Multi-Timescale Primal Dual Hybrid Gradient with Application to Distributed Optimization

Junhui Zhang[1] and Patrick Jaillet[1,2]

[1]Operations Research Center, MIT
[2]Department of Electrical Engineering and Computer Science, MIT

Abstract

We propose two variants of the Primal Dual Hybrid Gradient (PDHG) algorithm for saddle point problems with block decomposable duals, hereafter called Multi-Timescale PDHG (MT-PDHG) and its accelerated variant (AMT-PDHG). Through novel mixtures of Bregman divergence and multi-timescale extrapolations, our MT-PDHG and AMT-PDHG converge under arbitrary updating rates for different dual blocks while remaining fully deterministic and robust to extreme delays in dual updates.

We further apply our (A)MT-PDHG, augmented with the gradient sliding techniques introduced in [28, 27], to distributed optimization. The flexibility in choosing different updating rates for different blocks allows a more refined control over the communication rounds between different pairs of agents, thereby improving the efficiencies in settings with heterogeneity in local objectives and communication costs. Moreover, with careful choices of penalty levels, our algorithms show linear and thus optimal dependency on function similarities, a measure of how similar the gradients of local objectives are. This provides a positive answer to the open question whether such dependency is achievable for non-smooth objectives [4].

1 Introduction

We study the following saddle point problem with block decomposable dual variables and objectives:

$$\min_{X\in\mathcal{X}}\max_{y_s\in\mathbb{R}^{n_s},\ s\in[S]} F(X)+\sum_{s=1}^{S}\left(\langle K_sX,y_s\rangle-R_s^*(y_s)\right), \tag{$\mathcal{P}_b$}$$

where $\mathcal{X}\subset\mathbb{R}^d$ is a non-empty convex set and $F:\mathcal{X}\to\mathbb{R}$ is a convex function such that there exists $\mu, M\geq 0$,

$$\frac{\mu}{2}\|X-X'\|^2\leq F(X)-F(X')-\langle F'(X'),X-X'\rangle\leq M\|X-X'\|,\quad \forall X,X'\in\mathcal{X}, \tag{1}$$

where $F':\mathcal{X}\to\mathbb{R}^d$ is a subgradient oracle, i.e. for each $X'\in\mathcal{X}$, $F'(X')\in\partial F(X')$ is a subgradient. For instance, when $\|F'\|_*\leq M'$, then $M=2M'$ holds.[1] For each $s\in[S]$, $K_s\in\mathbb{R}^{n_s\times d}$ is a matrix, $R_s:\mathbb{R}^{n_s}\to\overline{\mathbb{R}}$ is a proper, convex, and lower-semicontinuous function, and $R_s^*:\mathbb{R}^{n_s}\to\overline{\mathbb{R}}$ is its Fenchel conjugate, defined as $R_s^*(y_s)=\sup_{y_s'\in\mathbb{R}^{n_s}}\langle y_s,y_s'\rangle-R_s(y_s')$.

Saddle point problems of the form ($\mathcal{P}_b$) with one block ($S=1$) have been widely studied due to its applications in various problems including linear programming [3], distributed optimization [28], and inverse problems such as image denoising [15, 14], to name a few. For $S>1$, stochastic algorithms which update random subsets of dual blocks have been proposed [13, 42].

In this work, building upon the Primal Dual Hybrid Gradient (PDHG) algorithms[15, 14], we propose Multi-Timescale Primal Dual Hybrid Gradient (MT-PDHG) and its accelerated variant (AMT-PDHG) where

[1]This is because $F(X')\geq F(X)+\langle F'(X),X'-X\rangle$, and so using Cauchy–Schwarz inequality

$$F(X)-F(X')-\langle F'(X'),X-X'\rangle\leq\langle F'(X)-F'(X'),X-X'\rangle\leq 2M\|X-X'\|.$$

dual blocks are updated periodically at potentially different rates. Our algorithms achieve the following three properties.

1. *Flexible*. (A)MT-PDHG converge under arbitrary updating rates for the dual blocks.
2. *Deterministic*. The updating schedules for the dual blocks are deterministic and periodic.
3. *Robust*. The convergence rates depend on the average, instead of the maximum, of the updating rates, thereby are robust to extreme delays (i.e. large updating rates).

The motivation behind this work is distributed optimization, a branch of optimization where multiple agents, each having access to only partial information about the (global) objective, work together to solve the global problem. As an example, in distributed empirical risk minimization for machine learning, the global objective function is the sum of local loss functions, each depending on the local dataset which is only available to one agent [5, 11, 2, 4, 25]. Examples of other applications include power system control [32, 33], multi-robot system control [12, 33, 21, 44], and signal processing [10, 29, 40]. More precisely, we study the following distributed optimization problem

$$\min_{x\in\overline{\mathcal{X}}} \sum_{v\in V} f_v(x) \tag{$\mathcal{P}_d$}$$

where $V = [m]$ represents m agents, $\overline{\mathcal{X}} \subset \mathbb{R}^{\overline{d}}$ is a nonempty, closed convex set, and $x^* \in \overline{\mathcal{X}}$ is an optimal solution to $(\mathcal{P}_d)$.

From $(\mathcal{P}_b)$ to lifted space reformulation of $(\mathcal{P}_d)$. We consider a lifted space reformulation of $(\mathcal{P}_d)$ – $(\mathcal{P}_d^{lift})$ in Section 4.1 – where agent v maintains and updates x_v, a local version of the decision variable x. The objective then becomes $\sum_{v\in V} f_v(x_v)$, with additional consensus constraints encouraging $x_v \approx x_{v'}$ for all $v, v' \in V$. The classical PDHG algorithm, when implemented in the distributed setting, requires communication only when dual variables are updated.

From $(\mathcal{P}_d)$ and $(\mathcal{P}_d^{lift})$ to algorithm design goals for $(\mathcal{P}_b)$. We target at distributed optimization under: (1) heterogeneity in local objectives and communication costs, and (2) communication bottlenecks with periodic connectivity constraints.

As an example of heterogeneity, in distributed empirical risk minimization, local loss functions are different due to randomness in the local datasets and variation of data distributions [2, 26, 43, 56, 48, 4]; the costs of communication could depend on factors such as the distance between agents, methods of communication, and amounts of data sent [8, 10, 41, 50, 51]. The heterogeneity makes it desirable to design algorithms allowing more *flexible* control over the numbers of communication rounds among different subsets of agents, which translates to the numbers of updates applied to each dual blocks. Existing block coordinate descent algorithms allow some level of control on this by using different selection probabilities for different blocks. However, stochastic algorithms could be impractical or inefficient due to factors such as unpredictability, random memory access [47], sampling overhead [18], and physical constraints on connectivity between agents [34].

We consider a "rate based design", where communication is scheduled at different rates between different pairs of agents, *deterministically*. In practice, the rates could be either picked by users and/or subject to physical, bandwidth, and/or energy constraints [55]. This brings in one challenge: how to ensure the convergence of the algorithms under periodic communication at different rates. Another challenge, and also our third desideratum, is to ensure that the performance of the algorithms should not be influenced by stragglers (dual blocks which are updated infrequently), meaning that the algorithms should be *robust* to extreme values in the updating rates.

Results. Our (A)MT-PDHG achieves the above three goals. To ensure convergence, we use careful mixtures of Bregman divergence and novel multi-timescale extrapolation. For the distributed optimization problem considered, to find an ϵ-suboptimal solution, the complexities of our algorithms achieve optimal dependency on ϵ: MT-PDHG needs $O(\overline{r}A/\epsilon)$ communication rounds and $O(\overline{r}/\epsilon^2)$ subgradient steps for Lipchitz objectives, and AMT-PDHG needs $O(\overline{r}A/\sqrt{\epsilon\mu})$ communication rounds and $O(\overline{r}/(\epsilon\mu))$ subgradient steps if the objectives are also μ-strongly convex. Here, $\overline{r}$ measures the "average rate of updates" for dual blocks, and A measures similarities between (subgradients of) local functions. In fact, the linear dependency

of communication rounds on A is optimal [4], thereby providing a positive answer to the open question whether such dependency is achievable for non-smooth objectives [4].

Numerical experiments for linear programming and support vector machine problem with regularized hinge losses confirm the effectiveness of our algorithms and demonstrate the above dependence on $\overline{r}$ and A.

1.1 Related works

Primal-Dual Hybrid Gradient and its block variant. Our algorithms are built upon the Primal-Dual Hybrid Gradient (PDHG) algorithms [14, 15]. For problems with block-decomposable duals, block-coordinate descent type of variants have been proposed, such as the Stochastic Primal-Dual Coordinate (SPDC) [57] and the Stochastic-PDHG (S-PDHG)[1, 13], to name a few. These algorithms update *random* subsets of the dual blocks at each iteration, where all blocks have strictly positive probability of being selected. Although the $O(1/k)$ rate of convergence still holds, due to the randomness, the convergence is only shown for the *expected* objective value suboptimality (for SPDC) or *expected* duality gap (for S-PDHG), and could have unsatisfactory performance due to large variance. Moreover, updating random subsets could potentially be inefficient – due to reasons such as random memory access [47] and potential overhead in computing sampling distributions [18] – or even impossible due to physical constraints – such as in distributed settings, blocks available for updates are subject to network connectivity and communication constraints [34].

Although deterministic block coordinate descent for convex optimization has been shown to converge, such as under the cyclic updating rule [47, 54], to the best of our knowledge, the *multi-timescale* updating rule we propose is the first *deterministic* block updating rule for PDHG with separable duals, such that different blocks could be updated different numbers of times, and the duality gap converges deterministically at the optimal rate.

As the dual blocks are not updated at each global iteration, they introduce "outdated" information to the dynamics. Most existing asynchronous optimization algorithms achieve convergence rates which depend on the *maximum delay* [37, 30, 38], with the exception of [17, 6], whose convergence rates depend on the average and the quantiles of delays, respectively. However, [17, 6] are designed for *convex optimization*. As a comparison, our (A)MT-PDHG are designed for *saddle point problems* and have convergence rates which depend on the *averages instead of maximum* of the updating rates, making them robust against extreme delays in the dual updates.

Non-smooth distributed optimization. Since the seminal works [10, 49], numerous algorithms have been proposed for non-smooth distributed optimization under various settings, and we refer readers to surveys such as [5, 35, 24]. For the function class of Lipschitz, non-smooth, convex objectives, most of these algorithms fall into the following two categories: subgradient based and dual based [28]. Subgradient based algorithms such as the incremental gradient method [9], decentralized subgradient method [36], and the dual averaging [19] usually require $O(1/\epsilon^2)$ rounds of communication, each followed by one gradient step. Within the function class, this achieves the optimal subgradient oracle complexity, but is suboptimal with respect to the communication rounds: as proven by [4, 41], the communication rounds needed is $O(1/\epsilon)$.

As a comparison, dual based algorithms, which dualize the consensus constraints, usually have better communication complexity: $O(1/\epsilon)$ rounds are needed for distributed ADMM [7, 52] and the decentralized communication sliding (DCS) [28], as examples. However, each round of communication is followed by optimization of Lagrangians or proximal updates, performed locally by each agent. To make the overall algorithm first-order, [28] proposes the Communication Sliding (CS) procedure, which approximates the proximal updates through $O(1/\epsilon)$ steps of (local) mirror descent, thereby achieving the $O(1/\epsilon^2)$ subgradient oracle complexity. The CS procedure has roots in the gradient sliding technique [27], which can save gradient computation for the smooth component when the objective involves a smooth and a non-smooth component.

For the class of strongly convex objectives, DCS can be accelerated, needing $O(1/\sqrt{\epsilon})$ rounds of communication and $O(1/\epsilon)$ gradient steps in total, both achieving the theoretical optimal [28]. In this work, due to the different time scales, we generalize the CS procedure for problems involving a mixture of Bregman divergences. In addition, we point out that for problems with *smooth* objectives, Local SGD – which applies gradient steps locally but communicates only once in a while – has been studied under various settings [58, 46, 53].

Lower bounds on communication. In [4], it is shown that for distributed convex optimization, $O(1/\epsilon)$ rounds of communication are needed for 1-Lipschitz objectives, and $O(1/\sqrt{\mu\epsilon})$ rounds are needed

when the objectives are also μ-strongly convex. These lower bounds are achieved by splitting a "chain like" objective into two, each given to one agent. [41] extends these results to a decentralized, network setting and shows the dependence of the lower bounds on the network diameter and communication delay. [50] provides lower bound and (nearly) optimal algorithm for a different setup, where distributed agents have stochastic first order oracles to the same smooth nonconvex objective, but computation and communication speeds are bounded and different for different edges and agents. Apart from the round complexity, [51] shows a dimension-dependent lower bound on the bit-complexity of communication.

In addition, motivated by distributed training in machine learning, communication lower and upper bounds have been established using function similarities [4, 43, 2, 48, 26, 22, 23]: for instance, in (distributed) empirical risk minimization, the local loss functions have the same functional form but use different subsets of data, thereby inheriting the similarity in data. In [4], function similarities are measured using norms of the differences in (sub)gradients (and Hessians if exist), and a communication round lower bound linear in this measure is shown for convex Lipschitz objectives and strongly convex objectives. Known algorithms that take advantage of function similarities usually require additional assumptions such as strong convexity and smoothness [2, 43, 56, 48, 26, 22, 23]. As pointed out in [4], there is no known algorithm which achieve these communication round lower bounds for non-smooth convex objectives. In this work, we formalize the notion of function similarity for non-smooth convex objectives (Definition 4.1), and show that the communication round complexity for our (A)MT-PDHG indeed achieve these lower bounds, thereby answering [4]'s open question positively.

1.2 Contributions

We propose multi-timescale PDHG for saddle point problems with block-decomposable duals, where different dual blocks are updated at different rates.

