{(k)}$, the primal update is obtained by noting that $\mathbf{w}^{(k)} = \sum_i \mathbf{w}_i^{(k)}$. Explicitly, the primal update is given by $\mathbf{w}^{(k+1)} = \mathbf{w}^k + \gamma(A_{i,y_i^*} - \mathbf{w}_i^{(k)})$. Note that $A_{i,y_i^*} = \frac{1}{\lambda n}\psi_(\mathbf{y}_i^*)$. This Block-Coordinate version can be easily extended to AP-BCFW. In our shared memory implementation, for OCR dataset, we do the line search computation and $\mathbf{w}_i^{(k)}$ update step on the workers instead of the server because these computations turn out to be expensive enough to make the server the bottleneck even for modest number of workers.

D. Other technical results and discussions

D.1. Oracle assumption and heterogeneous blocks

Recall that our results rely on the oracle assumption that $\mathcal{O}$ provides updates that are iid uniform over $[n]$ (Assumption A1). We discuss the implications and limitations of this assumption and then propose possible solutions.

Consider the setting where $\mathcal{O}$ consists of T possibly heterogeneous workers and each worker samples iid from $[n]$. As we discussed before, A1 holds under the additional condition that the time needed to complete one subroutine solve for Block i by Worker j does not depend on i.

Consider the simple example due to an anonymous reviewer: Let $\tau = 1, T = 2$ and there are a total of two blocks. Block 1 takes only a millisecond an Block 2 takes a year to solve for both workers. In this case, the first update received by the server is with probability $3/4$ for Block 1 and only $1/4$ for Block 2.

This could potentially limit the use of our parallel algorithm for applications such as structured predictions where sentences having different lengths, or cases where there are different sparsity level over data points/constraints depending on how we formulate the problem.

This is in fact not a problem unique to us, Assumption A1 is implicitly required in most existing analysis for asynchronous stochastic algorithms (e.g., Liu et al., 2014; Niu et al., 2011). As a result, they all share the same woe. One could argue that parallelization is the wrong problem to address when block subroutines significantly differ with each other. Efforts should be spent on perhaps solving the expensive subproblem in parallel. But still, even mild heterogeneity over blocks invalidates our convergence result.

Henceforth, we propose two simple ways to address this issue and discuss their pros and cons.

Padding: A naive solution is to per-calculate the time-complexity with respect to each block and inject artificial time padding on each user such that all blocks have the same time complexity.

Pre-select S: An alternative is to let the server randomly choose a coordinate subset S of size τ, and the workers can only work on S, either by independently sample from S or work on whichever that is not available.

Neither of the two solutions is completely satisfactory. The padding approach ensures all results in the paper to hold including those for the delayed oracles, but inevitably, the time to complete each block now depends on the most expensive block. The second approach has milder dependence on the worst block, in fact it depends only on the time for the fastest worker to solve the slowest problem in each chosen S. However, it requires sending an updated parameter to all workers in every iteration. It could still be robust to heterogeneous workers when τ is several times larger than T, and when workers work asynchronously within the mini-batch, we prove in Proposition 12 that the number of collisions is small.

Fully asynchronous parallelism over blocks with heterogeneous blocks without dependence on the slowest block remains an important open problem.

D.2. Controlling collisions in distributed setting

In the distributed setting, different workers might end up working on the same slot.

In Algorithm 1, different workers may end up working on the same coordinate block and the server will drop a number of updates in case of collision. The following proposition shows that for this potential redundancy is not excessive is small and for a large range of τ, we also show additional strong concentration to its mean.

Proposition 12. *In the distributed asynchronous update scheme above:*

i) The expected number of subroutine calls from all workers to complete each iteration is $\tau + \sum_{i=1}^{\tau-1} \frac{i}{n-i}$.

ii) If $0.02n < \tau < 0.6n$*, with probability at least* $1 - \exp(-n/60)$*, no more than* 2τ *random draws (*2τ *subroutine calls in total from all workers) suffice to complete each iteration.*

Proof. The first claim is the well-known coupon collector problem.

