Communication-Efficient Projection-Free Algorithm for Distributed Optimization

Yan Li¹, Chao Qu², and Huan Xu¹

¹H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology
²Department of Electrical Engineering, Technion

May 22, 2018

Abstract

Distributed optimization has gained a surge of interest in recent years. In this paper we propose a distributed projection free algorithm named Distributed Conditional Gradient Sliding (DCGS). Compared to the state-of-the-art distributed Frank-Wolfe algorithm, our algorithm attains the same communication complexity under much more realistic assumptions. In contrast to the consensus based algorithm, DCGS is based on the primal-dual algorithm, yielding a modular analysis that can be exploited to improve linear oracle complexity whenever centralized Frank-Wolfe can be improved. We demonstrate this advantage and show that the linear oracle complexity can be reduced to almost the same order of magnitude as the communication complexity, when the feasible set is polyhedral. Finally we present experimental results on Lasso and matrix completion, demonstrating significant performance improvement compared to the existing distributed Frank-Wolfe algorithm.

1 Introduction

Decentralized optimization methods have been widely used in the machine learning community. Compared to centralized optimization methods, they enjoy several advantages, including aggregating the computing power of distributed machines, robustness to dynamic network topologies, and preserving data privacy (Yuan et al., 2016). These advantages make them an attractive option when data are collected by distributed agents, and either communicating the data to a central processing agent is computational prohibitive, or due to privacy concern each agent needs to keep the local data privately.

After the seminal work in Tsitsiklis (1985); Tsitsiklis et al. (1986), there have been fruitful development on distributed optimization. For non-smooth objective functions, consensus-based subgradient methods have been analyzed in Nedic and Ozdaglar (2009); Sundhar Ram et al. (2010); Nedic (2011). Dual averaging method proposed in (Duchi et al., 2012) further explains the effect of the network topology on the convergence. Usually distributed subgradient methods converge at the rate of $O\left(\frac{\log(T)}{\sqrt{T}}\right)$. For smooth (and possibly strongly convex) objective, Shi et al. (2015) propose an exact first-order algorithm EXTRA and its proximal variant, which attain an improved rate of $O\left(\frac{1}{T}\right)$ for general smooth objective, and a linear rate of $O(e^{cT})$ for smooth and strongly convex objective.
with \( c < 1 \) (see also [Nedich et al. (2016); Yuan et al. (2016)]). Asynchronous decentralized (sub)gradient descent algorithms have also been proposed and analyzed [Nedic (2011); Tsiatsiklis et al. (1986)]. Another class of mainstream distributed algorithms are based on the dual method, which include the classic idea of dual decomposition [Tegelius et al. (2014)] and the celebrated alternating direction method of multipliers (ADMM) [Boyd et al. (2011); Makhdoumi and Ozdaglar (2017); Wei and Ozdaglar (2013)]. ADMM attains \( O(\frac{1}{T}) \) convergence rate for the smooth problem and \( O(c^T) \) for smooth and strongly convex problem, but such results rely on strong assumptions such as having no constraints [Shi et al. (2014)] or the local subproblem of each agent being easy to solve [Wei and Ozdaglar (2013)].

All the aforementioned methods can be categorized as projection-based methods, as they all require to take projection back to the feasible set of the constraints at each iteration. Though commonly assumed to be easy, in numerous applications such projection indeed can either be computational expensive (projection onto the trace norm ball, base polytopes [Fujishige and Isotani, (2011)] or even intractable [Collins et al., 2008]). Frank-Wolfe (FW) algorithm arises as a natural alternative in these scenarios. Unlike projection-based methods, FW assumes a linear oracle (LO) that solves a linear optimization problem over the feasible set which can be significantly easier than the projection. We refer to algorithms that avoid projection as projection-free algorithms. FW algorithm has been revisited in recent years for its projection-free property [Jaggi, 2013; 2011; Clarkson, 2010; Bach et al., 2012] and numerous improvements have been made. These include regularized FW [Harchaoui et al., 2015], linearly convergent variants under additional assumptions [Garber and Hazan, 2013; Lacoste-Julien and Jaggi, 2015], and stochastic variants [Hazan and Kale, 2012; Hazan and Lund, 2016].

Related Work. Despite the progress on centralized FW algorithm, results on distributed FW algorithm are surprisingly limited. Specialized versions of decentralized FW algorithm have been proposed. Wang et al. (2016) propose a distributed block-coordinate FW algorithm for block-separable feasible sets [Lacoste-Julien et al., 2013]. Bellet et al. (2013) consider a Lasso-type distributed learning problem. They neither assume nor exploit the fact that the global objective is a natural summation of each agent’s local objective, and their communication scheme is also different from the usual “within neighborhood” communication scheme and could be significantly more complicated. Lafond et al. (2016) consider a distributed FW algorithm also for the Lasso-type problem that leverages the sparsity of iterates to improve communication overhead. To the best of our knowledge, the most recent distributed FW (DFW) algorithm on general smooth convex problems is by [Wai et al. (2017)], wherein the DFW convergence rate is \( O(\frac{1}{T}) \) for smooth objectives; and \( O(\frac{1}{T^2}) \) for smooth and strongly convex objective under the assumption that the minimizer lies in the interior of constraint set. This assumption is almost unrealistic for two reasons: it implies the problem is essentially unconstrained, which usually fails to impose structural properties (such as sparsity, low-rankness) to the solution; and for a unconstrained problem the vanilla distributed gradient descent algorithm suffices to solve the problem efficiently [Yuan et al. (2016)]. Whether such restrictive assumption could be removed while retaining the same complexity remains an open question.

We should note that all the previously discussed methods share the same communication complexity and projection/LO complexity to obtain an \( \epsilon \)-optimal solution, regardless of being projection-based or projection-free. In practice, however, the time consumed by a single communication and a LO/projection often differ by orders of magnitude, which could incur significant latency. Modern CPUs perform IO at over 10 GB/s yet communication over TCP/IP is about 10 MB/s, this gap is even more significant when LO oracle
is already cheap. Consider the matrix completion problem, in Section 5 we will show that for an iteration consisting one round of communication and one LO, communication would take up over 97\% time. This implies in DFW the actual running time would be largely consumed by communication. To alleviate the problem of latency in communication expensive applications, whether it is possible to trade for a better communication complexity with a moderately worse LO complexity becomes another open question.

Contributions. In this paper we answer the above mentioned questions with an (almost) affirmative answer. Our contributions are the following:

- We propose a new distributed projection-free algorithm named Decentralized Conditional Gradient Sliding (DCGS), and show that it attains $O\left(\frac{1}{\epsilon}\right)$ communication complexity and $O\left(\frac{1}{\epsilon^2}\right)$ LO complexity for smooth convex objectives.

- Without assuming the minimizer being in the interior of the feasible set, we show that DCGS attains $O\left(\sqrt{\frac{1}{\epsilon}}\right)$ communication complexity and $O\left(\frac{1}{\epsilon^2}\right)$ LO complexity for smooth and strongly convex objectives.

- Our algorithm builds upon a distributed version of primal-dual algorithm and is hence modular. As a consequence, improvement on centralized FW can be easily exploited by DCGS. We demonstrate this advantage when the feasible set is polyhedral, for smooth and convex objective the LO complexity can be reduced to $\tilde{O}\left(\frac{1}{\epsilon}\right)$, while for smooth and strongly convex objective the LO complexity can be further reduced to $\tilde{O}\left(\sqrt{\frac{1}{\epsilon}}\right)$, which matches the result of (Wai et al., 2017), but without the restrictive assumption on the minimizer.