- To ensure convergence with arbitrary updating rates, we propose novel multi-timescale extrapolation steps for the dual updates (5) and mixtures of Bregman divergence for the primal updates (6). The duality gaps of our algorithms converge at the optimal rates: $O(1/k)$ and $O(1/k^2)$ for general and strongly convex objectives, respectively, despite the potentially outdated information in the dual.
- To the best of our knowledge, for the saddle point problems considered, our multi-timescale PDHG algorithms are the first *deterministic* updating mechanisms which still *allow different (arbitrary) updating rates for different dual blocks*.
- We quantify how updating rates influence convergence: the duality gaps show linear dependencies on *weighted averages* of the updating rates (Corollary 3.3) and its square (Corollary 3.5) for general and strongly convex objectives, respectively. Thus, our algorithms are *robust to the maximum updating rates*.[2]

We apply our algorithms to convex non-smooth distributed optimization. More specifically, we propose relaxing the consensus constraints via *generic convex, block-decomposable penalty functions* (Lemma 4.3 and Corollary 4.1), which generalize the *characteristic-function penalty* used in prior work. This yields saddle-point formulations ($\mathcal{P}_d^{lift}$) with block-decomposable dual variables, enabling our multi-timescale PDHG method, which is particularly well suited to settings with heterogeneous communication costs among agents.

Moreover, we show that with proper choices of the penalties, the communication round complexities of our algorithms have linear, and thus *optimal*, dependency on similarities between gradients of the local objectives. This provides positive answers to the open question whether the theoretical communication round lower bounds proposed in [4] can be attained.

1.3 Roadmap

In Section 2, we present additional details and assumptions about the saddle point problem ($\mathcal{P}_b$). Then, in Section 3.1, we propose our multi-timescale PDHG for ($\mathcal{P}_b$), and in Sections 3.2 and 3.3, we present the convergence properties of our algorithms with and without strong convexity, respectively.

[2] We call the inverses of frequencies the rates. Thus, the largest updating rate corresponds to the smallest frequency.

In Section 4, we apply our multi-timescale PDHG to distributed optimization problems: we first show that $(\mathcal{P}_d)$ can be reformulated in the lifted space in the form of $(\mathcal{P}_b)$ (Section 4.1) and propose a set of conditions on the consensus constraints (Section 4.2) and penalties (Section 4.3). The requirements on the penalties motivate the definition of function similarities (Definition 4.1). Then, we describe how multi-timescale PDHG can be applied and the communication involved (Section 4.4), and provide the convergence results in Sections 4.5 and 4.6.

Numerical experiments are provided in Section 5.

2 Setup

For the problem $(\mathcal{P}_b)$, we assume $\mathbb{R}^d$ is equipped with a norm $\|\cdot\|$ not necessarily generated by the Euclidean inner product, and we equip $\mathcal{X}$ with a distance generating function[3] $w_X : \mathcal{X} \to \mathbb{R}$ with modulus 1.

We further denote $\mathcal{Y}_s := \mathsf{dom}(R_s^*) = \{y_s \in \mathbb{R}^{n_s},\ R_s^*(y_s) < \infty\}$ as the domain of R_s^*, and define $R : \mathbb{R}^n \to \overline{\mathbb{R}}$ as $R(Y) = \sum_{s=1}^S R_s(y_s)$. It's easy to see that $R^*(Y) = \sum_{s=1}^S R_s^*(y_s)$ and $\mathcal{Y} := \mathsf{dom}(R^*) = \prod_{s=1}^S \mathsf{dom}(R_s^*)$. Similarly, for each $s \in [S]$, we assume that $\mathbb{R}^{n_s}$ is equipped with a norm $\|\cdot\|$ not necessarily generated by the Euclidean inner product, and we equip $\mathsf{dom}(R_s^*)$ with distance generating function $w_{y_s} : \mathsf{dom}(R_s^*) \to \mathbb{R}$ with modulus 1, and $w_Y(Y) := \sum_{s=1}^S w_{y_s}(y_s)$.

For convenience, we define $K : \mathbb{R}^d \to \mathbb{R}^n$ as $(KX)_s = K_s X$ for each $s \in [S]$. Thus, $(\mathcal{P}_b)$ can be compactly written as

$$\min_{X\in\mathcal{X}} \max_{Y\in\mathbb{R}^n} F(X) + \langle KX, Y\rangle - R^*(Y). \tag{$\mathcal{P}_c$}$$

In addition, since R_s is convex and lower-semicontinuous, $R_s = R_s^{**}$ (Theorem 11.1 [39]), and importantly,

$$R_s(K_s X) = \sup_{y_s\in\mathbb{R}^{n_s}} \langle K_s X, y_s\rangle - R_s^*(y_s).$$

Thus, $(\mathcal{P}_b)$ is also equivalent to the following primal-only formulation:

$$\min_{X\in\mathcal{X}} F(X) + \sum_{s=1}^S R_s(K_s X). \tag{$\mathcal{P}_p$}$$

In the rest of the work, when there is no confusion on the domain of the function, we abbreviate w_X, w_Y, w_{y_s} as w and the associated Bregman divergence as D. We also make the following assumption.

Assumption 2.1. *For any $\overline{y}_s \in \mathsf{dom}(R_s^*)$, $g \in \mathbb{R}^{n_s}$ the following problem can be solved exactly:*

$$\min_{y_s\in\mathsf{dom}(R_s^*)} R_s^*(y_s) + \langle g, y_s\rangle + D(y_s, \overline{y}_s).$$

For any $g \in \mathbb{R}^d$, the following problem can be solved exactly:

$$\min_{X\in\mathcal{X}} \langle g, X\rangle + w(X).$$

Performance measure. We use the duality gap to measure the performance of $Z = (X, Y)$. More precisely, we define $G : \mathcal{Z} \times \mathcal{Z} \to \overline{\mathbb{R}}$ where $\mathcal{Z} = \mathcal{X} \times \mathbb{R}^n$ as

$$G(X, Y; X', Y') := \langle KX, Y'\rangle + F(X) - R^*(Y') - \{\langle KX', Y\rangle + F(X') - R^*(Y)\}, \tag{2}$$

and we will provide upper bounds on $G(Z) := \sup_{Z'\in\mathcal{Z}} G(Z; Z')$ where Z is the output of our algorithms.

[3]For a convex closed set $\mathcal{S}$, a function $w : \mathcal{S} \to \mathbb{R}$ is a distance generating function [20] with modulus $\nu > 0$ w.r.t. $\|\cdot\|$ if w is continuously differentiable and

$$\langle x - z, \nabla w(x) - \nabla w(z)\rangle \geq \nu\|x - z\|^2, \quad \forall x, z \in \mathcal{S}.$$

The Bregman divergence generated by w is defined as $D_w(x, z) := w(x) - w(z) - \langle \nabla w(z), x - z\rangle$.

3 Multi-timescale primal-dual updates

To solve the saddle point problem $(\mathcal{P}_c)$, one of the most popular algorithms is the Primal-Dual Hybrid Gradient (PDHG) algorithm[14, 28, 13], which updates the primal variable X and the dual variable Y iteratively. At iteration $k = 0, 1, \ldots$,

$$\widetilde{X}^k = X^{k-1} + \alpha_k (X^{k-1} - X^{k-2}) \tag{3a}$$

$$Y^k = \operatorname*{argmin}_{Y \in \mathbb{R}^n} R^*(Y) + \langle -K\widetilde{X}^k, Y\rangle + \tau_k D(Y, Y^{k-1}) \tag{3b}$$

$$X^k = \operatorname*{argmin}_{X \in \mathcal{X}} F(X) + \langle K^* Y^k, X\rangle + \eta_k D(X, X^{k-1}), \tag{3c}$$

where $\tau_k, \eta_k > 0$ are parameters depending on the operator norm of K, and $\alpha_k \in [0, 1]$. The convergence rates of PDHG under various assumptions of the objective functions F and R have been well established [14]. For instance, for general convex F and R, the duality gap (of the ergodic mean) converges at the rate of $O(1/k)$; further, if F is strongly convex, the accelerated rate of $O(1/k^2)$ can be achieved.

Approximation to (3c) **through gradient sliding and auxiliary primal sequence.** For a generic (convex, potentially non-linear) objective function F, it's sometimes unreasonable to assume that one can find the exact minimizer in the update (3c). Indeed, in Assumption 2.1, we only assume that $\langle g, X\rangle + w(X)$ can be exactly minimized. Thus, [27, 28] propose and use gradient sliding techniques, which approximate the minimizer through multiple iterations of mirror descent. In addition, to ensure the convergence of the overall PDHG algorithm with inexact updates, an auxiliary sequence $\widehat{X}^k$ is constructed, and the right hand side of (3a) is replaced with $X^{k-1} + \alpha_k(\widehat{X}^{k-1} - X^{k-2})$.

Block-decomposable dual updates. In $(\mathcal{P}_b)$, both $R^*(Y) = \sum_{s=1}^S R_s^*(y_s)$ and $D(Y, Y^{k-1}) = \sum_{s=1}^S D(y_s, y_s^{k-1})$ are block-decomposable. Thus, the dual update (3b) is also block-decomposable:

$$y_s^k = \operatorname*{argmin}_{y \in \mathbb{R}^{n_s}} R_s^*(y_s) + \langle -K_s\widetilde{X}^k, y_s\rangle + \tau_k D(y_s, y_s^{k-1}), \quad s = 1, \ldots, S. \tag{4}$$

This makes block-coordinate descent type of algorithms possible. As an example, S-PDHG [13] updates a random subset of the dual blocks at each iteration. The flexibility in choosing the sampling distribution allows one to control the frequency of updates of different blocks. However, due to the randomness, the $O(1/k)$ rate of convergence is shown only for the expected duality gap.

To maintain the *deterministic convergence guarantee* as well as the *flexibility in choosing the number of updates applied to each dual block*, we propose a multi-timescale updating mechanism for $(\mathcal{P}_b)$, where different dual blocks are updated at potentially different rates. More precisely, denoting the global time using $k = 0, 1, \ldots$, then the dual block y_s is updated only at iteration $k = 0, r_s, 2r_s, \ldots$ for some positive integer r_s and remains fixed for all other iterations. Due to this multi-timescale mechanism, two challenges arise.

1. Information delay in y_s^k. In the primal update (3a), the term $K^*Y^k = \sum_{s=1}^S K_s^* y_s^k$ depends on y_s^k, yet $y_s^k = y_s^{\lfloor k/r_s\rfloor \times r_s}$ contains information only till iteration $\lfloor k/r_s\rfloor \times r_s \le k$, which could be "outdated". To mitigate the negative effect of "information delay", we propose using *mixtures of Bregman divergences* in (3c) to control how fast the primal sequence varies ((6)).

2. Multi-timescale information aggregation of X^k. The dual updates (3b) and (4) are defined for each global time k, and $\widetilde{X}^k$ depends on X^{k-2} and X^{k-1} only. In the multi-timescale setting, one can expect that $\widetilde{X}^k$ should depend on primal sequences over longer intervals (the length of which could depend on r_s), and could be different for different dual blocks. We propose *multi-timescale extrapolation* which aggregates $X^{(k-2r_s):(k-1)}$ for block s ((5)).

In the rest of this section, we state the exact updating procedure in Section 3.1, and provide convergence results in Sections 3.2 and 3.3 for general convex and strongly convex F respectively.

3.1 Multi-timescale updating procedure

In our multi-timescale updating procedure, we use $k = 0, 1, \ldots, N$ to denote the global time. Motivated by gradient sliding procedures in [28], we also allow approximate minimizers to the primal updates, and keep track of the pair $(X^k, \widehat{X}^k)$, which is assumed to satisfy the condition (7). This can be achieved using a generalized gradient sliding technique (Appendix A).

Initialization. We assume access to initializations $X^{init} \in \mathcal{X}$ and $y_s^{init} \in \mathcal{Y}_s$ for all $s \in [S]$, and we initialize $X^{k'} = \widehat{X}^{k'} = X^{init}$ and $y_s^{k'} = y_s^{init}$ for all $k' < 0$.

Dual updates. We associate each dual y_s with a rate $r_s \in \mathbb{N}$ and a local time $i_s = 0, 1, \ldots, N_s - 1$, such that $N + 1 = r_s N_s$. For $k = 0, 1, \ldots, N$, y_s remains dormant unless $k = r_s i_s$ for some $i_s \in \{0, 1, \ldots, N_s - 1\}$, where $y_s^{i_s}$ is computed as follows (which replaces (3a) and (3b)):

$$\widetilde{X}_s^{i_s} = \alpha_{s,i_s} \left(\sum_{k'=r_s i_s - r_s}^{r_s i_s - 1} \theta_{k'} (\widehat{X}^{k'} - X^{k'-r_s}) \right) + \sum_{k'=r_s i_s - r_s}^{r_s i_s - 1} \theta_{k'+r_s} X^{k'}, \tag{5a}$$

$$y_s^{i_s} = \operatorname*{argmin}_{y_s \in \mathbb{R}^{n_s}} \langle -\frac{1}{\sum_{k'=r_s i_s}^{r_s i_s + r_s - 1} \theta_{k'}} K_s \widetilde{X}_s^{i_s}, y_s \rangle + R_s^*(y_s) + \tau_{s,i_s} D(y_s, y_s^{i_s - 1}). \tag{5b}$$

Further, we denote $\overline{y}_s^k = y_s^{\lfloor k/r_s \rfloor}$, i.e. value of the dual block y_s at the global time k, and we abbreviate $\overline{Y}^k = (\overline{y}_s^k)_{s \in [S]}$.

Primal updates. First, for each k, we define

$$\Phi^k(X) := F(X) + \langle K^* \overline{Y}^k, X \rangle + \sum_{s=1}^{S} \eta_{k,s} D(X, X^{k-r_s}). \tag{6}$$

All primal variables are updated at each global time as *approximate* minimizer to (6), such that there exists some $\delta_k : \mathcal{X} \to \mathbb{R}$ which could depend on X^k, X^{k-1} and other algorithm parameters, the following is satisfied:

$$\Phi^k(\widehat{X}^k) \le \Phi^k(X) - (\frac{\mu}{C} + \sum_{s=1}^{S} \eta_{k,s}) D(X, X^k) + \delta_k(X), \quad \forall X \in \mathcal{X}, \tag{7}$$

where we assume (1) holds for some $\mu \ge 0$, and $D(X, X') \le \frac{C}{2}\|X - X'\|^2$ for some $0 < C \le \infty$.