The second claim requires an upper bound of the expectation. In expectation, we need $\frac{n}{n-k}$ balls to increase the unique count from k to $k+1$. So in expectation we need

$$1 + \frac{n}{n-1} + \frac{n}{n-2} + ... + \frac{n}{n-\tau+1} = \tau + \sum_{i=1}^{\tau-1} \frac{i}{n-i}$$
$$\leq \tau + \frac{1+2+\cdots+(\tau-1)}{n-\tau+1} = \tau + \frac{\tau(\tau-1)}{2(n-\tau+1)} < \tau\left[1 + \frac{1}{2(n/\tau-1)}\right].$$

To see the second claim, first defined f_t to be the number of non-empty bins after t random ball throws, which can be consider as a function of the t iid ball throws $X_1, X_2, ..., X_t$. It is clear that if we change only one of the X_i, f_t can be changed by at most 1. Also, note that the probability that any one bin being filled is $1-(1-\frac{1}{n})^t$, so $\mathbb{E}f_t = n\left[1-\left(1-\frac{1}{n}\right)^t\right]$.

By the McDiarmid's inequality, $\mathbb{P}\left[f_t < \mathbb{E}f_t - \epsilon\right] \leq \exp\left[-\frac{2\epsilon^2}{t}\right]$. Take $t = 2\tau$, and $\epsilon = \mathbb{E}f_{2\tau} - \tau$, then

$$\mathbb{P}\left[f_{2\tau} < \tau\right] \leq \exp\left[-\frac{1}{\tau}\left(n\left[1-\left(1-\frac{1}{n}\right)^{2\tau}\right]-\tau\right)^2\right] \leq \exp\left[-\frac{1}{\tau}\left(n\left[1-e^{-\frac{2\tau}{n}}\right]-\tau\right)^2\right]$$
$$= \exp\left[-n \cdot \frac{n}{\tau}\left(1-e^{-\frac{2\tau}{n}}-\frac{\tau}{n}\right)^2\right] \leq \exp\left[-Cn\right],$$

where C is some constant which is the smaller of the two evaluations of the function $\frac{n}{\tau}\left(1-e^{-\frac{2\tau}{n}}-\frac{\tau}{n}\right)^2$ at $\tau = 0.02n$ and $\tau = 0.6n$ (where the function is concave between the two). As a matter of fact, C can be taken as $\frac{1}{60}$.

Let g_τ be the number of balls that one throws that fills τ bins, the result is proven by noting that

$$\mathbb{P}(g_\tau \leq 2\tau) = \mathbb{P}(f_{2\tau} \geq \tau) \geq 1 - \exp\left[-Cn\right].$$

□

D.3. Curvature and Lipschitz Constant

In this section, we illustrate the relationship between the coordinate curvature constant, coordinate gradient Lipschitz conditions, and work out the typical size of the constants in Theorem 6. For the sake of discussion, we will focus on the quadratic function $f(x) = \frac{x^TAx}{2} + bTx$. We start by showing that for quadratic function. The constant that one can get via choosing a specific norm can actually match the curvature constant. To be completely explicit, we define gradient Lipschitz constant $L_{\|\cdot\|}$ with respect to a norm $\|\cdot\|$, this requires that for any x, y,

$$\|\nabla f(y) - \nabla f(x)\|_* \leq L\|x-y\|.$$

where $\|\cdot\|_*$ is the dual norm.

Proposition 13. *For quadratic functions with Hessian $A \succeq 0$, there exists a norm $\|\cdot\|$ such that the curvature constant $C_f = [D_{\|\cdot\|}]^2 L_{\|\cdot\|}$.*

Proof. We will show that this norm is simply the A-norm, $\|\cdot\|_A = \sqrt{(\cdot)^TA(\cdot)}$. The upper bound $C_f \leq [D_{\|\cdot\|_A}]^2 L_{\|\cdot\|_A}$ is a direct application of the result in Jaggi (2013, Appendix D). To show a lower bound it suffices to construct $s, x \in \mathcal{M}, \gamma \in [0,1]$ and $y = \gamma s + (1-\gamma)x$ such that

$$\frac{2}{\gamma^2}(f(y) - f(x) - \langle y-x, \nabla f(x)\rangle) = [D_{\|\cdot\|_A}]^2 L_{\|\cdot\|_A}.$$

For quadratic functions,

$$\frac{2}{\gamma^2}[f(y) - f(x) - \langle y-x, \nabla f(x)\rangle] = \frac{1}{2}(y-x)^TA(y-x) = \frac{1}{\gamma^2}\|y-x\|_A^2$$

Take $\gamma = 1$ and y, x on the boundary of $\mathcal{M}$ such that $\|y - x\|_A = D_{\|\cdot\|_A}$, as a result, we get $C_f \geq D_{\|\cdot\|_A}]^2$. It remains to show that the gradient Lipschitz constant with respect to $\mathcal{A}$-norm is 1, which directly follows from the Taylor expansion. □

Similar arguments work for $C_f^{(i)}$ and $C_f^{(S)}$ under the same norm. Clearly, this means that the corresponding restriction of the subset domain has $A_{i,i}$-norm or $A_{(S)}$-norm.