2 Problem formulation

We consider an undirected graph $G = (V, E)$, where $V = \{1, \ldots, m\}$ denotes the vertex set and $E \subset V \times V$ denotes the edge set. Each node $i \in V$ is associated with an agent indexed also by $i$, and has its local objective $f_i(x) : \mathbb{R}^d \to \mathbb{R}$. We define the neighborhood of agent $i$ to be $N(i) = \{ j \in V : (i, j) \in E \}$. Each agent $i$ can only communicate information with its neighbors. Naturally, $G$ is assumed to be connected or otherwise distributed optimization is impossible. Our objective is to minimize the summation of the local objectives, subject to the constraint that $x$ belongs to a closed compact convex set $X$, that is:

\[
\text{Minimize: } f(x) := \sum_{i=1}^{m} f_i(x) .
\]

We assume each function $f_i$ is $u$ (possibly 0)-strongly convex and $l$-smooth, i.e., $\frac{u}{2} \| y - x \|^2 \leq f_i(y) - f_i(x) - \nabla f_i(x)^T (y - x) \leq \frac{l}{2} \| y - x \|^2$. Our algorithm could also be easily adapted to the setting where $f_i$ has different smoothness and strong convexity. We present here only the homogeneous case for simplicity.

The distributed formulation can be reformulated as the following linearly constrained optimization problem. Consider each agent $i$ keeps its local copy of decision variable $x_i$, we can impose a linear constraint on $x = (x_1, \ldots, x_m)$ so that $x_i = x_j$ for all

\footnote{Throughout this paper we use $\tilde{O}()$ to hide any additional logarithmic factor}
Define the graph Laplacian $L \in \mathbb{R}^{m \times m}$ to be:

$$L_{ij} = \begin{cases} |N(i)| & \text{if } i = j, \\ -1 & \text{if } i \neq j \text{ and } (i, j) \in E, \\ 0 & \text{otherwise}. \end{cases}$$

Then (1) could be reformulated as:

$$\min_{x \in X^m} F(x) = \sum_{i=1}^{m} f_i(x_i) \quad \text{s.t.} \quad Lx = \mathbf{0},$$

Let $\otimes$ denotes the Kronecker product, here $L = L \otimes I_d$ and $X^m = \{(x_1, \ldots, x_m) : x_i \in X\}$. Since $G$ is assumed to be connected, (1) and (3) are equivalent.

We can further reformulate the linearly constrained problem as a bilinear saddle point problem. Observe that (3) is equivalent to:

$$\min_{x \in X^m} \max_{y \in \mathbb{R}^{md}} \langle Lx, y \rangle.$$ 

The bilinear saddle point problem (1) is well suited for the primal-dual algorithm proposed in (Chambolle and Pock, 2011). We present the orginal primal-dual algorithm applied to our problem in Algorithm 1.

Algorithm 1 Primal-dual algorithm (Chambolle and Pock, 2011)

1. Let $x^0 = x^{-1} \in X^m$ and $y^0 \in \mathbb{R}^{md}$ and $\{\alpha_k\}, \{\tau_k\}, \{\eta_k\}$ be given.
2. for $k = 1, \ldots, N$ do
3. $\tilde{x}^k = \alpha_k(x^{k-1} - x^{k-2}) + x^{k-1}$
4. $y^k = \arg \min_{y \in \mathbb{R}^{md}} \langle -Lx^k, y \rangle + \frac{\tau_k}{2} \|y - y^{k-1}\|^2$
5. $x^k = \arg \min_{x \in X^m} \langle Ly^k, x \rangle + F(x) + \frac{\eta_k}{2} \|x^{k-1} - x\|^2$
6. end for

Lan et al. (2017) observe that since $F(x)$ is a summation splitted across agents, all the updates in the primal-dual algorithm can be performed in a distributed way. They propose a distributed primal-dual algorithm and show that to find an $\epsilon$-optimal solution, one needs $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$ rounds of communication for a non-smooth convex objective and $\mathcal{O}(\frac{1}{\epsilon})$ for a non-smooth strongly convex objective, which improves upon the previous results. However, their algorithm still lies in the category of projection-based algorithms and they consider non-smooth problem, which is different from our setting.

3 Algorithm

In this section we present in Algorithm 3 the Decentralized Conditional Gradient Sliding for a general convex feasible set equipped with a linear oracle. At a high-level, DCGS is closely related to Conditional Gradient Sliding (CGS) algorithm proposed in Lan and Zhou (2016). However CGS considers only the primal problem, here we consider a primal-dual problem due to performing distributed optimization. As such, the analysis is significantly more involved.
The most important step of DCGS algorithm is in Line 8 where we update decision variable $x^k_i$ by calling the CG procedure defined in Line 12. If we define $\phi^k_i(x_i) = \langle w^k_i, x_i \rangle + f_i(x_i) + \frac{\eta_k}{2} \left\| x_i - x^{k-1}_i \right\|^2$, then the CG procedure could be seemed as the FW algorithm applied to $\min_{x_i \in X} \phi^k_i(x_i)$, with termination criterion $\langle \nabla \phi^k_i(x^k_i), x^k_i - x^{k-1}_i \rangle \leq e^k_i$, where the left hand side is often called the Wolfe gap. Below we make a few remarks on the communication mechanism, the main technical challenges and the modularity of our algorithm.

**Communication Mechanism.** At each outer iteration $k$, each local agent first computes $\tilde{x}^k_i$ based on extrapolation of two previous primal iterates, and broadcast $\tilde{x}^k_i$ to all of its neighbors $j \in N(i)$. After one round of broadcasting, each agent uses $\tilde{x}^k_j$ received from its neighbors and perform dual variable update $y^k_j$, then broadcast the updated dual variable to all of its neighbors $j \in N(i)$. After second round of broadcasting, each agent uses $y^k_j$ received from its neighbors and call the CG procedure to update primal variable $x^{k+1}_i$. Each iteration incurs two rounds of communication within the network, hence the overall communication complexity is the same as the outer iteration complexity.

**Trade-off between Communication and LO.** If we set $e^k_i = 0$ in Line 8 we are solving problem $\min_{x_i \in X} \phi^k_i(x_i)$ exactly. The outer iteration of DCGS then reduces to the primal-dual algorithm applied to our problem (11), implemented in a distributed fashion. By well-known results of the primal-dual algorithm (Chambolle and Pock, 2011; Lan et al., 2017), to get an $\epsilon$-optimal solution one needs $O(\frac{1}{\epsilon})$ iterations for a convex objective and $O(\frac{1}{\sqrt{\epsilon}})$ iterations for a strongly convex objective. From our previous discussion, this yields the communication complexity of DCGS to be $O(\frac{1}{\epsilon})$ and $O(\frac{1}{\sqrt{\epsilon}})$ respectively. However in
this extreme case LO calls in the CG procedure would be prohibitively large. Consequently, we need to carefully choose $e_i^k$ to ensure that the subproblem $\min_{x_i \in X} \phi_i^k(x_i)$ is solved in a controlled way: the convergence of outer iteration should be approximately at the same speed as the case when the subproblem is solved exactly, but meanwhile we need to keep LO complexity in the CG procedure to remain relatively small.

Modularity. The CG procedure in DCGS algorithm is where all the calls to LO take place. We believe there are not much room for improvement in terms of the communication complexity, as our complexity in the general case matches that of the DFW algorithm under additional (very strong) assumption that the optimal solution is in the interior of the feasible set. The room for improvement then lies in possibly reducing LO complexity. If we treat the CG procedure as a module in the DCGS algorithm, can we replace it with a module that runs much faster for specific objectives or feasible sets, and obtain a better DCGS variant? The answer is affirmative. As an example we will show that significant improvement on LO complexity could be made when the feasible set is polyhedral.

4 Theoretical Results

4.1 General Feasible Set

In this section we set suitable parameters to DCGS for convex and strongly convex objectives. We will present its convergence results, communication and LO complexity. We also present a detailed comparison with results of DFW in [Wai et al., 2017].

**Theorem 1** (Convergence for Smooth and Convex Objectives).

Set $\theta_k = \alpha_k = 1, \eta_k = 2 \|L\|, \tau_k = \|L\|, e_i^k = \|L\| \max(\|x_0^i - x^*\|^2, \|y_0^i\|^2)$ in Algorithm 2, where $\|L\|$ denotes the spectral norm of Laplacian matrix $L$. Assume each $f_i$ is $l$-smooth, we have:

$$F(\bar{x}_N) - F(x^*) \leq \frac{\|L\|}{N} \max(\|x_0 - x^*\|^2, \|y_0\|^2).$$

(5)

From (5) it is straightforward to establish communication complexity. Note that it only depends on the number of agents and the network topology, and is independent of the objective function $f$, which is a feature that DFW does not have.