In case Φ^k can be minimized exactly, say $F(X) = \langle g_F, X \rangle$ for some $g_F \in \mathbb{R}^d$ is a linear function, we can take $\widehat{X}^k = X^k = \operatorname{argmin}_{X \in \mathcal{X}} \Phi^k(X)$ to be the exact minimizer, and then we can take $\delta_k(X) = 0$ (Lemma A.1). Nevertheless, in (7), we also allow inexact minimizer. For generic convex objectives, this can be found through a generalization of the gradient sliding technique in [28]: we extend this technique from $S = 1$ to $S \ge 1$ (see Appendix A for more details). More concretely, following Corollary A.1, the following results hold.

Corollary 3.1. *Consider the following updates using GS, the generalized gradient sliding procedure in Algorithm 3, where $T_k \in \mathbb{N}$ and $\eta_k = \sum_{s=1}^{S} \eta_{k,s} > 0$, where $\eta_{k,s} \ge 0$,*

$$(X^k, \widehat{X}^k) = GS(F, \mathcal{X}, D, T_k, (\eta_{k,s})_{s \in [S]}, (X^{k-r_s})_{s \in [S]}, K^* \overline{Y}^k, X^{k-1}). \tag{8}$$

Assume that (1) holds with some $\mu \ge 0$, then with $\lambda_t = t + 1$ and $\beta_t = \frac{t}{2}$ for $t \ge 1$, (7) holds with

$$\delta_k(X) = \frac{2\eta_k}{T_k(T_k + 3)} \left(D(X, X^{k-1}) - D(X, X^k) \right) + \frac{4M^2}{\eta_k(T_k + 3)}.$$

Further, if (1) holds with some $\mu > 0$, and $D(X, X') \le \frac{C}{2}\|X - X'\|^2$ for some $C < \infty$, then with $\lambda_t = t$ and $\beta_t = \frac{(t+1)\mu}{2\eta_k C} + \frac{t-1}{2}$, (7) holds with

$$\delta_k(X) = \frac{2M^2/\eta_k}{T_k(T_k + 1)} \sum_{t=1}^{T} \frac{\lambda_t}{\beta_t}.$$

Final outputs. Denoting $\widehat{Z}^k = (\widehat{X}^k, \overline{Y}^k)$, then the output is

$$Z^N = (\sum_{k=0}^{N} \theta_k)^{-1} \sum_{k=0}^{N} \theta_k \widehat{Z}^k. \tag{9}$$

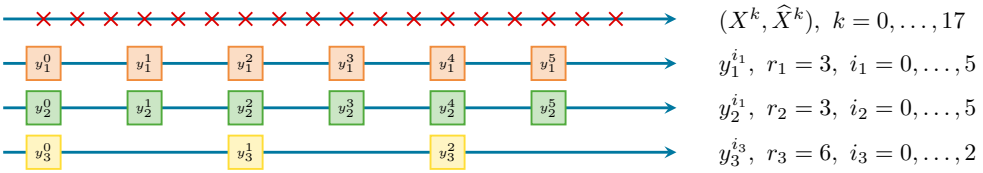


Figure 1: Updates for $S = 3$, $r_1 = r_2 = 3$ and $r_3 = 6$. Each marker represents one update: $(X^k, \widehat{X}^k)$ is updated at each global time $k = 0, 1, \ldots, 17$. If generalized gradient sliding is used, then this involves T_k iterations of mirror descent updates at iteration k. y_1 is updated at each local time $i_1 = 0, \ldots, 5$, i.e. global time $k = 0, 3, 6, \ldots, 15$, and similarly for y_2 and y_3.

Algorithm 1 (Accelerated) Multi-timescale PDHG

Input: $\{\alpha_{s,i_s}\}, \{\theta_k\}, \{\eta_{k,s}\}, \{\tau_{s,i_s}\}, \{r_s\}, X^{init}, Y^{init}$
Output: Primal dual pair Z^N
  Initialize $(X^{k'}, \widehat{X}^{k'}, Y^{k'}) \leftarrow (X^{init}, X^{init}, Y^{init})$ for all $k' < 0$
  for $k = 0, 1, \ldots, N$ **do** ▷ implicitly $i_s = \lfloor k/r_s \rfloor$ for all $s \in [S]$
    for $s \in [S]$ such that $k = 0 \pmod{r_s}$ **do**
      Dual update: compute $\widetilde{X}_s^{i_s}$ using (5a), then update $y_s^{i_s}$ using (5b)
    end for
    Primal update: compute $K^*\overline{Y}^k$ where $\overline{y}_s^k = y_s^{\lfloor k/r_s \rfloor}$ for $s \in [S]$, update $(X^k, \widehat{X}^k)$ satisfying (7)
  end for
  Compute Z^N using (9).

3.2 Convergence of the multi-timescale updates

In this section, we assume F is a generic convex function satisfying (1) for some $\mu \geq 0$. We propose conditions on the parameters of Algorithm 1 which ensure the convergence of the resulting multi-timescale updating procedure. The proof of convergence of Algorithm 1 follows a similar type of argument as the proof of convergence of PDHG [14] and the Decentralized Communication Sliding [28]: the primal updates (8) and dual updates control the following two terms ((10), (11)):

$$\left\{ \sum_{k=0}^{N} \langle K^*\overline{Y}^k, \widehat{X}^k - X \rangle + F(\widehat{X}^k) - F(X) \right\}$$
$$+ \left\{ \sum_{i_s=0}^{N_s-1} \langle -K_s \widetilde{X}_s^{i_s}, y_s^{i_s} - y_s \rangle + r_s (R_s^*(y_s^{i_s}) - R_s^*(y_s)) \right\}.$$

The above sum (approximately) matches the gap $\sum_{k=0}^{N} G(\widehat{Z}^k; Z)$ up to an additive term ((12))

$$\sum_{s=1}^{S} \sum_{k=0}^{N} \langle \widehat{X}^k - \widetilde{X}_s^{\lfloor k/r_s \rfloor}, K_s^*(y_s - \overline{y}_s^k) \rangle = \sum_{i_s=0}^{N_s-1} \langle \sum_{i=0}^{r_s-1} \widehat{X}^{r_s i_s + i} - \widetilde{X}_s^{i_s}, K_s^*(y_s - y_s^{i_s}) \rangle.$$

Notice that due to the different timescales for the duals, we bound the above terms *at dual time scales*: instead of controlling $\langle \widehat{X}^k - \widetilde{X}_s^{\lfloor k/r_s \rfloor}, K_s^*(y_s - y_s^{\lfloor k/r_s \rfloor}) \rangle$ for each k, we control the cumulative term (sum from $k = r_s i_s$ to $k = r_s(i_1 + 1) - 1$). With our choice of the $\widetilde{X}_s^{i_s}$ and the mixture terms used in primal proximal updates, the following results hold.

Theorem 3.1. *Assume that* (1) *holds with some* $M, \mu \geq 0$*, then with the following choice of parameters:* $\alpha_{s,i_s} = \alpha = 1$*,* $\theta_k = 1$*;* $\eta_{k,s} = \eta\rho_s$ *where* $\rho_s \geq 0$ *and* $\sum_{s=1}^S \rho_s = 1$*;* $\tau_s = \frac{2\widetilde{\kappa}_s^2}{\rho_s \eta}$ *where* $\widetilde{\kappa}_s := \sup_{\|y_s\| \leq 1} \|K_s^* y_s\|_*$*, denote* $\overline{r} = \sum_{s=1}^S r_s \rho_s$*, we have* $\forall Z \in \mathcal{Z} = \mathcal{X} \times \mathbb{R}^n$*,*

$$(N+1) \cdot G(Z^k; Z) \leq \eta \overline{r} D(X, X^{init}) - \eta D(X, X^N) + \sum_{k=0}^N \delta_k(X) + \sum_{s=1}^S \tau_s r_s \cdot \left\{ \frac{3}{2} D(y_s, y_s^{init}) - \frac{1}{2} D(y_s, y_s^{N_s - 1}) \right\}.$$

Proof of Theorem 3.1. **Primal update properties.** Summing (7) over k, and defining $\eta_{k,s} = 0$ for all $k < 0$ and $k \geq N+1$, we have

$$\begin{aligned}
&\sum_{k=0}^N \langle K^* \overline{Y}^k, \widehat{X}^k - X \rangle + F(\widehat{X}^k) - F(X) \\
&\leq \sum_{k=0-\max\{r_s\}}^N \left(\sum_{s=1}^S \eta_{k+r_s,s} \right) D(X, X^k) - \sum_{k=0}^N \sum_{s=1}^S \eta_{k,s} D(\widehat{X}^k, X^{k-r_s}) + \sum_{k=0}^N \left(\delta_k(X) - \eta_k D(X, X^k) \right) \\
&\leq \eta \overline{r} D(X, X^{init}) - \eta D(X, X^N) - \sum_{k=0}^N \sum_{s=1}^S \eta_{k,s} D(\widehat{X}^k, X^{k-r_s}) + \sum_{k=0}^N \delta_k(X), \qquad (10)
\end{aligned}$$

where the last step is because $X^k = X^{init}$ for all $k < 0$, and by our choice that $\eta_{k,s} = \eta\rho_s$ for $k = 0, 1, \ldots, N$,

$$\sum_{k=0-\max\{r_s\}}^{-1} \sum_{s=1}^S \eta_{k+r_s,s} \leq \eta \sum_{s=1}^S r_s \rho_s = \eta \overline{r}, \quad \sum_{s=1}^S \eta_{k'+r_s,s} \leq \eta (\sum_{s=1}^S \rho_s) = \eta, \ k' = 0, \ldots, N-1.$$

Dual update properties. By the updating rule for the dual, we have by Proposition 2 in [28] for any $y_s \in \mathbb{R}^{n_s}$,

$$\langle -\frac{1}{r_s} K_s \widetilde{X}_s^{i_s}, y_s^{i_s} - y_s \rangle + R_s^*(y_s^{i_s}) - R_s^*(y_s) \leq \tau_{s,i_s} \left(D(y_s, y_s^{i_s - 1}) - D(y_s, y_s^{i_s}) - D(y_s^{i_s}, y_s^{i_s - 1}) \right).$$

Thus, with $\tau_{s,i_s} = \tau_s$ for all i_s, summing over the above, we get

$$\begin{aligned}
&\sum_{i_s=0}^{N_s - 1} \langle -\frac{1}{r_s} K_s \widetilde{X}_s^{i_s}, y_s^{i_s} - y_s \rangle + R_s^*(y_s^{i_s}) - R_s^*(y_s) \\
&\leq \tau_s \left(D(y_s, y_s^{init}) - D_{w_s^y}(y_s, y_s^{N_s - 1}) \right) - \tau_s \cdot \sum_{i_s=0}^{N_s - 1} D(y_s^{i_s}, y_s^{i_s - 1}). \qquad (11)
\end{aligned}$$

Gap properties. Recall that for each $s \in [S]$, $\overline{y}_s^k = y_s^{\lfloor k/r_s \rfloor}$, thus we have

$$\begin{aligned}
\sum_{k=0}^N \left\{ \langle \widehat{X}^k, K_s^* y_s \rangle - \langle X, K_s^* \overline{y}_s^k \rangle \right\} &= \sum_{i_s=0}^{N_s - 1} \left\{ \sum_{i=0}^{r_s - 1} \langle \widehat{X}^{r_s i_s + i}, K_s^* y_s \rangle - r_s \langle X, K_s^* y_s^{i_s} \rangle \right\} \\
&= \sum_{i_s=0}^{N_s - 1} \langle \sum_{i=0}^{r_s - 1} \widehat{X}^{r_s i_s + i} - \widetilde{X}_s^{i_s}, K_s^* (y_s - y_s^{i_s}) \rangle \\
&\quad + \sum_{k=0}^N \langle \widehat{X}^k - X, K_s^* \overline{y}_s^k \rangle + \sum_{i_s=0}^{N_s - 1} \langle K_s \widetilde{X}_s^{i_s}, y_s - y_s^{i_s} \rangle. \qquad (12)
\end{aligned}$$

Recall that for $i_s = 0, 1, \ldots, N_s - 1$,

$$\widetilde{X}_s^{i_s} = \alpha\Big(\sum_{k'=r_s i_s - r_s}^{r_s i_s - 1} \widehat{X}^{k'} - \sum_{k'=r_s i_s - 2r_s}^{r_s i_s - r_s - 1} X^{k'}\Big) + \sum_{k'=r_s i_s - r_s}^{r_s i_s - 1} X^{k'}.$$

We first bound the first term in (12). Notice that for $i_s = 0, 1, \ldots, N_s - 1$, we have

$$\begin{aligned}
&\langle \sum_{i=0}^{r_s-1} \widehat{X}^{r_s i_s + i} - \widetilde{X}_s^{i_s}, K_s^*(y_s - y_s^{i_s})\rangle \\
&= \langle \sum_{i=0}^{r_s-1} (\widehat{X}^{r_s i_s + i} - X^{r_s(i_s-1)+i}) - \alpha \sum_{i=0}^{r_s-1} (\widehat{X}^{r_s(i_s-1)+i} - X^{r_s(i_s-2)+i}), K_s^*(y_s - y_s^{i_s})\rangle \\
&= \langle \sum_{i=0}^{r_s-1} (\widehat{X}^{r_s i_s + i} - X^{r_s(i_s-1)+i}), K_s^*(y_s - y_s^{i_s})\rangle \\
&\quad - \alpha\langle \sum_{i=0}^{r_s-1} (\widehat{X}^{r_s(i_s-1)+i} - X^{r_s(i_s-2)+i}), K_s^*(y_s - y_s^{i_s-1})\rangle \\
&\quad + \alpha\langle \sum_{i=0}^{r_s-1} (\widehat{X}^{r_s(i_s-1)+i} - X^{r_s(i_s-2)+i}), K_s^*(y_s^{i_s} - y_s^{i_s-1})\rangle.
\end{aligned}$$