We now consider the approximation constants due to the delays in Theorem 6, and work out more explicit bounds for quadratic functions and carefully chosen norm. Recall that the simple bound (8) has constant δ in the order of

$$\frac{\kappa\tau L^1_{\|\cdot\|} D^1_{\|\cdot\|} D^\tau_{\|\cdot\|}}{C_f^\tau}.$$

Suppose we use the A-norm, then $L^1_{\|\cdot\|} = L^\tau_{\|\cdot\|} = 1$, and $C_f^\tau = [D^\tau_{\|\cdot\|}]^2$, the bound can be reduced to

$$\delta = O(\frac{\tau D^1_{\|\cdot\|}}{D^\tau_{\|\cdot\|}}) = O(\kappa\sqrt{\tau}).$$

where the last step requires $\mathcal{M}_i$ to be all equivalent and A to be block-diagonal with identical $A_{(i)}$.

Similarly the strong bound (9) has constant δ in the order of

$$\delta = \tilde{O}\left(\frac{\tau L^1_{\|\cdot\|} D^1_{\|\cdot\|} D^{\kappa\tau}_{\|\cdot\|}}{C_f^\tau}\right) = \tilde{O}\left(\frac{\tau D^1_{\|\cdot\|} D^{\kappa\tau}_{\|\cdot\|}}{[D^\tau_{\|\cdot\|}]^2}\right) = \tilde{O}(\sqrt{\kappa\tau})$$

Again, the last step requires a strong assumption that $\mathcal{M}_i$ to be all equivalent and A to be block-diagonal with identical diagonal blocks. While these calculations only apply to specific case of a quadratic function with a lot of symmetry, we conjecture that in general the flexibility of choosing the norm will allow the ratio of these boundedness constants and C_f^τ to be a well-controlled constant and the typical dependence on the system parameter τ and κ should stay within the same ball park.

D.4. Examples and illustrations

In this section, we now derive specific instances of the Theorem 4 for the structural SVM and Group Fused Lasso. For the structural SVM, a simple generalization of Lacoste-Julien et al. (2013, Lemmas A.1, A.2) shows that in the worst case, using $\tau > 1$ offers no gain at all. Fortunately, if we consider more specific problems, using larger τ does yield faster convergence. We provide two such examples below.

Example 1 (**Structural SVM for multi-label classification (with random data)).** We describe the application to structural SVMs in detail in Section C (please see this section for details on notation). Here, we describe the convergence rate for this application. According to (Yu & Joachims, 2009), the compatibility function $\phi(x, y)$ for multiclass classification will be $[0, ..., 0, x^T, 0, ...0]^T/\lambda n$ where the only nonzero block that we fill with the feature vector is the (y)th block. So $\psi_i(x_i, j) = \phi(x_i, y_i) - \phi(x_i, j)$ looks like $[0, ..., 0,\ x_i^T, 0\ , ...0,\ -x_i^T, 0,\ ...0]^T/\lambda n$. This already ensures that $B = \frac{2}{n^2\lambda}$ provided x_i lie on a unit sphere. Suppose we have K classes and each class has a unique feature vector drawn randomly from a unit sphere in $\mathbb{R}^d$; furthermore, for simplicity assume we always draw $\tau < K$ data points with τ distinct labels[4]$\mu \leq \sqrt{\frac{c \log d}{d}} \frac{2}{n^2\lambda}$, for some constant c. In addition, if $d \geq \tau^2\sqrt{c \log d}$, then with high probability

$$C_f^\tau \leq \frac{8\tau + 8\tau^2\sqrt{\frac{c\log d}{d}}}{n^2\lambda} = O\left(\frac{c\tau}{n^2\lambda}\right),$$

which yields a convergence rate $O(\frac{R^2}{\lambda\tau k})$, where $R :=$ $\max_{i\in[n], y\in\mathcal{Y}_i} \|\psi_i(y)\|_2$ using notation from Lemmas A.1 and A.2 of Lacoste-Julien et al. (2013).