**Corollary 1** (Complexity for Smooth and Convex Objectives).

Under the same conditions as in Theorem 1 to get a solution such that $F(\bar{x}_N) - F(x^*) \leq \epsilon$, the number of communications and LO for each agent are respectively bounded by:

$$O\left(\frac{\|L\| \max(\|x_0 - x^*\|^2, \|y_0\|^2)}{\epsilon}\right)$$

(6)

and

$$O\left(\frac{m \|L\| (l + \|L\|) \max(\|x_0 - x^*\|^2, \|y_0\|^2)}{\epsilon^2}\right)$$

(7)

**Detailed Comparison.** DFW in [Wai et al., 2017] has communication and LO complexity both bounded by $O\left(\frac{lmD^2G}{\epsilon}\right)$, where $D$ denotes the diameter of the feasible set and $G$ inversely relates to the spectral gap of the weighted communication matrix. If we
set $y^0 = 0$, and observe that \( \|x^0 - x^*\| = O(\frac{1}{m}) \), then it can be seen that our algorithm is at least \( O(\frac{lG}{m}) \) times better in terms of the communication complexity, and at most \( O\left( \frac{m(l+\|L\|}{\epsilon} \right) \) worse in LO complexity. Suppose in application where the agents network is set beforehand so that \((m, \|L\|, G) \) be treated as constants, as objective becomes increasingly non-smooth, our algorithm outperforms DFW by factor of \( O(\frac{l}{\epsilon}) \) in communication with \( O(\frac{1}{\epsilon}) \) worsened LO complexity. For tasks that communication is time consuming but LO is much cheaper (e.g., matrix completion), such a trade-off can be significant, especially when we are not solving for a high precision solution.

**Theorem 2** (Convergence for Smooth and Strongly Convex Objectives).

Set \( \alpha_k = \frac{k}{k+1}, \theta_k = k+1, \eta_k = \frac{4\|L\|^2}{(k+1)^2}, \tau_k = \frac{\max(u\|x^0 - x^*\|^2, \|L\|^2\|y^0\|^2/u)}{mN_k} \) in Algorithm \( \mathcal{A} \). Assume \( f_i \) is \( u \)-strongly convex and \( l \)-smooth, we have:

\[
F(x_N) - F(x^*) \leq \frac{1}{N^2} \max \left( u \|x^0 - x^*\|^2, \frac{\|L\|^2\|y^0\|^2}{u} \right).
\]

Similarly to establishing (6) and (7), we bound the communication and LO complexity in the following corollary.

**Corollary 2** (Complexity for Smooth and Strongly Convex Objective).

Under the same conditions as in Theorem 2, to get a solution such that \( F(x_N) - F(x^*) \leq \epsilon \), the number of communications and LO for each agent can be respectively bounded by:

\[
O\left( \sqrt{\frac{\max(u \|x^0 - x^*\|^2, \|L\|^2\|y^0\|^2/u)}{\epsilon}} \right),
\]

and

\[
O\left( \frac{ml \max(u \|x^0 - x^*\|^2, \|L\|^2\|y^0\|^2/u)}{\epsilon^2} \right).
\]

**Detailed Comparison.** DFW in [Wai et al., 2017] has both communication and LO complexity bounded by \( O(\frac{1}{\sqrt{\epsilon}}) \), but requires the minimizer to be bounded way from boundary of the feasible set. Our complexity result does not rely on this unrealistic assumption that often fails, especially when the constraint should be active to impose structural assumption (e.g., sparsity) on the solution. It is then fair to compare with their result in the convex setting. Our result can be deemed as trading for a better communication complexity from \( O(\frac{1}{\epsilon}) \) to \( O(\frac{1}{\epsilon}) \), with a moderately worse LO complexity from \( O(\frac{1}{\epsilon}) \) to \( O(\frac{1}{\epsilon}) \).

4.2 Polyhedral Feasible Sets

The trade-off between communication and LO however is almost unnecessary, when the feasible set is polyhedral. Specifically, DCGS achieves the same communication and LO complexity (with additionally logarithmic factor), regardless of \( f_i \) being strongly convex or not. This improvement is a direct result from the modularity of DCGS; we replace the CG procedure in DCGS with a faster one adapted from [Lacoste-Julien and Jaggi, 2015]. We present the modified DCGS for a polyhedral feasible set in Algorithm 3.
Under the same conditions as in Theorem 2, suppose $X$ in (9), and has LO complexity:

Corollary 4 (Smooth and Convex: polyhedral set).

Under the same conditions as in Theorem 2, suppose $X$ is a polyhedral set with pyramidal width $W$ and width $D$, for DCGS each agent has the same communication complexity as in (2), and has LO complexity:

$$\tilde{O}\left(\frac{D^2 \log m(l + \|L\|) \max(\|x^0 - x^*\|^2, \|y^0\|^2)}{\epsilon}\right).$$

Improvements. Observe that the LO complexity is now at the same order of magnitude as in [Wai et al., 2017] in terms of dependence on $\epsilon$. If we set $y^0 = 0$, LO complexity of DCGS in this setting is at most $\tilde{O}(\frac{D^2 \log m(l + \|L\|)}{\epsilon})$ worse than in [Wai et al., 2017], which could even be better when the network is poorly connected (so that $G$ is large). Our complexity also depends on $\frac{D^2}{W^2}$ which could be interpreted as the condition number of the polyhedral constraint set $X$.

Corollary 3 (Smooth and strongly convex: polyhedral set).

Under the same conditions as in Theorem 2, suppose $X$ is a polyhedral set with pyramidal width $W$ and width $D$, for DCGS each agent has the same communication complexity as in (2), and has LO complexity:

$$\tilde{O}\left(\frac{D^2 l \log m}{W^2} \sqrt{\frac{\max(u \|x^0 - x^*\|^2, \|L\|^2 \|y^0\|^2)}{\epsilon}}\right).$$

Improvements. The LO complexity is now improved to $\tilde{O}(\frac{l}{\sqrt{\epsilon}})$ which is of the same order of magnitude as in [Wai et al., 2017] in terms dependence on $\epsilon$, but this result makes no assumption on the minimizer. If we choose $y^0 = 0$, our LO complexity reduces to: $\tilde{O}\left(\frac{D^2 l \log m}{W^2} \frac{\|x^0 - x^*\|}{\sqrt{\epsilon}}\right)$ which has a clean interpretation in terms of its dependency on condition number of the objective and condition number of the polyhedral constraint.
5 Experimental Results

In this section we present experiments comparing DCGS with the existing distributed FW algorithm in (Wai et al., 2017) and demonstrate the superiority of our algorithm. For both of the following experiments, we set the number of agents $m = 10$, and the associated network to be a 1-connected cycle, i.e. each agent $i$ is connected to its previous one $i - 1$ and the latter one $i + 1$.

**Lasso.** We compare DCGS and DFW applying to the Lasso problem on a synthetic dataset and E2006-tfidf dataset (Kogan et al., 2009). The Lasso problem is formulated as:

$$
\min_{\|\theta\|_1 \leq \rho} \left\| X\theta - Y \right\|^2 = \sum_{i=1}^{n} (X_i^T\theta - Y_i)^2.
$$

(13)

Similar experiment was also conducted in (Lacoste-Julien and Jaggi, 2015) which showed linear convergence of FW variant over the polyhedral set. For synthetic data, we generate $n = 2000$ samples, with $X_i$ sampled i.i.d. from $N(0, I_d)$ and $d = 10000$. We generate $\|\theta_0\| = 100$ with randomly selected 100 non-zero entries, and $Y_i = X_i^T\theta_0 + \epsilon_i$ with $\epsilon_i \sim N(0, 1)$. For the real dataset, $d = 150360$ and we randomly draw $n = 4000$ samples from the entire E2006-tfidf dataset. We set $\rho = 10^3$ and distribute data evenly across agents. The results are presented in Figure 1. We observed significant performance improvement of DCGS over DFW. For the synthetic data, Figure 1(a) shows that DCGS with a moderately number of LO, converges to a high-quality solution that has loss by orders of magnitude better than DFW. The gap on communication complexity is even more significant. DFW algorithm takes more than 800 rounds of communications while DCGS only takes 3 rounds. We observe similar performance gap on E2006-tfidf dataset in Figure 1(c, d).