Thus, with $\alpha = 1$, and recall that for $i_s = 0$, $i = 0, \ldots, r_s - 1$, $\widehat{X}^{r_s(i_s-1)+i} - X^{r_s(i_s-2)+i} = X^{init} - X^{init} = \mathbf{0}$, we have

$$\begin{aligned}
&\sum_{i_s=0}^{N_s-1} \langle \sum_{i=0}^{r_s-1} \widehat{X}^{r_s i_s + i} - \widetilde{X}_s^{i_s}, K_{s,v}^*(y_s - y_s^{i_s})\rangle \\
&= \langle \sum_{i=0}^{r_s-1} (\widehat{X}^{N - r_s + i} - X^{N - 2r_s + i}), K_s^*(y_s - y_s^{N_s-1})\rangle \\
&\quad + \sum_{i_s=1}^{N_s-1} \langle \sum_{i=0}^{r_s-1} (\widehat{X}^{r_s(i_s-1)+i} - X^{r_s(i_s-2)+i}), K_s^*(y_s^{i_s} - y_s^{i_s-1})\rangle \\
&\leq \sum_{i=0}^{r_s-1} \|\widehat{X}^{r_s(N_s-1)+i} - X^{r_s(N_s-2)+i}\| \cdot \|K_s^*(y_s - y_s^{N_s-1})\|_* \\
&\quad + \sum_{i_s=1}^{N_s-1} \sum_{i=0}^{r_s-1} \|\widehat{X}^{r_s(i_s-1)+i} - X^{r_s(i_s-2)+i}\| \cdot \|K_s^*(y_s^{i_s} - y_s^{i_s-1})\|_*
\end{aligned}$$

Thus, for any $\rho > 0$, we have

$$\begin{aligned}
&\sum_{i_s=0}^{N_s-1} \langle \sum_{i=0}^{r_s-1} \widehat{X}^{r_s i_s + i} - \widetilde{X}_s^{i_s}, K_{s,v}^*(y_s - y_s^{i_s})\rangle \\
&\leq \sum_{k=0}^{N} \frac{\rho}{2} \|\widehat{X}^k - X^{k - r_s}\|^2 + \frac{r_s \widetilde{\kappa}_s^2}{2\rho} \Big(\sum_{i_s=1}^{N_s-1} \|y_s^{i_s} - y_s^{i_s-1}\|^2 + \|y_s - y_s^{N_s-1}\|^2\Big). \qquad (13)
\end{aligned}$$

Bounding the gap. Thus, with (10), (11), and (13), we have the following upper bound on the gap

$$
\begin{aligned}
&\sum_{k=0}^{N} G(\widehat{X}^k, \overline{Y}^k; Z) \\
&= \sum_{k=0}^{N} \left\{ F(\widehat{X}^k) - R^*(Y) - F(X) + R^*(\overline{Y}^k) \right\} + \sum_{s=1}^{S} \sum_{k=0}^{N} \left\{ \langle K_s \widehat{X}^k, Y_s \rangle - \langle K_s X, \overline{Y}^k \rangle \right\} \\
&\leq \eta \overline{r} D(X, X^{init}) - \eta D(X, X^N) - \sum_{k=0}^{N} \sum_{s=1}^{S} \eta_{k,s} D(\widehat{X}^k, X^{k-r_s}) + \sum_{k=0}^{N} \delta_k(X) \\
&\quad + \sum_{s=1}^{S} \tau_s r_s \left\{ D(y_s, y_s^{init}) - D(y_s, y_s^{N_s-1}) - \sum_{i_s=0}^{N_s-1} D(y_s^{i_s}, y_s^{i_s-1}) \right\} \\
&\quad + \sum_{k=0}^{N} \sum_{s=1}^{S} \frac{\eta \rho_s}{2} \|\widehat{X}^k - X^{k-r_s}\|^2 + \sum_{s=1}^{S} \frac{r_s \widetilde{\kappa}_s^2}{2\eta\rho_s} \Big(\sum_{i_s=1}^{N_s-1} \|y_s^{i_s} - y_s^{i_s-1}\|^2 + \|y_s - y_s^{N_s-1}\|^2 \Big)
\end{aligned}
$$

where we take $\rho = \eta \rho_s$ in (11). Thus, with $\frac{\widetilde{\kappa}_s^2}{\rho_s \tau_s} \leq \frac{\eta}{2}$ for all $s \in [S]$, we have

$$
\begin{aligned}
\sum_{k=0}^{N} G(\widehat{X}^k, \overline{Y}^k; Z) &\leq \eta \overline{r} D(X, X^{init}) - \eta D(X, X^N) + \sum_{k=0}^{N} \delta_k(X) \\
&\quad + \sum_{s=1}^{S} \tau_s r_s \cdot \left\{ \frac{3}{2} D(y_s, y_s^{init}) - \frac{1}{2} D(y_s, y_s^{N_s-1}) \right\}.
\end{aligned}
$$

The result follows since Z^N is the ergodic mean of $(\widehat{X}^k, \overline{Y}^k)$ and F, R are convex. □

When the primal update (3c) is approximated through the generalized communication sliding procedure, from Corollary 3.1, we have the following results.

Corollary 3.2. *Under the conditions in Theorem 3.1, and assume that $(X^k, \widehat{X}^k)$ are constructed using the generalized communication sliding (8) with $\lambda_t = t + 1$, $\beta_t = t/2$, and $T_k = T \geq 1$. Then the following holds for all $Z \in \mathcal{Z}$*

$$
\begin{aligned}
&(N+1) \cdot G(Z^N; Z) \\
&\leq \eta \left\{ \frac{3}{2} \overline{r} D(X, X^{init}) - D(X, X^N) \right\} \\
&\quad + \frac{1}{\eta} \left\{ \sum_{s=1}^{S} \frac{\widetilde{\kappa}_s^2 r_s}{\rho_s} \left\{ 3 D(y_s, y_s^{init}) - D(y_s, y_s^{N_s-1}) \right\} + \frac{4M^2(N+1)}{T+3} \right\}
\end{aligned}
$$

Proof of Corollary 3.2. Recall that from Corollary 3.1, we have for $k = 0, 1, \ldots, N$, with $\eta_k = \eta$ and $T_k = T$

$$
\delta_k(X) = \frac{2\eta}{T(T+3)} \left(D(X, X^{k-1}) - D(X, X^k) \right) + \frac{4M^2}{\eta(T+3)}.
$$

Thus, summing over k, we have

$$
\sum_{k=0}^{N} \delta_k(X) = \frac{2\eta}{T(T+3)} \left(D(X, X^{init}) - D(X, X^N) \right) + \frac{4M^2(N+1)}{\eta(T+3)}
$$

The result follows from noticing that for any $T \geq 1$, $\frac{2}{T(T+3)} \leq \frac{1}{2} \leq \frac{\overline{r}}{2}$.

□

Corollary 3.3. *For $\widehat{X} \in \mathcal{X}$, assume that the following are finite:*

$$D(\widehat{X}, X^{init}) \le D^X < \infty, \qquad \sup_{y_s \in \mathsf{dom}(R_s^*)} D(y_s, y_s^{init}) \le D_s^y < \infty.$$

Under the conditions in Corollary 3.2, taking $\eta = (\sum_{s'=1}^S \widetilde{\kappa}_{s'}\sqrt{D_{s'}^y})\sqrt{\frac{8}{3D^X}}$, $\rho_s = \frac{\widetilde{\kappa}_s\sqrt{D_s^y}}{\sum_{s'=1}^S \widetilde{\kappa}_{s'}\sqrt{D_{s'}^y}}$, and $T \ge \lfloor \frac{4M^2(N+1)}{\overline{r}(\sum_{s=1}^S \widetilde{\kappa}_s\sqrt{D_s^y})^2} \rfloor$ where $\overline{r} := \sum_{s=1}^S r_s\rho_s$, we have

$$\sup_{Y' \in \mathbb{R}^n} G(Z^N; \widehat{X}, Y') \le \frac{2\sqrt{6}\overline{r} \cdot (\sum_{s=1}^S \widetilde{\kappa}_s\sqrt{D_s^y}) \cdot \sqrt{D^X}}{N+1}. \tag{14}$$

Proof of Corollary 3.3. From Corollary 3.2, we first notice that with $\rho_s = \frac{\widetilde{\kappa}_s\sqrt{D_s^y}}{\sum_{s'=1}^S \widetilde{\kappa}_{s'}\sqrt{D_{s'}^y}}$

$$\sum_{s=1}^S \frac{\widetilde{\kappa}_s^2 r_s D_s^y}{\rho_s} = \overline{r}(\sum_{s'=1}^S \widetilde{\kappa}_{s'}\sqrt{D_{s'}^y})^2.$$

Thus, we have

$$T+3 \ge \frac{4M^2(N+1)}{\overline{r}(\sum_{s=1}^S \widetilde{\kappa}_s\sqrt{D_s^y})^2} \implies \frac{4M^2(N+1)}{T+3} \le \sum_{s=1}^S \frac{\widetilde{\kappa}_s^2 r_s D_s^y}{\rho_s}$$

Thus, with the additional assumptions, we get

$$\begin{aligned}
\sup_{Y' \in \mathbb{R}^n} G(Z^N; \widehat{X}, Y') &\le (N+1)^{-1}\left\{\frac{3\eta D^X}{2}(\sum_{s=1}^S r_s\rho_s) + \frac{4}{\eta}(\sum_{s=1}^S \frac{\widetilde{\kappa}_s^2 r_s D_s^y}{\rho_s})\right\} \\
&= (N+1)^{-1}\left\{\frac{3\eta D^X}{2} \cdot \overline{r} + \frac{4}{\eta}\overline{r}(\sum_{s'=1}^S \widetilde{\kappa}_{s'}\sqrt{D_{s'}^y})^2\right\} \\
&= \frac{2\sqrt{6}\overline{r}(\sum_{s'=1}^S \widetilde{\kappa}_{s'}\sqrt{D_{s'}^y}) \cdot \sqrt{D^X}}{N+1}.
\end{aligned}$$

□

Discussion on the complexities. Corollary 3.3 implies that to find an ϵ-suboptimal solution, one can take

$$N = O(\frac{\overline{r} \cdot (\sum_{s=1}^S \widetilde{\kappa}_s\sqrt{D_s^y}) \cdot \sqrt{D^X}}{\epsilon}), \quad T = O(\frac{M^2\sqrt{D^X}}{\epsilon \sum_{s=1}^S \widetilde{\kappa}_s\sqrt{D_s^y}}),$$

making the total number of subgradient oracles to F

$$NT = O(\frac{M^2N^2}{\overline{r}(\sum_{s'=1}^S \widetilde{\kappa}_{s'}\sqrt{D_{s'}^y})^2}) = O(\frac{\overline{r}M^2D^X}{\epsilon^2}).$$

This agrees with [28] for the case when $r_s = 1$ for all s.

Discussion on the costs. To further illustrate the benefits of having different update frequencies for different duals, we analyze the "cost" of Algorithm 1. Precisely, we assume that the cost of one update to y_s is c_s. This can be used to model the computation costs of matrix-vector multiplications ($K_s\widetilde{X}_s^{i_s}$ in (5a) and updating the $K_s^*\overline{y}_s^k$ term in (6)). As another motivation, as will be seen in Section 4 and Algorithm 2, when applying our multi-timescale PDHG to distributed optimization problems, only one round of communication is needed for each update of y_s. Thus, the costs here can also represent the communication costs for every dual update.

Below, we consider the case when the total cost is additive. With the above N, the dual variable y_s is updated $O(\frac{\overline{r}}{r_s\epsilon})$ times, which is different for duals with different r_s. Thus, suppose one is allowed to choose

the update frequencies $\{r_s\}_{s\in[S]}$, to minimize the total cost to find an ϵ suboptimal solution, the following should be (approximately) minimized

$$O(\overline{r}\sum_{s=1}^{S}\frac{c_s}{r_s}) = O((\sum_{s=1}^{S}\rho_s r_s)\cdot(\sum_{s=1}^{S}\frac{c_s}{r_s})).$$

With $r_s \propto \sqrt{c_s/\rho_s}$[4], the above becomes $O((\sum_{s=1}^{S}\sqrt{c_s\rho_s})^2)$. As a comparison, the strategy where all $r_s = r'_0$ are the same has the cost $O(\sum_{s=1}^{S} c_s)$. By Cauchy–Schwarz inequality, $(\sum_{s=1}^{S}\sqrt{c_s\rho_s})^2 \le \sum_{s=1}^{S} c_s$, and the difference can be very large when $\{c_s\rho_s\}_{s\in[S]}$ are very different, thereby showing the benefit of optimizing the updating rates $\{r_s\}_{s\in[S]}$ when $\{c_s/(\widetilde{\kappa}_s\sqrt{D_s^y})\}_{s\in[S]}$ are heterogeneous.

The additive cost is motivated by resources consumption when sending messages along each edge. In general, the total cost can be an arbitrary set function of the set of duals updated. For instance, to model time required to send messages (in parallel) where total time depends on the largest time, the cost could be $\max_{s\in\mathcal{S}} c_s$. Thus, the flexibility in choosing the updating rates allows the algorithm users to adapt the rates to the cost structures, leading to potentially lower costs.

3.3 Accelerated convergence under strong convexity

With strong convexity of F, i.e. $\mu > 0$, the convergence rate can be improved from $1/N$ to $1/N^2$, with a different set of parameters. In the following, we present these results.