This analysis suggests that a good rule-of-thumb is that we should choose τ to be at most the number of categories for the classification. If each class is a mixture of random draws from the unit sphere, then we can choose τ to be the underlying number of mixture components.

[4] This is an oversimplification but it offers a rough rule-of-thumb. In practice, C_f^τ should be in the same ballpark as our estimate here.

Example 2 (Group Fused Lasso). The Group Fused Lasso aims to solve (typically for $q = 2$)

$$\min_X \quad \tfrac{1}{2}\|X - Y\|_F^2 + \lambda\|XD\|_{1,q}, \qquad q > 1, \tag{18}$$

where $X, Y \in \mathbb{R}^{d\times n}$, and column y_t of Y is an observed noisy d-dimensional feature vector at time $1 \le t \le n$. The matrix $D \in \mathbb{R}^{n\times(n-1)}$ is the differencing matrix that takes the difference of feature vectors at adjacent time points (columns). The formulation aims to filter the trend that has some piecewise constant structures. The dual to (18) is

$$\begin{aligned} &\max_U \; -\frac{1}{2}\|UD^T\|_F^2 + \mathrm{tr}UD^TY^T \\ &\text{s.t.} \quad \|U_{:,t}\|_p \le \lambda, \; \forall t = 1, ..., n-1, \end{aligned}$$

where p is conjugate to q, i.e., $1/p + 1/q = 1$. This block-constrained problem fits our structure (1). For this problem, we find that $B \le 2\lambda^2 d$ and $\mu \le 2\lambda^2 d/(n-1)$, which yields

$$C_f^\tau \le 16\tau\lambda^2 d.$$

Consequently, the rate of convergence becomes $O(\frac{n^2\lambda^2 d}{\tau k})$. In this case, batch FW will have a better rate of convergence than BCFW[5].

Example 3 (Structural SVM worst-case bound)**.** For structural SVM with arbitrary data (including even pathological/trivial data), using notation from Lemmas A.1 and A.2 of Lacoste-Julien et al. (2013), define $R := \max_{i\in[n],y\in\mathcal{Y}_i} \|\psi_i(y)\|_2$. Then we can provide an upper bound

$$B, \mu \le \frac{R^2}{\lambda n^2} \quad \Longrightarrow \quad C_f^\tau \le \frac{4\tau^2 R^2}{\lambda n^2}. \tag{19}$$

In this case, for any $\tau = 1, ..., n$, the rate of convergence will be the same $O(\frac{R^2}{\lambda k})$.

An illustration for the group fused lasso Figure 7 shows a typically application for group fused lasso (filtering piecewise constant multivariate signals whose change poitns are grouped together).