**Matrix completion.** We compare DCGS and DFW applying to matrix completion problems on synthetic dataset and MovieLens 100K dataset (Harper and Konstan, 2015). We remark that matrix completion is in fact a communication expensive task. As a toy example, consider a $10^4 \times 10^4$ matrix which takes 800 MB memory, sending this matrix with 10 MB/s network speed takes 80 seconds, however an LO on a 4-core machine with Intel(R) Core(TM) i5-6267U CPU @ 2.90GHz processor and 16GB RAM takes less than 2 seconds. This means for algorithms such as DFW, over 97% of computation time would be waiting for the communication to complete. We present our simulation results in Figure 2. For the synthetic data set we generate the ground truth matrix $\Theta^* \in \mathbb{R}^{d \times d}$ with $d = 200$ and rank $r = 5$. We randomly sample $n = 5000$ entries $\{(a(i), b(i))\}_{i=1}^n$ and observe $\Theta^*_{a(i)b(i)} = \Theta^*_{a(i)b(i)} + \epsilon_i$ with $\epsilon_i \sim N(0, 0.1)$. For MovieLens 100K dataset we want to recover the rating matrix $\Theta^* \in \mathbb{R}^{d_1 \times d_2}$ with $d_1 = 943$, $d_2 = 1682$ and we observed
\( n = 10^5 \) ratings. We set \( \rho = 10^4 \) and run the algorithms on solving the following objective:

\[
\min_{\|X\|_\ast \leq \rho} \sum_{i=1}^{n} \left( X_{a(i)b(i)} - \Theta_{a(i)b(i)}^{P} \right)^2.
\] (14)

For the synthetic data, Figure 2a and 2b show DCGS and DFW need comparable LO to converge to a moderate precision solution, however DFW takes significantly more rounds of communication (800 vs. 5). Since communication is the main computation bottle as we discussed above, DCGS would significantly outperform DFW in terms the actual running time. We observe similar performance gap on MovieLens 100K dataset in Figure 2c,2d. Our experiment results thus suggest the applicability of DCGS in communication expensive applications.

![Figure 2](image_url)

Figure 2: Matrix Completion. (a)(b) are for the synthetic data; (c)(d) are for the real data.

6 Conclusions

In this paper, we propose a communication efficient, distributed projection-free algorithm called DCGS. We show that DCGS is communication efficient under convex and strongly convex setting without restrictive assumptions in existing work, and demonstrate the superiority of DCGS in communication expensive learning tasks such as matrix completion. We also show DCGS can be further improved when the feasible set is polyhedral, which is also validated by our numerical experiments. Future research directions include developing asynchronous DCGS variant and extending DCGS to non-convex settings.
References

Bach, F., Lacoste-Julien, S., Obozinski, G., 2012. On the equivalence between herding and conditional gradient algorithms. ICML’12. Omnipress, USA, pp. 1355–1362.

Bellet, A., Liang, Y., Garakani, A. B., Balcan, M.-F., Sha, F., 2015. A Distributed Frank-Wolfe Algorithm for Communication-Efficient Sparse Learning. pp. 478–486.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., Jan. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. 3 (1), 1–122.

Chambolle, A., Pock, T., May 2011. A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision 40 (1), 120–145.

Clarkson, K. L., Sep. 2010. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. ACM Trans. Algorithms 6 (4), 63:1–63:30.

Collins, M., Globerson, A., Koo, T., Carreras, X., Bartlett, P. L., Jun. 2008. Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. J. Mach. Learn. Res. 9, 1775–1822.

Duchi, J. C., Agarwal, A., Wainwright, M. J., March 2012. Dual averaging for distributed optimization: Convergence analysis and network scaling. IEEE Transactions on Automatic Control 57 (3), 592–606.

Fujishige, S., Isotani, S., 2011. A submodular function minimization algorithm based on the minimum-norm base. Pacific Journal of Optimization.

Garber, D., Hazan, E., Jan. 2013. A Linearly Convergent Conditional Gradient Algorithm with Applications to Online and Stochastic Optimization. ArXiv e-prints.

Harchaoui, Z., Juditsky, A., Nemirovski, A., Aug 2015. Conditional gradient algorithms for norm-regularized smooth convex optimization. Mathematical Programming 152 (1), 75–112.

Harper, F. M., Konstan, J. A., Dec. 2015. The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. 5 (4), 19:1–19:19.

Hazan, E., Kale, S., 2012. Projection-free online learning. ICML’12. Omnipress, USA, pp. 1843–1850.

Hazan, E., Luo, H., 2016. Variance-reduced and projection-free stochastic optimization. ICML’16. JMLR.org, pp. 1263–1271.

Jaggi, M., 2011. Sparse convex optimization methods for machine learning. Ph.D. thesis, ETH Zurich.

Jaggi, M., 2013. Revisiting frank-wolfe: Projection-free sparse convex optimization. ICML’13. JMLR.org, pp. 1–427–1–435.

Kogan, S., Levin, D., Routledge, B. R., Sagi, J. S., Smith, N. A., 2009. Predicting risk from financial reports with regression. NAACL ’09. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 272–280.
Lacoste-Julien, S., Jaggi, M., 2015. On the global linear convergence of frank-wolfe optimization variants. NIPS’15. MIT Press, Cambridge, MA, USA, pp. 496–504.

Lacoste-Julien, S., Jaggi, M., Schmidt, M., Pletscher, P., 2013. Block-coordinate frank-wolfe optimization for structural svms. ICML’13. JMLR.org, pp. I–53–I–61.

Lafond, J., Wai, H. T., Moulines, E., March 2016. D-fw: Communication efficient distributed algorithms for high-dimensional sparse optimization. In: ICASSP’2016. pp. 4144–4148.

Lan, G., Lee, S., Zhou, Y., Jan. 2017. Communication-Efficient Algorithms for Decentralized and Stochastic Optimization. ArXiv e-prints.

Lan, G., Zhou, Y., 2016. Conditional gradient sliding for convex optimization. SIAM Journal on Optimization 26 (2), 1379–1409.

Makhdoumi, A., Ozdaglar, A., Oct 2017. Convergence rate of distributed admm over networks. IEEE Transactions on Automatic Control 62 (10), 5082–5095.

Nedic, A., June 2011. Asynchronous broadcast-based convex optimization over a network. IEEE Transactions on Automatic Control 56 (6), 1337–1351.

Nedic, A., Ozdaglar, A., Jan 2009. Distributed subgradient methods for multi-agent optimization. IEEE Transactions on Automatic Control 54 (1), 48–61.

Nedich, A., Olshevsky, A., Shi, W., Jul. 2016. Achieving Geometric Convergence for Distributed Optimization over Time-Varying Graphs. ArXiv e-prints.

Shi, W., Ling, Q., Wu, G., Yin, W., 2015a. Extra: An exact first-order algorithm for decentralized consensus optimization. SIAM Journal on Optimization 25 (2), 944–966.

Shi, W., Ling, Q., Wu, G., Yin, W., Nov 2015b. A proximal gradient algorithm for decentralized composite optimization. IEEE Transactions on Signal Processing 63 (22), 6013–6023.

Shi, W., Ling, Q., Yuan, K., Wu, G., Yin, W., April 2014. On the linear convergence of the admm in decentralized consensus optimization. IEEE Transactions on Signal Processing 62 (7), 1750–1761.

Sundhar Ram, S., Nedic, A., Veeravalli, V. V., Dec 2010. Distributed stochastic subgradient projection algorithms for convex optimization. Journal of Optimization Theory and Applications 147 (3), 516–545.