Theorem 3.2. *Assume that* (1) *holds with some $M, \mu > 0$. Further assume that $D(X, X') \le \frac{C}{2}\|X - X'\|^2$ for all $X, X' \in \mathcal{X}$ for some $1 \le C < \infty$. Let $\{\rho_s\}_{s\in[S]}$ be a distribution over $[S]$, $\overline{r} = \sum_{s=1}^{S} r_s\rho_s$ and similarly define $\overline{r^2}$ and $\overline{r^3}$.*

With $\alpha_{s,i_s} = 1$, $\theta_k = k + 2\overline{r^2}/\overline{r}$, $\eta_k = \frac{\mu}{2\overline{r}C}(k + \overline{r^2}/\overline{r})$, $\eta_{k,s} = \eta_k\rho_s$, $\tau_{s,i_s}(\sum_{k'=r_s i_s}^{r_s i_s + r_s - 1}\theta_{k'}) = \tau_s = \frac{\widetilde{\kappa}_s^2}{\rho_s}\cdot\frac{4r_s\overline{r^2}C}{\mu}$. We have $\forall Z \in \mathcal{Z} = \mathcal{X}\times\mathbb{R}^n$,

$$G(Z^N; Z) \le \frac{2}{N(N+1)}\left\{\frac{5(\overline{r^2}/\overline{r})^2)}{2C}D(X, X^{init}) + \sum_{k=0}^{N}\theta_k\delta_k(X) + \sum_{s=1}^{S}\tau_s D(y_s, y_s^{init})\right\}.$$

Proof of Theorem 3.2. **Primal update properties.** Taking a weighted sum of (7) over k, and defining $\eta_{k,s} = 0$ for all $k < 0$ and $k \ge N+1$, we have

$$\begin{aligned}
&\sum_{k=0}^{N}\theta_k\left\{\langle K^*\overline{Y}^k, \widehat{X}^k - X\rangle + F(\widehat{X}^k) - F(X)\right\}\\
&\le \sum_{k=0}^{N}\theta_k\left(\sum_{s=1}^{S}\eta_{k,s}\left(D(X, X^{k-r_s}) - D(\widehat{X}^k, X^{k-r_s})\right) - (\frac{\mu}{C} + \eta_k)D(X, X^k) + \delta_k(X)\right)\\
&\le \frac{5(\overline{r^2}/\overline{r})^2)}{2C}D(X, X^{init}) - \sum_{k=0}^{N}\theta_k\sum_{s=1}^{S}\eta_{k,s}D(\widehat{X}^k, X^{k-r_s}) + \sum_{k=0}^{N}\theta_k\delta_k(X),
\end{aligned}$$

where the last step is because the coefficients of the term $D(X, X^k)$ (denoting $\theta_k = \eta_k = 0$ for all $k \ge N+1$) for $k = 0, 1, \ldots, N$ is the following

$$\begin{aligned}
&\sum_{s=1}^{S}\theta_{k+r_s}\eta_{k+r_s,s} - (\frac{\mu}{C} + \eta_k)\theta_k\\
&\le \frac{\mu}{2\overline{r}C}\left\{\sum_{s=1}^{S}(k + r_s + 2\overline{r^2}/\overline{r})(k + r_s + \overline{r^2}/\overline{r})\rho_s - (k + \overline{r^2}/\overline{r} + 2\overline{r})(k + 2\overline{r^2}/\overline{r})\right\}\\
&= \frac{\mu}{2\overline{r}C}\left\{\left(k^2 + (2\overline{r} + \frac{3\overline{r^2}}{\overline{r}})k + (4\overline{r^2} + 2(\overline{r^2}/\overline{r})^2)\right) - \left(k^2 + (2\overline{r} + \frac{3\overline{r^2}}{\overline{r}})k + (4\overline{r^2} + 2(\overline{r^2}/\overline{r})^2)\right)\right\} = 0,
\end{aligned}$$

[4]Here and below, $\propto$ means (approximately) proportional to, i.e. there exists $r_0 \in \mathbb{R}$ such that $r_s \approx r_0\sqrt{c_s/\rho_s}$ for all $s \in [S]$.

Since $X^k = X^{init}$ for all $k < 0$, the coefficient for the term $D(X, X^{init})$ is

$$\sum_{s=1}^{S}\sum_{k=0}^{r_s-1}\eta_{k,s}\theta_k \le \frac{\mu}{2\overline{r}C}\sum_{s=1}^{S}\rho_s\cdot r_s(r_s+\frac{\overline{r^2}}{\overline{r}})(r_s+2\frac{\overline{r^2}}{\overline{r}}) = \frac{\mu(\overline{r^3}/\overline{r}+5(\overline{r^2}/\overline{r})^2)}{2C}.$$

Dual update properties. Similar to (11), we get

$$\begin{aligned}
&\sum_{i_s=0}^{N_s-1}\left\{\langle -K_s\widetilde{X}_s^{i_s}, y_s^{i_s}-y_s\rangle + (\sum_{k'=r_si_s}^{r_si_s+r_s-1}\theta'_k)(R_s^*(y_s^{i_s})-R_s^*(y_s))\right\}\\
&\le \sum_{i_s=0}^{N_s-1}\tau_{s,i_s}(\sum_{k'=r_si_s}^{r_si_s+r_s-1}\theta'_k)\left\{D(y_s,y_s^{i_s-1})-D(y_s,y_s^{i_s})-D(y_s^{i_s},y_s^{i_s-1})\right\}\\
&= \tau_s\left\{D(y_s,y_s^{init})-D(y_s,y_s^{N_s-1})-\sum_{i_s=0}^{N_s-1}D(y_s^{i_s},y_s^{i_s-1})\right\}.
\end{aligned} \tag{15}$$

Gap properties. Notice that for each $s\in[S]$, we have

$$\begin{aligned}
&\sum_{k=0}^{N}\theta_k\left\{\langle\widehat{X}^k, K_s^*y_s\rangle - \langle X, K_s^*\overline{y}_s^k\rangle\right\}\\
&= \sum_{i_s=0}^{N_s-1}\langle\sum_{i=0}^{r_s-1}\theta_{r_si_s+i}\widehat{X}^{r_si_s+i}-\widetilde{X}_s^{i_s}, K_s^*(y_s-y_s^{i_s})\rangle\\
&\quad+\sum_{k=0}^{N}\theta_k\langle\widehat{X}^k-X, K_s^*\overline{y}_s^k\rangle + \sum_{i_s=0}^{N_s-1}\langle K_s\widetilde{x}_s^{i_s}, y_s-y_s^{i_s}\rangle.
\end{aligned} \tag{16}$$

We first bound the first term in (12). With $\alpha_{s,i_s}=1$

$$\begin{aligned}
&\sum_{i_s=0}^{N_s-1}\langle\sum_{i=0}^{r_s-1}\theta_{r_si_s+i}\widehat{X}^{r_si_s+i}-\widetilde{X}_s^{i_s}, K_{s,v}^*(y_s-y_s^{i_s})\rangle\\
&=\langle\sum_{i=0}^{r_s-1}(\theta_{N-r_s+i}(\widehat{X}^{N-r_s+i}-X^{N-2r_s+i})), K_s^*(y_s-y_s^{N_s-1})\rangle\\
&\quad+\sum_{i_s=1}^{N_s-1}\langle\sum_{i=0}^{r_s-1}(\theta_{r_s(i_s-1)+i}(\widehat{X}^{r_s(i_s-1)+i}-X^{r_s(i_s-2)+i})), K_s^*(y_s^{i_s}-y_s^{i_s-1})\rangle\\
&\le\sum_{i=0}^{r_s-1}\theta_{r_s(N_s-1)+i}\|\widehat{X}^{r_s(N_s-1)+i}-X^{r_s(N_s-2)+i}\|\cdot\|K_s^*(y_s-y_s^{N_s-1})\|_*\\
&\quad+\sum_{i_s=1}^{N_s-1}\sum_{i=0}^{r_s-1}\theta_{r_s(i_s-1)+i}\|\widehat{X}^{r_s(i_s-1)+i}-X^{r_s(i_s-2)+i}\|\cdot\|K_s^*(y_s^{i_s}-y_s^{i_s-1})\|_*
\end{aligned}$$

Bounding the gap. Putting the above together, and for convenience, denoting $y_s^{N_s} = y_s$, we have

$$\begin{aligned}
&\sum_{k=0}^{N} \theta_k G(\widehat{X}^k, \overline{Y}^k; Z) \\
&\leq \frac{5(\overline{r^2}/\overline{r})^2)}{2C} D(X, X^{init}) - \sum_{k=0}^{N} \theta_k \sum_{s=1}^{S} \eta_{k,s} D(\widehat{X}^k, X^{k-r_s}) + \sum_{k=0}^{N} \theta_k \delta_k(X) \\
&\quad + \sum_{s=1}^{S} \tau_s \left\{ D(y_s, y_s^{init}) - D(y_s, y_s^{N_s-1}) - \sum_{i_s=1}^{N_s-1} D(y_s^{i_s}, y_s^{i_s-1}) \right\} \\
&\quad + \sum_{s=1}^{S} \left\{ \sum_{k=0}^{N} \frac{\theta_k \eta_{k,s}}{2} \|\widehat{X}^k - X^{k-r_s}\|^2 + \sum_{i_s=1}^{N_s} \frac{\widetilde{\kappa}_s^2}{2\rho_s} (\sum_{i=0}^{r_s-1} \frac{\theta_{r_s(i_s-1)+i}}{\eta_{r_s(i_s-1)+i}}) \|y_s^{i_s} - y_s^{i_s-1}\|^2 \right\} \\
&\leq \frac{5(\overline{r^2}/\overline{r})^2)}{2C} D(X, X^{init}) + \sum_{k=0}^{N} \theta_k \delta_k(X) + \sum_{s=1}^{S} \tau_s D(y_s, y_s^{init})
\end{aligned}$$

using

$$\tau_s = \frac{\widetilde{\kappa}_s^2}{\rho_s} \cdot \frac{4 r_s \overline{r} C}{\mu} \geq \frac{\widetilde{\kappa}_s^2}{\rho_s} \cdot \max_{i_s \in [N_s]} (\sum_{i=0}^{r_s-1} \frac{\theta_{r_s(i_s-1)+i}}{\eta_{r_s(i_s-1)+i}}).$$

The result then follows from $\sum_{k=0}^{N} \theta_k \geq \frac{N(N+1)}{2}$ and convexity of F and R. □

Corollary 3.4. *Under the conditions in Theorem 3.2, and assume that $(X^k, \widehat{X}^k)$ are constructed using the generalized communication sliding (8) with $T_k/N = T/N \geq \max(\frac{5}{\sqrt{D_1}}, \frac{64\overline{r}}{D_1})$ where $D_1 = \frac{\mu^2(\overline{r^2}/\overline{r})^2}{2M^2C^2} D_0$, $\lambda_t = t$ and $\beta_t^k = \frac{(t+1)\mu}{2\eta_k C} + \frac{t-1}{2}$ for $t = 1, \ldots, T_k$. Then the following holds for all $Z \in \mathcal{Z}$,*

$$\begin{aligned}
G(Z^N; Z) \leq \frac{2}{N(N+1)} \Bigg\{ &\frac{\mu(\overline{r^3}/\overline{r} + 5(\overline{r^2}/\overline{r})^2)}{2C} D(X, X^{init}) \\
&+ \frac{\mu(\overline{r^2}/\overline{r})^2}{C} D_0 + \frac{4\overline{r}C}{\mu} \sum_{s=1}^{S} \frac{\widetilde{\kappa}_s^2 r_s}{\rho_s} D(y_s, y_s^{init}) \Bigg\}.
\end{aligned}$$

Proof of Corollary 3.4. Recall that from Corollary 3.1, the following holds:

$$\delta_k(X) = \frac{2M^2/\eta_k}{T_k(T_k+1)} \sum_{t=1}^{T_k} \frac{\lambda_t}{\beta_t}.$$

Thus, the result follows from Theorem 3.2 and the following bound, which we show next:

$$\sum_{k=0}^{N} \theta_k \delta_k(X) \leq \frac{\mu(\overline{r^2}/\overline{r})^2}{C} D_0.$$

The bound on the rest of the terms is since

$$\frac{1}{\eta_k} \sum_{t=1}^{T_k} \frac{\lambda_t}{\beta_t^k} = \sum_{t=1}^{T_k} \frac{2tC/\mu}{(t+1) + (t-1)(k + \overline{r^2}/\overline{r})/2\overline{r}} \leq \frac{C}{\mu} (1 + \frac{4(T_k - 1)}{1 + (k + \overline{r^2}/\overline{r})/2\overline{r}}),$$

and notice that

$$\sum_{k=0}^{N} \frac{k + 2\overline{r^2}/\overline{r}}{k + \overline{r^2}/\overline{r} + 2\overline{r}} \leq 2(N+1) \leq 4N,$$

and since $N+1 \geq r_{\max} := \max_{s\in[S]} r_s$, we have

$$\overline{r^2} = \sum_{s=1}^{S} \rho_s r_s^2 \leq \sum_{s=1}^{S} \rho_s r_s (N+1) = (N+1)\overline{r} \implies \overline{r^2}/\overline{r} \leq N+1 \leq 2N.$$

Thus, for $T_k/N = T/N \geq \max(\frac{5}{\sqrt{D_1}}, \frac{64\overline{r}}{D_1})$ where $D_1 = \frac{\mu^2(\overline{r^2}/\overline{r})^2}{2M^2C^2}D_0$, we have

$$\begin{aligned}\sum_{k=0}^{N} \frac{2M^2\theta_k}{\eta_k T_k(T_k+1)} \sum_{t=1}^{T_k} \frac{\lambda_t}{\beta_t^k} &\leq \frac{2M^2C}{\mu}\left\{\sum_{k=0}^{N}\frac{k+2\overline{r^2}/\overline{r}}{T_k(T_k+1)} + \sum_{k=0}^{N}\frac{4(k+2\overline{r^2}/\overline{r})}{(1+(k+\overline{r^2}/\overline{r})/2\overline{r})(T_k+1)}\right\} \\ &\leq \frac{2M^2C}{\mu}\left\{\frac{10N^2}{T^2} + \frac{32N\overline{r}}{T}\right\} \leq \frac{\mu}{C}D_0(\overline{r^2}/\overline{r})^2.\end{aligned}$$

□

As a direct consequence, we have the following theorem.