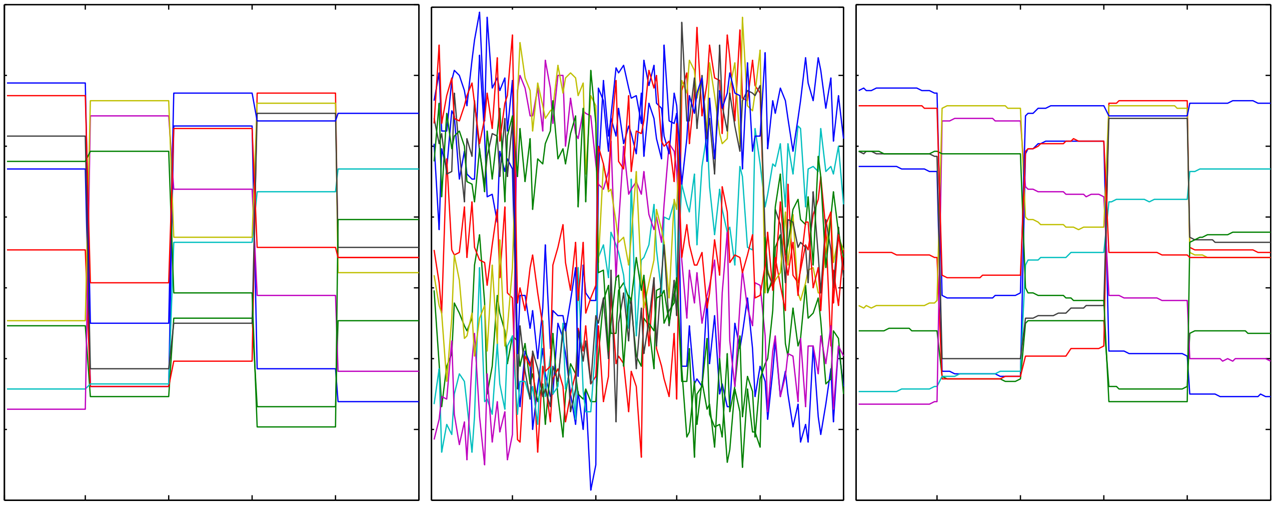

Figure 7. Illustration of the signal data used in the Fused Lasso experiments. We show the original signal (left), the noisy signal given to the algorithm (middle), and the signal recovered after performing the fused lasso optimization (right).

[5] Observe that C_f^τ does not have an n^2 term in the denominator to cancel out the numerator. This is because the objective function is not appropriately scaled with n like it does in the structural SVM formulation.

D.5. Comparison to parallel block coordinate descent

With some understanding on C_f^τ, we can now explicitly compare the rate of convergence in Theorem 2 with parallel BCD (Liu et al., 2014; Richtárik & Takáč, 2015) under the assumption of $\mu = O(B/\tau)$ — a fair and equally favorable case to all of these methods. We acknowledge that more general treatments of ESO property in more recent extensions of Richtárik & Takáč (2015) in a similar flavor as our (7) (see e.g., Qu & Richtárik 2014) but similar results are not available for the asynchronous version. To facilitate comparison, we will convert the constants in all three methods to block coordinate gradient Lipschitz constant L_i, which obeys

$$f(x + s_{[i]}) \leq f(x) + \langle s_{[i]}, \nabla f(x) \rangle + L_i \|s_{[i]}\|^2, \tag{20}$$

for any $x \in \mathcal{M}, s_{(i)} \in \mathcal{M}_i$. Observe that $B_i \leq 4L_i \mathrm{diam}(\mathcal{M}_i)^2 = L_i \max_{x_i^*, x_i \in \mathcal{M}_i} \|x_i - x_i^*\|^2$, so

$$B \leq \frac{1}{n} \sum_i L_i \max_{x_i, x_i^*} \|x_i - x_i^*\| \tag{21}$$

$$\leq \frac{1}{n} \sum_i L_i \max_x \|x - x^*\|^2 = \mathbb{E}_i(L_i) R^2 \tag{22}$$

where $R := \max_x \|x - x^*\|$. The rate of convergence for the three methods (with τ oracle calls considered as one iteration) are given below.

Method	Rate
AP-BCFW (Ours)	$O_p\left(\frac{n\mathbb{E}_i(L_i)R^2}{\tau k}\right)$
P-BCD[6]	$O_p\left(\frac{n\mathbb{E}_i(L_i)R^2}{\tau k}\right)$
AP-BCD[7]	$O_p\left(\frac{n \max_i L_i R^2}{\tau k}\right)$

The comparison illustrates that these methods have the same $O(1/k)$ rate and almost the same dependence on n and τ despite the fact that we use a much simpler linear oracle. Nothing comes for free though: Nesterov acceleration does not apply for Frank-Wolfe based methods in general, while a careful implementation of parallel coordinate descents can achieve $O(1/k^2)$ rate without any full-vector interpolation in every iteration (Fercoq & Richtárik, 2015). Also, Frank-Wolfe methods usually need additional restrictive conditions or algorithmic steps to get linear convergence for strongly convex problems.

These facts somewhat limits the applicability of our method to cases when projection can be computed as efficiently as (2). However, as is surveyed in (Jaggi, 2013), there are many interesting cases when (2) is much cheaper than projections, e.g., projection onto a nuclear norm ball takes $O(n^3)$ while (2) takes only $O(n^2)$.

Lastly, we note that in the fully asynchronous setting, we obtained an exponential improvement on the dependence of delay comparing to that in (Liu et al., 2014). It is unclear whether this is a unique property of the block-coordinate Frank-Wolfe algorithm or similar results can be obtained for projection based block-coordinate descent.

[6] In (Richtárik & Takáč, 2015, Theorem 19)

[7] In (Liu et al., 2014, Theorem 3)