Terelius, H., Topcu, U., Murray, R. M., 2011. Decentralized multi-agent optimization via dual decomposition. IFAC Proceedings Volumes 44 (1), 11245 – 11251, 18th IFAC World Congress.

Tsitsiklis, J., Bertsekas, D., Athans, M., Sep 1986. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. IEEE Transactions on Automatic Control 31 (9), 803–812.

Tsitsiklis, J. N., 1985. Problems in decentralized decision making and computation. Ph.D. thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science.
Wai, H. T., Lafond, J., Scaglione, A., Moulines, E., Nov 2017. Decentralized frank-wolfe algorithm for convex and nonconvex problems. IEEE Transactions on Automatic Control 62 (11), 5522–5537.

Wang, Y.-X., Sadhanala, V., Dai, W., Neiswanger, W., Sra, S., Xing, E., 20–22 Jun 2016. Parallel and distributed block-coordinate frank-wolfe algorithms. Vol. 48 of Proceedings of Machine Learning Research. PMLR, New York, New York, USA, pp. 1548–1557.

Wei, E., Ozdaglar, A., Jul. 2013. On the O(1/k) Convergence of Asynchronous Distributed Alternating Direction Method of Multipliers. ArXiv e-prints.

Yuan, K., Ling, Q., Yin, W., 2016. On the convergence of decentralized gradient descent. SIAM Journal on Optimization 26 (3), 1835–1854.
Supplemental Material

Communication Efficient Projection Free Algorithm for Distributed Optimization

Outline. In Appendix A we prove the convergence results for Decentralized Conditional Gradient Sliding. In Appendix B we establish the communication complexity and the LO complexity of DCGS for the general constraint set. In Appendix C we establish the communication complexity and the LO complexity of DCGS for the polyhedral constraint set.

A Proof of Main Theorem

A.1 Proof of Theorem 1

In this subsection we prove the convergence result for smooth and convex objective. Denote $Y = \mathbb{R}^{md}$ and recall the saddle point problem defined in (4), for $z = (x, y), z' = (x', y') \in X^m \times Y$, we define the primal-dual gap function to be:

$$Q(z, z') = F(x) + \langle Lx, y \rangle - F(x) - \langle Lx, y \rangle$$

(15)

Note that if $z^* = (x^*, y^*)$ is a saddle point to (4), then $Q(z^*, z) \leq 0$ and $Q(z, z^*) \geq 0$ for any $z \in X^m \times Y$. It is then natural to measure the quality of a solution $z$ to problem (4) by $\sup_{z \in X^m \times Y} Q(z^*, z)$. To handle unboundedness of $Y$ here, we define the modified gap function to be:

$$g_Y(s, z) = \sup_{y \in Y} Q(z; (x^*, y)) - \langle s, y \rangle$$

(16)

In fact, we have the following proposition.

Proposition 1 ((Lan et al., 2017)). If we have $g_Y(s, z) \leq \epsilon$ for any $\epsilon > 0$, then we must have $Lx = s$ and $F(x) - F(x^*) \leq \epsilon$.

this claim is straightforward to establish by following the definition of $g_Y(s, z)$ and a proof by contradiction argument.

By construction of $y^k_i$ in Algorithm 2 we know that:

$$\langle v^k_i, y^k_i - y_i \rangle = \frac{\tau_i}{2} \left[ \left\| y_i - y^{k-1}_i \right\|^2 - \left\| y_i - y^k_i \right\|^2 - \left\| y^{k-1}_i - y^k_i \right\|^2 \right]$$

(17)

Similar inequality could be established for $x^k_i$. Recall in Line 8 of Algorithm 2 we are solving the subproblem

$$\min_{x_i \in X} \phi^k_i(x_i) = \langle w^k_i, x_i \rangle + f_i(x_i) + \frac{\eta_k}{2} \left\| x_i - x_i^{k-1} \right\|^2$$

(18)
with the output $x_i^k$ satisfying inequality $\langle \phi(x_i^k), x_i^k - x_i^{k-1} \rangle \leq \epsilon_i^k$. Since $\phi(x_i)$ is strongly convex, we have: $\phi(x_i) - \phi(x_i^k) \geq \langle \nabla \phi(x_i^k), x_i - x_i^k \rangle + \frac{L}{2} \| x_i - x_i^k \|$. Combine this two inequalities with some algebraic rearrangements yields the following:

\[
\langle w_i^k, x_i^k - x_i \rangle + f_i(x_i^k) - f_i(x_i) \leq \frac{\eta_k}{2} \left[ \| x_i - x_i^{k-1} \|^2 - \| x_i - x_i^k \|^2 - \| x_i^{k-1} - x_i^k \|^2 \right] + \epsilon_i^k
\]

(19)

Summing up the previous two inequalities and using the definition of $Q(\cdot, \cdot)$, $w_i^k$, $\epsilon_i^k$ we have:

\[
Q(z^k, z) = F(x^k) - F(x) + \langle Lx, y \rangle - \langle Lx, y^k \rangle
\]

\[
= \langle L(x^k - \bar{x}^k), y - y^k \rangle + \frac{\eta_k}{2} \left[ \| x^k - x \|^2 - \| x^k - x^k \|^2 - \| x^{k-1} - x^k \|^2 \right]
\]

\[
+ \frac{\tau_k}{2} \left[ \| y - y^{k-1} \|^2 - \| y - y^k \|^2 - \| y^k - y^{k-1} \|^2 \right] + \sum_{i=1}^{m} \epsilon_i^k
\]

(20)

We define the right hand side of previous equation by $\Delta_k$, and we are going to handle the weighted sum of the first three terms in $\Delta_k$ separately. For the first term:

\[
\sum_{k=1}^{N} \theta_k \langle L(x^k - \bar{x}^k), y - y^k \rangle = \sum_{k=1}^{N} \theta_k \left[ \langle L(x^k - x^{k-1}), y - y^k \rangle - \alpha_k \langle L(x^{k-1} - x^{k-2}), y - y^k \rangle \right]
\]

(21)

\[
= \sum_{k=1}^{N} \left[ \theta_k \langle L(x^k - x^{k-1}), y - y^k \rangle - \alpha_k \theta_k \langle L(x^{k-1} - x^{k-2}), y - y^{k-1} \rangle \right]
\]

\[
+ \sum_{k=1}^{N} \alpha_k \theta_k \langle L(x^{k-1} - x^{k-2}), y^{k-1} - y^k \rangle
\]

\[
= \sum_{k=1}^{N} \left[ \theta_k \langle L(x^k - x^{k-1}), y - y^k \rangle - \theta_{k-1} \langle L(x^{k-1} - x^{k-2}), y - y^{k-1} \rangle \right]
\]

\[
+ \sum_{k=1}^{N} \alpha_k \theta_k \langle L(x^{k-1} - x^{k-2}), y^{k-1} - y^k \rangle
\]

(22)

\[
= \theta_N \langle L(x^N - x^{N-1}), y - y^N \rangle
\]

\[
+ \sum_{k=1}^{N} \alpha_k \theta_k \langle L(x^{k-1} - x^{k-2}), y^{k-1} - y^k \rangle
\]

(23)

where (21) follows from definition of $\bar{x}^k$. In (22) we use condition $\alpha_k \theta_k = \theta_{k-1}$ which follows from our parameters setting. (23) comes from telescoping the first summation in (22) and the condition that $x^0 = x^{-1}$. We could bound the weighted sum of the second
We further rewrite the first summation term in (26) as the following:

\[
\sum_{k=1}^{N} \frac{\theta_k \eta_k}{2} \left[ \left\| x^{k-1} - x \right\|^2 - \left\| x^k - x \right\|^2 - \left\| x^{k-1} - x^k \right\|^2 \right] = \frac{\theta_1 \eta_1}{2} \left\| x^0 - x \right\|^2 - \frac{\theta_N \eta_N}{2} \left\| x^N - x \right\|^2 \\
+ \sum_{k=1}^{N-1} \frac{\theta_{k+1} \eta_{k+1} - \theta_k \eta_k}{2} \left\| x^k - x \right\|^2 - \sum_{k=1}^{N} \frac{\theta_k \eta_k}{2} \left\| x^{k-1} - x^k \right\|^2 \\
\leq \frac{\theta_1 \eta_1}{2} \left\| x^0 - x \right\|^2 - \frac{\theta_N \eta_N}{2} \left\| x^N - x \right\|^2 \\
- \sum_{k=1}^{N} \frac{\theta_k \eta_k}{2} \left\| x^{k-1} - x^k \right\|^2
\]