Corollary 3.5. *For $\widehat{X} \in \overline{\mathcal{X}}$, assume that the following are finite:*

$$D(\widehat{X}, X^{init}) \leq D^X < \infty, \qquad \sup_{y_s \in \mathsf{dom}(R_s^*)} D(y_s, y_s^{init}) \leq D_s^y < \infty.$$

Under the conditions in Corollary 3.4, taking $\rho_s = \frac{\widetilde{\kappa}_s\sqrt{D_s^y}}{\sum_{s'=1}^{S}\widetilde{\kappa}_{s'}\sqrt{D_{s'}^y}}$ and $D_0 = D^X$,

$$\sup_{Y'\in\mathbb{R}^n} G(Z^N; \widehat{X}, Y') \leq \frac{2}{N(N+1)}\left\{\frac{\mu(\overline{r^3}/\overline{r} + 7(\overline{r^2}/\overline{r})^2)}{2C}D^X + \frac{4C(\overline{r})^2}{\mu}(\sum_{s'=1}^{S}\widetilde{\kappa}_{s'}\sqrt{D_{s'}^y})^2\right\}.$$

Notice that (1) implies that $\|x - x'\| \leq \frac{M}{\mu}$ for all $x, x' \in \mathcal{X}$. Thus, one can take $D^X = O(\frac{CM^2}{\mu^2})$. The resulting upper bound, when $\overline{r^3} = O((\overline{r})^3)$ and $\overline{r^2} = O((\overline{r})^2)$, becomes

$$\sup_{Y'\in\mathbb{R}^n} Q(Z^N; \widehat{X}, Y') = O(\frac{\overline{r}^2}{\mu N^2}\left\{M^2 + C(\sum_{s'=1}^{S}\widetilde{\kappa}_{s'}\sqrt{D_{s'}^y})^2\right\}).$$

4 Application to distributed optimization

For the problem $(\mathcal{P}_d)$, we assume that $f_v : \overline{\mathcal{X}} \to \mathbb{R}$ is a convex and possibly non-smooth objective function such that for some $M_f, \mu_f \geq 0$, we have for all $v \in V$,

$$\frac{\mu_f}{2}\|x - x'\|^2 \leq f_v(x) - f_v(x') - \langle f'(x'), x - x'\rangle \leq M_f\|x - x'\|, \quad \forall x, x' \in \overline{\mathcal{X}}, \tag{17}$$

where $f_v' : \mathcal{X} \to \mathbb{R}^d$ is a subgradient oracle, i.e. $f_v'(x) \in \partial f_v(x)$ for all $x \in \mathcal{X}$, and f_v' is only available to agent v. For instance, when $\|f_v'\|_* \leq M_f$, $M = 2M_f$ holds.

To apply the proposed saddle point algorithms to the distributed optimization problem $(\mathcal{P}_d)$, we adopt the popular approach of a lifted space reformulation[28, 45], where the decision variables become $(x_v)_{v\in V}$, and x_v is agent v's local version of the decision variable x. The agents collaborate to reach *consensus* on an approximate *minimizer* of $(\mathcal{P}_d)$. Below, we provide the detailed lifted space reformulation (Section 4.1), its connection to communication protocols (Section 4.2), and heterogeneity between the local objectives f_v (Section 4.3). Then, in Section 4.4, we describe how our proposed multi-timescale PDHG can be applied, and provide the convergence guarantee performance.

4.1 Lifted space formulation

To set the stage, we denote $\mathcal{X} = \prod_{v\in V} \overline{\mathcal{X}} \subset \mathbb{R}^d$, where $d = m\overline{d}$ is the dimension of the lifted space. it's easy to see that since $\overline{\mathcal{X}}$ is convex, so is the resulting $\mathcal{X}$. In the lifted space, to ensure that agents (approximately) reach consensus, that is $x_v \approx x_{v'}$ for all $v, v' \in V$, we need to impose the *consensus constraint*, which results in the following *penalized*problem:

$$\min_{X=(x_v)_{v\in V}\in\mathcal{X}} \max_{Y=(y_1,\dots,y_S)\in\mathbb{R}^n} F(X) + \sum_{s=1}^{S} \left(\langle K_s X, y_s\rangle - R_s^*(y_s)\right), \quad F(X) := \sum_{v\in V} f_v(x_v). \tag{$\mathcal{P}_d^{lift}$}$$

Notice that the formulation ($\mathcal{P}_d^{lift}$) is exactly the same as the formulation in ($\mathcal{P}_b$), thereby making (multi-timescale) PDHG applicable. Moreover, by (17), (1) holds with $M = \sqrt{m}M_f$ and $\mu = \mu_f$. Below, we provide the details of K_s and R_s in ($\mathcal{P}_d^{lift}$).

4.1.1 Consensus constraints

In ($\mathcal{P}_d^{lift}$), $K_s \in \mathbb{R}^{n_s\times d}$ is a matrix such that $\cap_{s\in S}\ker(K_s)$ is the subspace in $\mathbb{R}^d$ where all $\{x_v\}_{v\in V}$ are the same. For convenience, we denote $K_s X = \sum_{v\in V} K_{s,v}x_v$ for $K_{s,v} : \mathbb{R}^{\overline{d}} \to \mathbb{R}^{n_s}$, and abbreviate $K : \mathbb{R}^d \to \mathbb{R}^n$ where $(KX)_s = K_s X$. We make the following assumption.

Assumption 4.1. *$KX = \mathbf{0}$ if and only if $x_v = x_{v'}$ for all $v, v' \in V$.*

As an example, denoting $\Pi : \mathbb{R}^d \to \mathbb{R}^d$ as the projection such that for any $X \in \mathbb{R}^d$, $\Pi(X)_v = \frac{1}{m}\sum_{v'\in V} x_{v'}$, then we can take $K = I - \Pi$. Moreover, for any K satisfying Assumption 4.1, $K^*(KK^*)^\dagger K = I - \Pi$ holds.

We point out that the kernel condition in Assumption 4.1 is the only requirement we impose on K. However, to make sure the resulting primal-dual algorithm can be implemented in a distributed fashion, additional sparsity requirements are needed depending on how agents communicate with each other. We provide details, examples, and the rationale behind the block-decomposable formulation in Section 4.2.

4.1.2 Penalties

In ($\mathcal{P}_d^{lift}$), we assume that $R_s : \mathbb{R}^{n_s} \to \overline{\mathbb{R}}$ is proper, convex, and lower-semicontinous, and $R_s(\mathbf{0}) = 0$. Under these conditions, ($\mathcal{P}_b$) (and thus ($\mathcal{P}_d^{lift}$)) admits a primal-only formulation ($\mathcal{P}_p$). In this formulation, $R_s(K_s X)$ becomes a penalty term, penalizing the deviation of $K_s X$ from $\mathbf{0}$. We further define $R : \mathbb{R}^n \to \overline{\mathbb{R}}$ as $R(y_1, \dots, y_S) = \sum_{s=1}^{S} R_s(y_s)$.

As an example, if for all $s \in [S]$, R_s is the characteristic function of the set $\{\mathbf{0}\}$, i.e. $R_s(\mathbf{0}) = 0$ and $R_s(y_s) = \infty$ for $y_s \neq \mathbf{0}$, then ($\mathcal{P}_d^{lift}$) is equivalent to ($\mathcal{P}_d$), and $R_s^*(y_s) = 0$ for all $y_s \in \mathbb{R}^{n_s}$. As another example, R_s can be any scaled norm, for instance $R_s(y_s) = \lambda\|y_s\|_p$ for some $p \geq 1$ and $\lambda > 0$, then $R_s^*(y_s) = 0$ for $\|y_s\|_q \leq \lambda$ and $R_s^*(y_s) = \infty$ otherwise, where $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$ (i.e. $p^{-1} + q^{-1} = 1$). That is R_s^* is the characteristic function of the dual-norm-ball of size λ.

Notice that with penalties different from the characteristic functions of $\{\mathbf{0}\}$, the problem ($\mathcal{P}_d^{lift}$) becomes a "relaxation" of ($\mathcal{P}_d$), thereby are not equivalent. As one can imagine, the two formulations get closer as the consensus constraints are penalized more. In fact, there are two tensions in choosing good penalties. Take $S = 1$ and $R(Y) = \lambda\|Y\|$ as an example.

- On one hand, with larger penalty on the consensus constraint violation (i.e. larger λ), ($\mathcal{P}_d^{lift}$) will become a better proxy for ($\mathcal{P}_d$); this encourages larger λ.
- On the other hand, the diameter of $\mathsf{dom}(R^*)$ is $O(\lambda)$, and as suggested by Corollaries 3.3 and 3.4, the complexities of our algorithms are $O(\lambda)$; this encourages smaller λ.

In Section 4.3, we propose a set of conditions on the penalties to achieve a balance between these two tensions, such that "good solutions" to ($\mathcal{P}_d^{lift}$) and ($\mathcal{P}_b$) (as measured using the duality gap (2)) are also "good solutions" to ($\mathcal{P}_d$) (as measured by objective value suboptimality and constraint violation in (18)), and the resulting algorithms have favorable dependence on the problem parameters.

4.1.3 Performance measure

To measure the performance of $X \in \mathcal{X}$, following [28], we consider the (ϵ, δ)-solution, satisfying the following conditions

$$F(X) \le F(X^*) + \epsilon, \quad \|(I - \Pi)X\| \le \delta. \tag{18}$$

That is, X is ϵ-suboptimal in terms of the objective value, while violating the consensus constraints by at most δ.[5] Nevertheless, our algorithms have performance guarantees on the duality gap of the saddle point fromulation. To transfer such duality gap guarantee back to (ϵ, δ)-solution guarantee, in Section 4.3, we propose additional requirements for the regularization R.

4.2 Agents, communication, and additional requirements on K

By distributed optimization, we mean that the objective functions $\{f_v\}_{v\in V}$ are distributed among m *primal agents*: for each $v \in V$, $\mathsf{Agent}(x_v)$ has access to f_v', the first order oracle for f_v, and is responsible for updating the variable x_v. In addition, we assume that there are S *dual agents*: for each $s \in [S]$, $\mathsf{Agent}(y_s)$ is responsible for updating the variable y_s.

We assume that for any pair $(s, v) \in [S]\times V$ such that $K_{s,v} \neq \mathbf{0}$, $\mathsf{Agent}(x_v)$ and $\mathsf{Agent}(y_s)$ can communicate (in both directions). For instance, all agents might be nodes in a connected graph with vertices $[S] \cup V$ (representing S dual agents and m primal agents), and communication can be realized through edges (directly) or through paths (i.e. with the help of intermediate agents). In particular, since the graph is connected, any pair can communicate, but the resources consumed and/or time taken by communication between different pairs could be (significantly) different.

At this point, we abstract away from how such communication is realized, and leave the discussion of the costs of communication to Section 4.4. Below, we provide two such realizations: *decentralized* and *hierarchical*, and provide examples in Figure 2.

Decentralized setting. In this setup, we assume that the dual variables are kept and updated by primal agents, respecting a graph based communication constraints. Precisely, let $\mathcal{G} = (V, E)$ denote an undirected, connected graph, and for each $s \in [S]$, we assign all tasks of $\mathsf{Agent}(y_s)$ to $\mathsf{Agent}(x_{v_s})$ for some $v_s \in V$, such that $\{v_s, v'\} \in E$ for each $K_{s,v'} \neq \mathbf{0}$.

As an example, let $W \in \mathbb{R}^{V\times V}$ be a doubly stochastic matrix such that $W_{v,v'} \neq 0$ only if $\{v, v'\} \in E$ or $v = v'$, and $\ker(I - W) = \mathsf{Span}(\mathbf{1})$ (and so $K := (I - W) \otimes I_{\overline{d}}$ satisfies Assumption 4.1). We can choose $S = m$, $n_s = d$, and decompose K as $K_s := (I - W)_s \otimes I_{\overline{d}}$,

$$K_s X = \sum_{v\in V} (I - W)_{s,v} x_v = x_s - \sum_{\{v,s\}\in E} W_{s,v} x_v, \quad s = 1, \dots, m.$$

Thus, $\mathsf{Agent}(y_s)$'s tasks can be assigned to $\mathsf{Agent}(x_s)$.

Hierarchical setting. In this setup, we assume that there is an underlying tree with nodes $[S]\cup V$, where all non-leaf nodes ($[S]$) correspond to dual agents and all leaf nodes (V) correspond to primal agents. Each non-leaf node can communicate with its child nodes directly. Precisely, for $s \in [S]$, we use $\mathsf{Chi}(s) \subset [S] \cup V$ to denote the child nodes of $\mathsf{Agent}(y_s)$, and $\mathsf{Des}(s) \subset V$ to denote all *primal agents* in the subtree rooted at $\mathsf{Agent}(y_s)$.

For convenience, for each $s \in [S]$, we denote the "mean" of all descendants of $\mathsf{Agent}(y_s)$ as $\overline{x}_s = |\mathsf{Des}(s)|^{-1} \sum_{j\in\mathsf{Des}(s)} x_j$. Then, consider $K_s : \mathbb{R}^d \to \mathbb{R}^{|\mathsf{Chi}(s)|\overline{d}}$ defined as

$$(K_s X)_i = \overline{x}_i - \overline{x}_s = \overline{x}_i - \sum_{j\in\mathsf{Chi}(s)} \frac{|\mathsf{Des}(j)|}{|\mathsf{Des}(s)|} \overline{x}_j, \quad i \in \mathsf{Chi}(s). \tag{19}$$

Then, it is easy to see that K satisfies Assumption 4.1, and since $\overline{x}_j$ can be computed in a bottom up manner, $\{K_s\}_{s\in[S]}$ can be realized through this tree. In addition, the set of $\{K_s\}_{s\in[S]}$ admits the following orthogonality properties which will be useful in choosing R. We defer the proof to Appendix B.

[5] In [28], $\|KX\| \le \delta$ is used instead of $\|(I - \Pi)X\|$, and K is assumed to be the Laplacian matrix for the underlying graph of communication. However, we use a generic K satisfying condition 4.1. In particular, for any K that is a valid choice, λK is also valid for any $\lambda \neq 0$. Thus, it makes sense to "normalize" K, and we use $I - \Pi = K^*(KK^*)^\dagger K$.