(24)

where in (21) we use the condition \( \theta_{k+1} \eta_{k+1} \leq \theta_k \eta_k \) which follows from our parameters setting. Similarly we can bound the weighted sum of the third term in (20):

\[
\sum_{k=1}^{N} \frac{\theta_k \eta_k}{2} \left[ \left\| y^{k-1} - x \right\|^2 - \left\| y^k - y \right\|^2 - \left\| y^{k-1} - y^k \right\|^2 \right] = \frac{\theta_1 \eta_1}{2} \left\| y^0 - y \right\|^2 - \frac{\theta_N \eta_N}{2} \left\| y^N - y \right\|^2 \\
+ \sum_{k=1}^{N-1} \frac{\theta_{k+1} \eta_{k+1} - \theta_k \eta_k}{2} \left\| y^k - y \right\|^2 - \sum_{k=1}^{N} \frac{\theta_k \eta_k}{2} \left\| y^{k-1} - y^k \right\|^2 \\
\leq \frac{\theta_1 \eta_1}{2} \left\| y^0 - y \right\|^2 - \frac{\theta_N \eta_N}{2} \left\| y^N - y \right\|^2 \\
- \sum_{k=1}^{N} \frac{\theta_k \eta_k}{2} \left\| y^{k-1} - y^k \right\|^2
\]

(25)

Now sum up (24), (21), (24) we get:

\[
\sum_{k=1}^{N} \theta_k \Delta_k \leq \theta_N \langle \mathbf{L}(x^N - x^{N-1}), y - y^N \rangle \\
+ \sum_{k=1}^{N} \theta_k [\alpha_k \langle \mathbf{L}(x^{k-1} - x^{k-2}), y^{k-1} - y^k \rangle - \frac{\eta_k}{2} \left\| x^{k-1} - x^k \right\|^2 - \frac{\tau_k}{2} \left\| y^{k-1} - y^k \right\|^2 ] \\
+ \frac{\theta_1 \eta_1}{2} \left\| x^0 - x \right\|^2 - \frac{\theta_N \eta_N}{2} \left\| x^N - x \right\|^2 \\
+ \frac{\theta_1 \tau_1}{2} \left\| y^1 - y \right\|^2 - \frac{\theta_N \tau_N}{2} \left\| y^N - y \right\|^2 + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_i k_i^2
\]

(26)

We further rewrite the first summation term in (26) as the following:

\[
\sum_{k=1}^{N} \theta_k [\alpha_k \langle \mathbf{L}(x^{k-1} - x^{k-2}), y^{k-1} - y^k \rangle - \frac{\eta_k}{2} \left\| x^{k-1} - x^k \right\|^2 - \frac{\tau_k}{2} \left\| y^{k-1} - y^k \right\|^2 ] \\
= \sum_{k=2}^{N} \left[ \theta_k \alpha_k \langle \mathbf{L}(x^{k-1} - x^{k-2}), y^{k-1} - y^k \rangle - \frac{\theta_{k-1} \eta_{k-1}}{2} \left\| x^{k-2} - x^{k-1} \right\|^2 - \frac{\theta_k \tau_k}{2} \left\| y^{k-1} - y^k \right\|^2 \\
- \frac{\theta_N \eta_N}{2} \left\| x^{N-1} - x^N \right\|^2 - \frac{\eta_1 \tau_1}{2} \left\| y^0 - y^1 \right\|^2
\]

(27)
Combine (26) and (27) we obtain:

\[
\sum_{k=1}^{N} \theta_k \Delta_k \leq \theta_N \langle L(x^N - x^{N-1}), y - y^N \rangle - \frac{\theta_N \eta N}{2} \|x^{N-1} - x^N\|^2 - \frac{\theta_1 \tau_1}{2} \|y^0 - y^1\|^2 \\
+ \sum_{k=2}^{N} \left[ \theta_k \alpha_k \langle L(x^{k-1} - x^{k-2}), y^{k-1} - y^k \rangle - \frac{\theta_{k-1} \eta_{k-1}}{2} \|x^{k-2} - x^{k-1}\|^2 - \frac{\theta_k \tau_k}{2} \|y^{k-1} - y^k\|^2 \right] \\
+ \frac{\theta_1 \eta_1}{2} \|x^0 - x\|^2 - \frac{\theta_N \eta N}{2} \|x^N - x\|^2 \\
+ \frac{\theta_1 \tau_1}{2} \|y^0 - y\|^2 - \frac{\theta_N \tau_N}{2} \|y^N - y\|^2 + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_k e_i^k \quad (28)
\]

The summation in the second line could be in fact upper bounded by 0 as the following:

\[
\sum_{k=2}^{N} \left[ \theta_k \alpha_k \langle L(x^{k-1} - x^{k-2}), y^{k-1} - y^k \rangle - \frac{\theta_{k-1} \eta_{k-1}}{2} \|x^{k-2} - x^{k-1}\|^2 - \frac{\theta_k \tau_k}{2} \|y^{k-1} - y^k\|^2 \right] \\
\leq \sum_{k=2}^{N} \left[ \theta_k \alpha_k \|L\| \|x^{k-1} - x^{k-2}\| \|y^{k-1} - y^k\| - \frac{\theta_{k-1} \eta_{k-1}}{2} \|x^{k-2} - x^{k-1}\|^2 - \frac{\theta_k \tau_k}{2} \|y^{k-1} - y^k\|^2 \right] \\
\leq \sum_{k=2}^{N} \left( \frac{\|L\|^2}{2 \tau_k \theta_k} - \frac{\theta_{k-1} \eta_{k-1}}{2} \right) \|x^{k-2} - x^{k-1}\|^2 \quad (29) \\
= \sum_{k=2}^{N} \left( \frac{\theta_{k-1} \alpha_k \|L\|^2}{2 \tau_k} - \frac{\theta_{k-1} \eta_{k-1}}{2} \right) \|x^{k-2} - x^{k-1}\|^2 \leq 0 \quad (30)
\]

where in (29) we use the Young’s inequality, in (30) we use the condition \(\theta_k \alpha_k = \theta_{k-1}, \alpha_k \|L\|^2 \leq \tau_k \eta_{k-1}\) which follows from our parameter setting. In summary we get:

\[
\sum_{k=1}^{N} \theta_k Q(z^k, z) = \sum_{k=1}^{N} \theta_k \Delta_k \leq \theta_N \langle L(x^N - x^{N-1}), y - y^N \rangle - \frac{\theta_N \eta N}{2} \|x^{N-1} - x^N\|^2 \\
+ \frac{\theta_1 \eta_1}{2} \|x^0 - x\|^2 - \frac{\theta_N \eta N}{2} \|x^N - x\|^2 \\
+ \frac{\theta_1 \tau_1}{2} \|y^0 - y\|^2 - \frac{\theta_N \tau_N}{2} \|y^N - y\|^2 + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_k e_i^k \quad (31)
\]

Our next objective is to bound the right hand side of (31) as a linear function on \(y\).
Collecting all the linear term of \( y \) after some rearrangement, we get:

\[
\sum_{k=1}^{N} \theta_k Q(z^k, z) \leq \theta_N \langle y^N, L(x^{N-1} - x^N) \rangle - \frac{\theta_N \eta_{N}}{2} \|x^{N-1} - x^N\| - \frac{\theta_N \tau_{N}}{2} \|y^N\|^2 \\
+ \frac{\theta_1 \eta_1}{2} \|x^0 - x\|^2 + \frac{\theta_1 \tau_1}{2} \|y^0\|^2 \\
+ \langle y, \theta_N L(x^N - x^{N-1}) + \theta_1 \tau_1 (y^N - y^0) \rangle + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_k e_i^k \\
\leq \theta_N \|L\| \|x^{N-1} - x^N\| \|y^N\| - \frac{\theta_N \eta_{N}}{2} \|x^{N-1} - x^N\| - \frac{\theta_N \tau_{N}}{2} \|y^N\|^2 \\
+ \frac{\theta_1 \eta_1}{2} \|x^0 - x\|^2 + \frac{\theta_1 \tau_1}{2} \|y^0\|^2 \\
+ \langle y, \theta_N L(x^N - x^{N-1}) + \theta_1 \tau_1 (y^N - y^0) \rangle + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_k e_i^k \\
\leq \left( \frac{\theta_N \|L\|^2}{2 \eta_{N}} - \frac{\theta_1 \tau_1}{2} \right) \|y^N\|^2 + \frac{\theta_1 \eta_1}{2} \|x^0 - x\|^2 + \frac{\theta_1 \tau_1}{2} \|y^0\|^2 \\
+ \langle y, \theta_N L(x^N - x^{N-1}) + \theta_1 \tau_1 (y^N - y^0) \rangle + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_k e_i^k \\
(32) \\
\leq \frac{\theta_1 \eta_1}{2} \|x^0 - x\|^2 + \frac{\theta_1 \tau_1}{2} \|y^0\|^2 \\
+ \langle y, \theta_N L(x^N - x^{N-1}) + \theta_1 \tau_1 (y^N - y^0) \rangle + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_k e_i^k \\
(33)
\]

where in (32) we use the Young’s inequality, in (33) we used the condition that \( \theta_N \|L\|^2 \leq \theta_1 \tau_1 \eta_{N} \) which is satisfied by our parameters. Let us define \( s^N = \theta_N L(x^N - x^{N-1}) + \theta_1 \tau_1 (y^N - y^0) \), then we have shown that:

\[
\sum_{k=1}^{N} \theta_k Q(z^k, z) \leq \frac{\theta_1 \eta_1}{2} \|x^0 - x\|^2 + \frac{\theta_1 \tau_1}{2} \|y^0\|^2 + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_k e_i^k + \langle y, s^N \rangle \\
(34)
\]

Choosing \( z = (x^*, y) \) in the left hand side, and using the convexity of \( Q(\cdot, z) \), we immediately have:

\[
Q(\mathbf{z}^N; (x^*, y)) \leq (\sum_{k=1}^{N} \theta_k)^{-1} \left( \frac{\theta_1 \eta_1}{2} \|x^0 - x^*\|^2 + \frac{\theta_1 \tau_1}{2} \|y^0\|^2 + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_k e_i^k + \langle y, s^N \rangle \right) \\
(35)
\]

From the definition of \( g_Y(s, z) \), and define \( \mathbf{s}^N = (\sum_{k=1}^{N} \theta_k)^{-1} s^N \), we have:

\[
g_Y(\mathbf{s}^N, \mathbf{z}^N) \leq (\sum_{k=1}^{N} \theta_k)^{-1} \left( \frac{\theta_1 \eta_1}{2} \|x^0 - x^*\|^2 + \frac{\theta_1 \tau_1}{2} \|y^0\|^2 + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_k e_i^k \right) \\
(36)
\]

which then implies \( \mathbf{Lx}^N = \mathbf{s}^N \) and:

\[
F(\mathbf{x}^N) - F(x^*) \leq (\sum_{k=1}^{N} \theta_k)^{-1} \left( \frac{\theta_1 \eta_1}{2} \|x^0 - x^*\|^2 + \frac{\theta_1 \tau_1}{2} \|y^0\|^2 + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_k e_i^k \right) \\
\]
Now plug in choice of $\alpha_1 = \theta_k = 1$, $\eta_k = 2 \|L\|$, $\tau_k = \|L\|$ and $e_i^k = \|L\| \max \|x^0-x^0\| \|y^0\|^2$ yields our convergence result.

### A.2 Proof of Theorem 2

In this subsection we prove the convergence result for the smooth and strongly convex objective. For strongly convex $f_i$, again by the update for $x_i^k$ in Algorithm 2 we have $\langle \nabla \phi^k_i(x_i^k), x_i^k - x_i^{k-1} \rangle \leq e_i^k$. Since $\phi^k_i(x_i)$ is strongly convex we have: $\phi^k_i(x_i) - \phi^k_i(x_i^k) \geq \langle \nabla \phi^k_i(x_i^k), x_i - x_i^k \rangle + (u + \frac{\eta_k}{2}) \|x_i - x_i^k\|$. Combine this two inequalities with some algebraic rearrangements we get:

$$
\langle w_i^k, x_i - x_i \rangle + f_i(x_i^k) - f_i(x_i) \leq \frac{\eta_k}{2} \|x_i - x_i^k\|^2 - \left(\frac{\eta_k}{2} + u\right) \|x_i - x_i^k\|^2 - \frac{\eta_k}{2} \|x_i^{k-1} - x_i^k\|^2 + e_i^k
$$

Note we still have have $\mathbf{17}$ since the update for $y_i^k$ does not change. Following the same argument of $\mathbf{20}$ we have:

$$
Q(z^k, z) = F(x^k) - F(x) + \langle Lx^k, y \rangle - \langle Lx, y^k \rangle = \langle L(x^k - \hat{x}^k), y - y^k \rangle + \left(\frac{\eta_k}{2} \|x^{k-1} - x\|^2 - \left(\frac{\eta_k}{2} + u\right) \|x^k - x\|^2 - \frac{\eta_k}{2} \|x^{k-1} - x^k\|^2 \right)
$$

$$
+ \frac{\tau_k}{2} \left(\|y - y^{k-1}\|^2 - \|y - y^k\|^2 - \|y^k - y^{k-1}\|^2\right) + \sum_{i=1}^m e_i^k
\tag{37}
$$

still we can bound the weighted sum the of first term as in $\mathbf{28}$ and the weighted sum of third term as in $\mathbf{25}$, the paramter condition required by establishing them is still satisfied. Handling the second term is also essentially the same:

$$
\sum_{k=1}^N \theta_k \left(\frac{\eta_k}{2} \|x^{k-1} - x\|^2 - \left(\frac{\eta_k}{2} + u\right) \|x^k - x\|^2 - \frac{\eta_k}{2} \|x^{k-1} - x^k\|^2 \right)
$$

$$
= \frac{\theta_1 \eta_1}{2} \|x^0 - x\|^2 + \sum_{k=1}^{N-1} \frac{\theta_{k+1} \eta_{k+1} - \theta_k(\eta_k + u)}{2} \|x^k - x\|^2 - \frac{\theta_N(\eta_N + u)}{2} \|x^N - x\|^2
$$

$$
- \sum_{k=1}^N \frac{\theta_k \eta_k}{2} \|x^{k-1} - x^k\|^2
$$

$$
\leq \frac{\theta_1 \eta_1}{2} \|x^0 - x\|^2 - \frac{\theta_N(\eta_N + u)}{2} \|x^N - x\|^2 - \sum_{k=1}^N \frac{\theta_k \eta_k}{2} \|x^{k-1} - x^k\|^2 \tag{38}
$$

where $\mathbf{38}$ comes from the condition $\theta_{k+1} \eta_{k+1} \leq \theta_k(\eta_k + u)$ which follows from our parameter setting. Now add up $\mathbf{23}$, $\mathbf{25}$, $\mathbf{38}$ and combine with $\mathbf{37}$ we have the following:

$$
\sum_{k=1}^N \theta_k Q(z^k, z) = \sum_{k=1}^N \theta_k \Delta_k \leq \theta_N (L(x^N - x^{N-1}), y - y^N) - \frac{\theta_N \eta_N}{2} \|x^{N-1} - x^N\|^2
$$

$$
+ \frac{\theta_1 \eta_1}{2} \|x^0 - x\|^2 - \frac{\theta_N(\eta_N + u)}{2} \|x^N - x\|^2
$$

$$
+ \frac{\theta_1 \tau_1}{2} \|y^0 - y\|^2 - \frac{\theta_N \tau_N}{2} \|y^N - y\|^2 + \sum_{k=1}^N \sum_{i=1}^m \theta_k e_i^k \tag{39}
$$
We now need to bound the right hand side of (39) by a linear function on $y$. This is exactly the same as in establishing (33) and hence we omit the tedious detail. In summary, we get the following bound that is identical to (33):

$$\sum_{k=1}^{N} \theta_k Q(z^k, z) \leq \frac{\theta_1 \eta_1}{2} \|x^0 - x\|^2 + \frac{\theta_1 \tau_1}{2} \|y^0\|^2$$