Lemma 4.1. *Let $\{K_s\}_{s\in[S]}$ be as defined in (19). Then for $s \neq s' \in [S]$, $K_s K_{s'}^* = \mathbf{0}$. In addition, denoting $\Pi_s := K_s^*(K_s K_s^*)^\dagger K_s$, we have for any $\widetilde{X}, \widehat{X} \in \mathbb{R}^d$*

$$\langle \widehat{X}, \Pi_s \widetilde{X} \rangle = \langle \Pi_s \widehat{X}, \Pi_s \widetilde{X} \rangle = \sum_{i \in \mathsf{Chi}(s)} |\mathsf{Des}(i)| \cdot \langle (K_s \widehat{X})_i, (K_s \widetilde{X})_i \rangle.$$

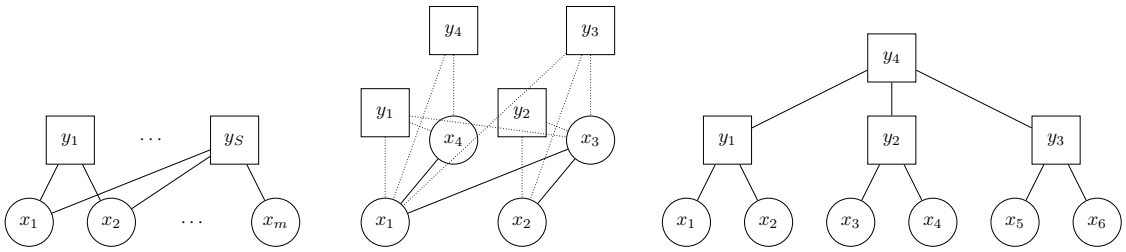


Figure 2: Left: abstract setting with m primal agents and S dual agents. Middle: realization in the decentralized setting, where $S = m = 4$, $\mathsf{Agent}(x_s) = \mathsf{Agent}(y_s)$, and the underlying graph is $(V, E = \{\{1,3\},\{1,4\},\{2,3\}\})$. Right: realization in the hierarchical setting.

4.3 Requirements for R and function similarity

Recall that when R is the characteristic function of $\{\mathbf{0}\}$, the penalized formulation $(\mathcal{P}_d^{lift})$ is equivalent to $(\mathcal{P}_d)$. In this section, we discuss the requirements for R such that the duality gap provides upper bounds on the suboptimality of the objective value and the violation of the consensus constraints.

First, we have the following upper bounds on the suboptimality of the objective value.

Lemma 4.2. *For any $\widehat{X} \in \mathcal{X}$ such that $K\widehat{X} = \mathbf{0}$,*

$$F(X) \leq F(\widehat{X}) + \sup_{Y' \in \mathsf{dom}(R^*)} G(X, Y; \widehat{X}, Y').$$

In particular, if $\sup_{Y' \in \mathsf{dom}(R^*)} G(X, Y; X^*, Y') \leq \epsilon$, then $F(X) \leq F(X^*) + \epsilon$, where $X^* = (x^*)_{v\in V}$ and x^* is an optimal solution to $(\mathcal{P}_d)$.

Proof of Lemma 4.2. Recall that we have

$$\sup_{Y \in \mathbb{R}^n} \langle KX, Y \rangle + F(X) - R^*(Y) = F(X) + R(KX).$$

In addition, since $R(\mathbf{0}) = 0$, we have $R^*(Y) = \sup_{Y' \in \mathbb{R}^n} \langle Y', Y \rangle - R(Y') \geq \langle \mathbf{0}, Y \rangle - R(\mathbf{0}) = 0$, and so

$$\langle K\widehat{X}, Y \rangle + F(\widehat{X}) - R^*(Y) \leq F(\widehat{X}), \quad \forall\ \widehat{X} \in \mathcal{X},\ K\widehat{X} = \mathbf{0}.$$

The second claim follows directly from the first since $KX^* = \mathbf{0}$. □

To connect the duality gap with the constraint violation $\|(I - \Pi)X\|$ in (18), or with the objective value suboptimality of $\frac{1}{m}\sum_{v\in V} x_v$, it turns out additional requirements are needed for R.

For convenience, we denote $\sigma_{\min}^+(K_s) = \min_{X \in \mathbb{R}^d,\ \Pi_s X \neq \mathbf{0}} \frac{\|K_s X\|_*}{\|\Pi_s X\|}$, where the numerator uses the dual norm to the norm in $\mathbb{R}^{n_s}$ and the denominator uses the norm in $\mathbb{R}^d$. As an example, when all norms are l_2 norms, $\sigma_{\min}^+(K_s)$ is the smallest non-zero singular value of K_s.

4.3.1 Requirements on R under orthogonaltiy

Below, we show that if K_s measures the constraint violation in *orthogonal* subspaces, then as long as R_s grows fast enough, the duality gap provides an upper bound on the constraint violation $\|(I - \Pi)X\|$ and the suboptimality of ΠX.

Lemma 4.3. *Further assume that for any $s \neq s' \in [S]$, $K_s K_{s'}^* = \mathbf{0}$, and for each $s \in [S]$, denoting*

$$\Pi_s = K_s^*(K_s K_s^*)^\dagger K_s, \quad a_s \geq \sup_{X' \in \mathcal{X},\ KX' = \mathbf{0}} \|\Pi_s \nabla F(X')\|_*,$$

where $\nabla F : \mathcal{X} \to \mathbb{R}^d$ is an arbitrary subgradient oracle, i.e. $(\nabla F(X))_v \in \partial f_v(x_v)$. If $\sup_{Y' \in \mathsf{dom}(R^*)} G(X, Y; X^*, Y') \leq \epsilon$,

1. *X is an $(\epsilon, \epsilon/\xi)$-solution if for each $s \in [S]$,*

$$R_s(y_s) \geq R_s^{ccv}(y_s) := \frac{\xi + a_s}{\sigma_{\min}^+(K_s)} \|y_s\|_*. \tag{20}$$

2. *the projected solution ΠX is an $(\epsilon(1 + 1/\xi), 0)$-solution if for each $s \in [S]$,*

$$a_s > 0, \quad R_s(y_s) \geq R_s^{prj}(y_s) := \frac{(1+\xi)a_s}{\sigma_{\min}^+(K_s)} \|y_s\|_*. \tag{21}$$

In Lemma 4.3 (and Corollary 4.1 below), the superscript *ccv* means $\{R_s^{ccv}\}_{s\in[S]}$ ($\{\widehat{R}_s^{ccv}\}_{s\in[S]}$) are designed to provide guarantees on the consensus constraint violation, and the superscript *prj* means $\{R_s^{prj}\}_{s\in[S]}$ ($\{\widehat{R}_s^{prj}\}_{s\in[S]}$) are designed to provide guarantees the projected solution ΠX.

Proof of Lemma 4.3. First, notice that by the orthogonality of $\{K_s\}_{s\in[S]}$, for any $Y \in \mathbb{R}^n$

$$(KK^*Y)_s = K_s(\sum_{s'=1}^S K_{s'}^* y_{s'}) = K_s K_s^* y_s, \ \forall s \in [S].$$

That is, KK^* is diagonal, and so

$$((KK^*)^\dagger Y)_s = (K_s K_s^*)^\dagger y_s, \ \forall s \in [S].$$

Thus, we can make the following decomposition

$$K^*(KK^*)^\dagger KX = \sum_{s=1}^S K_s^*(K_s K_s^*)^\dagger K_s X = \sum_{s=1}^S \Pi_s X.$$

For convenience, we denote $\widetilde{X} := \Pi X$, and by Lemma 4.3,

$$\sup_{Y' \in \mathsf{dom}(R^*)} G(X, Y; X^*, Y') \leq \epsilon \implies F(X) + R(KX) \leq F(X^*) + \epsilon \leq F(\widetilde{X}) + \epsilon. \tag{22}$$

In addition, using the convexity of F,

$$\begin{aligned} F(\widetilde{X}) - F(X) &\leq -\langle \nabla F(\widetilde{X}), (I - \Pi)X \rangle = -\sum_{s=1}^S \langle \nabla F(\widetilde{X}), \Pi_s X \rangle \\ &\leq \sum_{s=1}^S \|\Pi_s \nabla F(\widetilde{X})\|_* \cdot \|\Pi_s X\| \leq \sum_{s=1}^S a_s \cdot \|\Pi_s X\|. \end{aligned} \tag{23}$$

For the first claim, since $\|K_s X\|_* \geq \|\Pi_s X\| \sigma_{\min}^+(K_s)$, with the first condition (20) on R_s, we have

$$R_s(K_s X) \geq (\xi + a_s) \cdot \|\Pi_s X\|. \tag{24}$$

Combining the (22), (23), and (24), we get

$$\xi \cdot \sum_{s=1}^S \|\Pi_s X\| \leq \epsilon \implies \|(I - \Pi)X\| = \|\sum_{s=1}^S \Pi_s X\| \leq \sum_{s=1}^S \|\Pi_s X\| \leq \epsilon/\xi.$$

For the second claim, following a similar argument as above but with the second condition (21) on R_s, we get

$$\sum_{s=1}^{S} a_s \cdot \|\Pi_s X\| \le \epsilon/\xi. \tag{25}$$

Thus, using (22), (23), and (25), we have

$$F(\widetilde{X}) \le F(X) + \epsilon/\xi \le F(X^*) + \epsilon/\xi + \epsilon.$$

□

We would like to point out that in (23), $\langle \nabla F(\widetilde{X}), \Pi_s X\rangle$ is upper bounded using $\|\nabla F(\widetilde{X})\|_* \cdot \|\Pi_s X\|$. A tighter upper bound could be obtained if one has more information about the set $\mathcal{G}_s := \{\Pi_s \nabla F(X'),\ X' \in \mathcal{X},\ KX' = \mathbf{0}\}$. Indeed, $\langle \nabla F(\widetilde{X}), \Pi_s X\rangle \le \sup_{G_s \in \mathcal{G}_s} \langle G_s, \Pi_s X\rangle$, and so the inner product can be bounded using the support function of the set $\mathcal{G}_s$.

4.3.2 Function similarity for general convex functions

The terms $\{a_s\}_{s\in[S]}$ in Lemma 4.3 can be viewed as a "decomposition" of the function variation into different subspaces spanned by (the row spaces of) $\{K_s\}_{s\in[S]}$. To be more concrete, consider the hierarchical setting presented in Section 4.2, which satisfies exactly the conditions in Lemma 4.3 due to Lemma 4.1. Defining $\mu_s(i) = \frac{|\mathsf{Des}(i)|}{|\mathsf{Des}(s)|}$ for $i \in \mathsf{Chi}(s)$ as a probability measure, and assuming that all norms are l_2 norms, then by Lemma 4.1,

$$\|\Pi_s \nabla F\|_*^2 = |\mathsf{Des}(s)| \cdot \mathsf{Var}_{i\sim\mu_s}(\overline{f'}_i), \quad \overline{f'}_i = \frac{\sum_{j\in\mathsf{Des}(i)} f'_j}{|\mathsf{Des}(i)|},\ i \in \mathsf{Chi}(s), \tag{26}$$

where for a random vector V, we denote $\mathsf{Var}(V) := \mathbb{E}[\|V - \mathbb{E}[V']\|_*^2]$. Thus, $\|\Pi_s \nabla F\|_*$ measures the function variation among the *descendants of different child nodes* of $\mathsf{Agent}(y_s)$, i.e. among $\left\{\sum_{j\in\mathsf{Des}(i)} f'_j\right\}_{i\in\mathsf{Chi}(s)}$. As a result, the agents closer to the root of the tree, with more descendants, take care of function variation at *larger scales*, but at *lower resolution*, since for all $i \in \mathsf{Chi}(s)$, the variation inside $\left\{f'_j(x)\right\}_{j\in\mathsf{Des}(i)}$ has been taken care of by the dual agents in each sub-tree rooted at i.

For general but still orthogonal $\{K_s\}_{s\in[S]}$, a_s measures the function variation along the span of K_s. With this interpretation in mind, we make the following definition regarding function similarity.

Definition 4.1. *Assume that for all $s \ne s' \in [S]$, $K_s K_{s'}^* = \mathbf{0}$. We say that the set of functions $\{f_v\}_{v\in V}$ is $\{(a_s, K_s)\}_{s\in[S]}$-similar if there exists a subgradient oracle $\nabla F : \mathcal{X} \to \mathbb{R}^d$, i.e. $(\nabla F(X))_v \in \partial f_v(x_v))$, such that for each $s \in [S]$,*

$$\Pi_s = K_s^*(K_s K_s^*)^\dagger K_s, \quad a_s \ge \sup_{X'\in\overline{\mathcal{X}},\ KX'=\mathbf{0}} \|\Pi_s \nabla F(X')\|_*.$$

If $S = 1$ and $\Pi_1 = I - \Pi$, we abbreviate $\{(a_1, K_1)\}$-similar as a_1-similar.

For instance, if $S = 1$ and all norms are l_2 norms, then Assumption 4.1 requires that $\Pi_1 = I - \Pi$, and one can take a_1 as

$$a_1^2 \ge \sup_{x\in\overline{\mathcal{X}}} \sum_{v\in V} \|f'_v(x) - \frac{1}{m}\sum_{v'\in V} f'_{v'}(x)\|^2.$$

Thus, if $\|f'_v(x)\| \le M_f$ for all $v \in V, x \in \mathcal{X}$, we can also take $a_1 = 2\sqrt{m}M_f$.

Comparisons with existing notions of function similarity. [22] proposes the *bounded gradient dissimilarity* for differentiable convex objectives, which coincides with our Definition 4.1 when $S = 1$ and when the objectives are differentiable. For twice differentiable objectives, function similarity is also defined in terms of differences in Hessians, i.e. $\|\nabla^2 f_v - \nabla^2 f_{v'}\|$ [48, 24, 4, 22]. For general convex functions which could be non-differentiable, [4] informally defines it (δ-*relatedness* in their terminology) as the condition that "subgradients of local functions are at most δ-different from each other". Our Definition 4.1 formalizes this idea, and extend it to the case where $S > 1$.

4.3.3 Requirements on R_s without orthogonality

The above Lemma 4.3 imposes orthogonality assumptions on $\{K_s\}_{s\in[S]}$. In the more general case where such assumptions do not hold, one can always view $(\mathcal{P}_b)$ as a problem with only 1 block, with K and R as the corresponding operator and regularization. Applying Lemma 4.3, we get the following corollary.