Then following the same argument as in the convex case. Define $s^N$ and $s^N$ as before, we then can conclude $Lx^N = s^N$ and:

$$F(x^N) - F(x^*) \leq \left(\sum_{k=1}^{N} \theta_k\right)^{-1} \left(\frac{\theta_1 \eta_1}{2} \|x^0 - x^*\|^2 + \frac{\theta_1 \tau_1}{2} \|y^0\|^2 + \sum_{k=1}^{N} \sum_{i=1}^{m} \theta_k e_i^k\right)$$

Plug in definition of $\alpha_k = \frac{k+1}{k+1}, \theta_k = k+1, \eta_k = \frac{k u}{2}, \tau_k = \frac{4\|L\|^2}{(k+1) u}$ and $e_i^k = \frac{\max(u \|x^0 - x^*\|^2, \|y^0\|^2)}{2\|y^0\|^2}$ yields our convergence result.

**B Complexity of DCGS: General Constraints**

**B.1 Proof of Corollary 1**

**Proof.** From Theorem 1, we know that to get an $\epsilon$-optimal solution, we need at most

$$N = \mathcal{O}\left(\frac{l}{\epsilon} \max\left(\|x^0 - x^*\|^2, \|y^0\|^2\right)\right)$$

number of outer iterations in DCGS. Now we bound the number of calls to LO in the $k$-th outer iteration. Recall the CG procedure could be seemed as solving the subproblem in Line 8 of Algorithm 2 by Frank-Wolfe algorithm. From the well known result (Jaggi, 2013) we know that for solving a $l$-smooth function using Frank-Wolfe algorithm, with terminating wolfe-gap being $\epsilon$, the number of iterations could be bounded by $\frac{D^2}{2\epsilon^2}$. Observe that the objective function $\phi^k$ in the subproblem has smoothness $l + \eta_k$, the total LO calls for each agent could be bounded by:

$$\sum_{k=1}^{N} \frac{(l + \eta_k) D^2}{e_i^k}$$

$$= \sum_{k=1}^{N} \frac{(l + \|L\|) D^2 m N}{\max(\|x^0 - x^*\|^2, \|y^0\|^2)}$$

$$= \frac{(l + \|L\|) D^2 m N^2}{\max(\|x^0 - x^*\|^2, \|y^0\|^2)}$$

$$= \mathcal{O}\left(\frac{(l + \|L\|) D^2 m \max(\|x^0 - x^*\|^2, \|y^0\|^2)}{\epsilon^2}\right)$$

**B.2 Proof of Corollary 2**

**Proof.** From Theorem 2, we know that to get an $\epsilon$-optimal solution, we need at most

$$N = \mathcal{O}\left(\sqrt{\frac{\max(u \|x^0 - x^*\|^2, \|y^0\|^2)}{\epsilon}}\right)$$

number of outer iterations in DCGS. Following the
same argument as in [11], we could be the LO complexity for each agent by:

$$\sum_{k=1}^{N} \frac{(l + \eta_k)D^2}{\epsilon^2_k}$$

$$= \sum_{k=1}^{N} \frac{(l + ku)D^2 mNk}{\max(u \|x^0 - x^*\|^2, \|L\|\|y^0\|^2)}$$

$$= \sum_{k=1}^{N} \frac{(l + ku)D^2 mNk}{\max(u \|x^0 - x^*\|^2, \|L\|\|y^0\|^2)}$$

$$\leq O\left( \sum_{k=1}^{N} \frac{lD^2 mNk^2}{\max(u \|x^0 - x^*\|^2, \|L\|\|y^0\|^2)} \right)$$

$$= O\left( \frac{lD^2 mN^4}{\max(u \|x^0 - x^*\|^2, \|L\|\|y^0\|^2)} \right)$$

$$= O\left( \frac{l mD^2 \max(u \|x^0 - x^*\|^2, \|L\|\|y^0\|^2)}{\epsilon^2} \right)$$

C Complexity of DCGS: Polyhedral Constraints

C.1 Proof of Corollary 3

Proof. From Theorem 4 we know that to get an $\epsilon$-optimal solution, we need at most $N = O\left( \frac{\|L\|}{\epsilon} \max(\|x^0 - x^*\|^2, \|y^0\|^2) \right)$ number of outer iteration in DCGS. Now we bound the number of calls to LO in the $k$-th outer iteration. Our PCG procedure in DCGS could be seemed as pairwise Frank-Wolfe algorithm in [Lacoste-Julien and Jaggi, 2015] applied to subproblem $\min_{x_i \in X} \phi^k_i(x_i)$. It has been shown that for a $u$-strongly convex and $l$-smooth function over a polyhedral set that has width $D$ and pyramidal width $W$, pairwise FW achieves a linear convergence rate of the wolfe gap. Specifically if we let $g^t$ denotes the wolfe gap at $t$-th iteration of pairwise FW algorithm, we have:

$$g^t \leq (1 - \frac{u}{4l}(\frac{W}{D})^2)^{\frac{t}{2}} (f(x^0) - f(x^*)) \tag{42}$$

Observe that at the $k$-th iteration of DCGS, the subproblem in Line 8 of DCGS has objective that is $l + \eta_k$ smooth and $\eta_k$ strongly convex, hence we could bound the LO of
each agent by:

\[
\mathcal{O}
\left(\sum_{k=1}^{N} \frac{(L + \eta_k) m D^2}{\eta_k W^2} \log \left(\frac{1}{\epsilon_i} \right)\right)
\]

\[
\leq \mathcal{O}
\left(\sum_{k=1}^{N} \frac{(L + \|L\|) m D^2}{\|L\| W^2} \log \left(\frac{m N}{\max\left(\|x^0 - x^*\|, \|y^0\|^2\right)} \right)\right)
\]

\[
= \tilde{\mathcal{O}}(\frac{D^2 \log m (l + \|L\|) \max(\|x^0 - x^*\|, \|y^0\|^2)}{\epsilon})
\]  \hfill (43)

\[\square\]

### C.2 Proof of Corollary 4

**Proof.** From Theorem 2, we know that to get an \(\epsilon\)-optimal solution, we need at most

\[N = \mathcal{O}\left(\sqrt{\frac{\max(\|x^0 - x^*\|^2, \|y^0\|^2)}{\epsilon}}\right)\]

number of outer iteration of DCGS. Following the same argument as in (43), and note that the objective in subproblem \(\min_{x_i \in X} \phi^k_i(x_i)\) is \(l + \eta_k\) smooth and \(u + \eta_k\) strongly convex, we could bound the LO complexity of each agent by:

\[
\mathcal{O}
\left(\sum_{k=1}^{N} \sum_{i=1}^{m} \frac{(L + \eta_k) D^2}{(u + \eta_k) W^2} \log \left(\frac{1}{\epsilon_i} \right)\right)
\]

\[
\leq \mathcal{O}
\left(\sum_{k=1}^{N} \frac{m L D^2}{u W^2} \log \left(\frac{m N k}{\max(\|x^0 - x^*\|, \|y^0\|^2)} \right)\right)
\]

\[
= \tilde{\mathcal{O}}(\frac{m L D^2}{u W^2} N \log(Nm))
\]

\[
= \tilde{\mathcal{O}}(\frac{D^2 \log m \|x^0 - x^*\|}{\sqrt{u} \sqrt{\epsilon}})
\]  \hfill (44)

\[\square\]