Corollary 4.1. *Denoting* $\widehat{a}_1 \geq \sup_{X'\in\mathcal{X},\ KX'=\mathbf{0}} \|(I-\Pi)\nabla F(X')\|_*$ *where* $\nabla F : \mathcal{X} \to \mathbb{R}^d$ *is an arbitrary subgradient oracle, i.e.* $(\nabla F(X))_v \in \partial f_v(x_v)$. *If* $\sup_{Y'\in\mathsf{dom}(R^*)} G(X,Y;X^*,Y') \leq \epsilon$,

1. *X is an $(\epsilon, \epsilon/\xi)$-solution if for each $s \in [S]$,*

$$R_s(y_s) \geq \widehat{R}_s^{ccv}(y_s) := \frac{\xi + \widehat{a}_1}{\sigma^+_{\min}(K)}\|y_s\|_*, \tag{27}$$

2. *assume that $\widehat{a}_1 > 0$, then the projected solution ΠX is an $(\epsilon(1+1/\xi), 0)$-solution if for each $s \in [S]$,*

$$R_s(y_s) \geq \widehat{R}_s^{prj}(y_s) := \frac{(1+\xi)\widehat{a}_1}{\sigma^+_{\min}(K)}\|y_s\|_*. \tag{28}$$

Comparisons with [28] when $S = 1$. Assume that $\widehat{R}_1^{prj}$ in (28) is used for some constant $\xi > 0$ and $\widehat{a}_1 = 2\sqrt{m}M_f$, where M_f (defined below) is an upper bound on the norm of *the subgradient oracle* $f_v' \in \partial f_v$ (i.e. only one subgradient in the subdifferential for each $x \in \overline{\mathcal{X}}$, $v \in V$). Then, the diameter of $\mathsf{dom}(R^*)$ is $O(\frac{\sqrt{m}M_f}{\sigma^+_{\min}(K)})$. In [28], it is shown that for $(\mathcal{P}_b)$ with R being the characteristic function of $\{\mathbf{0}\}$, there exists an optimal dual solution $\|Y^*\| \leq \frac{\sqrt{m}\widehat{M}_f}{\sigma^+_{\min}(K)}$, where $\widehat{M}_f$ is an upper bound on the norms of *all subgradients* $g \in \partial f_v$:

$$\widehat{M}_f := \sup_{x\in\overline{\mathcal{X}},\ v\in V,\ g\in\partial f_v(x)} \|g\|_* \geq M_f := \sup_{x\in\overline{\mathcal{X}},\ v\in V} \|f_v'(x)\|_*.$$

Thus, even without function similarity, our $\widehat{R}_1^{prj}$ provides better control over the dual variables, leading to faster convergence.

4.4 Applying (accelerated) multi-timescale PDHG to distributed optimization

In this section, we present the distributed implementation of Algorithm 1 to the primal-dual formulation of the problem $(\mathcal{P}_d^{lift})$. We first point it out that if the distance generating function w_X in the lifted space is separable, i.e. $w_X(X) = \sum_{v\in V} w_{x_v}(x_v)$, then the Bregman divergence is also separable, i.e. $D(X, X') = \sum_{v\in V} D(x_v, x_v')$. Now revising the updates (3c) and (3a) as well as our proposed multi-timescale updates (5a), (6), and the approximation using the generalized communication sliding (8), we see that they are all decomposable w.r.t. the primal agents, and can be implemented locally by each $\mathsf{Agent}(x_v)$ without any communication. More precisely, defining

$$\phi_v^k(x_v) := f_v(x_v) + \langle \sum_{s=1}^S K_{s,v}^* \overline{y}_s^k, x_v\rangle + \sum_{s=1}^S \eta_{k,s} D(x_v, x_v^{k-r_s}), \tag{29}$$

then we require that

$$\phi_v^k(\widehat{x}_v^k) \leq \phi_v^k(x_v) - (\frac{\mu_f}{C} + \eta_k)D(x_v, x_v^k) + \delta_{k,v}(x_v), \quad \forall x_v \in \overline{\mathcal{X}}, \tag{30}$$

which can be achieved through the generalized communication sliding procedure applied to each x_v locally by $\mathsf{Agent}(x_v)$. Thus, the only communication needed for the primal dual updates is to make sure the following two conditions are met:

- the dual $\mathsf{Agent}(y_s)$ knows $K_s\widetilde{X}_s^{i_s}$ at iteration $k = r_s i_s$, which can be realized if at the beginning of iteration $k = r_s i_s$, each primal agent v computes $\widetilde{x}_v^{i_s}$ and send it to the dual agent y_s, then the dual agent computes $\sum_{v\in V} K_{s,v}\widetilde{x}_v^{i_s}$;

- the primal $\mathsf{Agent}(x_v)$ knows $\sum_{s=1}^S K_{s,v}^* \overline{y}_s^k$ at iteration k; this can be achieved if after each dual update, the dual agent y_s sends $y_s^{i_s} - y_s^{i_s-1}$ to all primal agents, then the primal agents can update $\sum_{s=1}^S K_{s,v}^* \overline{y}_s^k$ using $K_{s,v}^*(y_s^{i_s} - y_s^{i_s-1})$.

We provide such implementation in Algorithm 2, and explicitly mark the steps which require communication in green. Here, we provide few remarks. Notice that in Algorithm 1, $\mathsf{Agent}(y_s)$ calculates $\sum_{v\in V} K_{s,v}\widetilde{x}_{s,v}^{i_s}$ and sends $y_s^{i_s} - y_s^{i_s-1}$, a vector in $\mathbb{R}^{n_s}$, $\mathsf{Agent}(x_v)$ calculates $K_{s,v}^*(\overline{y}_s^k - \overline{y}_s^{k-1})$ and sends $\widetilde{x}_{s,v}^{i_s}$, a vector in $\mathbb{R}^{\overline{d}}$. In fact, there are many task assignment strategies: for instance, $K_{s,v}\widetilde{x}_{s,v}^{i_s}$ can also be computed by $\mathsf{Agent}(x_v)$, and the message from $\mathsf{Agent}(x_v)$ to $\mathsf{Agent}(y_s)$ will be $K_{s,v}\widetilde{x}_{s,v}^{i_s}$. This is preferable if $\mathsf{Agent}(x_v)$ can compute matrix-vector products faster/at lower cost than $\mathsf{Agent}(y_s)$. Due to this variability, in the cost analysis below, we take a "modular" perspective and assume that the cost of updating y_s (including all matrix-vector multiplication and communication) is c_s.

Algorithm 2 (Accelerated) Multi-timescale PDHG for distributed optimization

Input: $\{\alpha_{s,i_s}\}, \{\theta_k\}, \{\eta_{k,s}\}, \{\tau_{s,i_s}\}, \{r_s\}, X^{init}, Y^{init}$
Output: Primal dual pair Z^N
Initialize $(X^{k'}, \widehat{X}^{k'}, Y^{k'}) \leftarrow (X^{init}, X^{init}, Y^{init})$ for all $k' < 0$
for $k = 0, 1, \ldots, N$ **do** ▷ implicitly $i_s = \lfloor k/r_s \rfloor$ for all $s \in [S]$
  for $s \in [S]$ such that $k = 0 \pmod{r_s}$ **do**
    for $v \in V$ such that $K_{s,v} \neq \mathbf{0}$ **do**
      $\mathsf{Agent}(x_v)$ computes $\widetilde{X}_s^{i_s}$ using (5a) and sends it to $\mathsf{Agent}(y_s)$
    end for
    Dual update: $\mathsf{Agent}(y_s)$ computes $\sum_{v\in V} K_{s,v}\widetilde{x}_v^{i_s}$, then updates $y_s^{i_s}$ using (5b)
    $\mathsf{Agent}(y_s)$ sends $y_s^{i_s} - y_s^{i_s-1}$ (y_s^0 if $i_s = 0$) to $\mathsf{Agent}(x_v)$ for all $v \in V$ such that $K_{s,v} \neq \mathbf{0}$
  end for
  for $v \in V$ **do**
    Primal update: $\mathsf{Agent}(x_v)$ computes $K^*\overline{Y}^k$ where $\overline{y}_s^k = y_s^{\lfloor k/r_s \rfloor}$ for $s \in [S]$, updates $(x_v^k, \widehat{x}_v^k)$ satisfying (30)
  end for
end for
All $\mathsf{Agent}(x_v)$ and $\mathsf{Agent}(y_s)$ computes their components of Z^N using (9).

4.5 Convergence for general convex objectives

With additional assumptions specific to distributed optimization, and with proper choices of R_s's, the duality gap for $(\mathcal{P}_d^{lift})$ can be related to the suboptimality in terms of objective values F and/or violation of the consensus constraint for the original problem $(\mathcal{P}_d)$. Next, we establish such connection.

Corollary 4.2. *Assume that all norms are the l_2 norm, and take $y_s^0 = \mathbf{0}$, $w_{y_s}(y_s) = \frac{1}{2}\|y_s\|^2$, $w_x(x) = \frac{1}{2}\|x\|^2$, and $w_X(X) = \sum_{v\in V} w_x(x_v)$. Assume that conditions of Corollary 3.3 hold, $D(X^*, X^{init}) \leq D^X$, and the following holds for A specified below*

$$N \geq \frac{2\sqrt{3}\overline{r}A\sqrt{D^X}}{\epsilon}.$$

1. *If $\{f_v\}_{v\in V}$ is $\widehat{a}_1$-similar, take $\rho_s = \frac{\|K_s\|}{\sum_{s'=1}^S \|K_{s'}\|}$*

   (a) $\frac{1}{N+1}\sum_{k=0}^N \widehat{X}^k$ *is an $(\epsilon, \epsilon/\xi)$-solution if $R_s = \widehat{R}_s^{ccv}$ as defined in (27) (then $\sqrt{2D_s^y} = \frac{\xi+\widehat{a}_1}{\sigma_{\min}^+(K)}$) and $A = \frac{(\sum_{s=1}^S \|K_s\|)}{\sigma_{\min}^+(K)} \cdot (\xi + \widehat{a}_1)$;*

   (b) $\Pi(\frac{1}{N+1}\sum_{k=0}^N \widehat{X}^k)$ *is an $(\epsilon(1+1/\xi), 0)$-solution if $R_s = \widehat{R}_s^{prj}$ as defined in (28) (then $\sqrt{2D_s^y} = \frac{(1+\xi)\widehat{a}_1}{\sigma_{\min}^+(K)}$) and $A = \frac{(1+\xi)(\sum_{s=1}^S \|K_s\|)\cdot\widehat{a}_1}{\sigma_{\min}^+(K)}$.*

2. If $\{f_v\}_{v\in V}$ is $\{(a_s, K_s)\}_{s\in[S]}$-similar,

(a) $\frac{1}{N+1}\sum_{k=0}^{N}\widehat{X}^k$ is an $(\epsilon, \epsilon/\xi)$-solution if $R_s = R_s^{ccv}$ as defined in (20) (then $\sqrt{2D_s^y} = \frac{\xi+a_s}{\sigma_{\min}^+(K_s)}$), $\rho_s = (\frac{\xi+a_s}{\sigma_{\min}^+(K_s)})/(\sum_{s'=1}^{S}\frac{\xi+a_{s'}}{\sigma_{\min}^+(K_{s'})})$, and $A = \sum_{s=1}^{S}(\xi+a_s)\cdot\frac{\|K_s\|}{\sigma_{\min}^+(K_s)}$;

(b) $\Pi(\frac{1}{N+1}\sum_{k=0}^{N}\widehat{X}^k)$ is an $(\epsilon(1+1/\xi), 0)$-solution if $R_s = R_s^{prj}$ satisfies (21) (then $\sqrt{2D_s^y} = \frac{(1+\xi)a_s}{\sigma_{\min}^+(K_s)}$), $\rho_s = (\frac{a_s}{\sigma_{\min}^+(K_s)})/(\sum_{s'=1}^{S}\frac{a_{s'}}{\sigma_{\min}^+(K_{s'})})$, and $A = (1+\xi)(\sum_{s=1}^{S}a_s\cdot\frac{\|K_s\|}{\sigma_{\min}^+(K_s)})$.

Thus, the communication round N depends on $\overline{r}$, the weighted average of the rates at which the duals are updated, as well as A, which measures the function similarities.

Bounds using the Lipschitz constants. Consider the case where $\|f_v'\| \le M_f$, and to guarantee that $\frac{1}{N+1}\sum_{k=0}^{N}\widehat{X}^k$ is an $(\epsilon, \epsilon/\xi)$-solution, in Corollary 4.2, with $\{f_v\}_{v\in V}$ $\widehat{a}_1$-similar, we can take $\xi = \widehat{a}_1 = 2\sqrt{m}M_f$ which gives the following N_1, and with $\{f_v\}_{v\in V}$ $\{(a_s, K_s)\}_{s\in[S]}$-similar, we can take $\xi = a_s = 2\sqrt{m}M_f$ for all $s\in[S]$, which gives the following N_2:

$$N_1 = O(\frac{\overline{r}M_f\sqrt{mD^X}}{\epsilon}\cdot\frac{\sum_{s=1}^{S}\|K_s\|}{\sigma_{\min}^+(K)}),\quad N_2 = O(\frac{\overline{r}M_f\sqrt{mD^X}}{\epsilon}\cdot(\sum_{s=1}^{S}\frac{\|K_s\|}{\sigma_{\min}^+(K_s)})).$$

Both N_1 and N_2 depend linearly in $\overline{r}$. However, when $\{K_s\}_{s\in[S]}$ are orthogonal, as discussed in Section 4.3, $\sigma_{\min}^+(K) \le \sigma_{\min}^+(K_s)$ for all s, and so in terms of the rounds of communication N, it appears that orthogonality allows a more refined (i.e. s-dependent) control over the decomposition of the function variation and thus the dual domain size, thereby achieving better convergence. In addition, similar to the argument in Section 3.2, when the cost of updating y_s is c_s and total cost is additive, one should choose $r_s \propto \sqrt{c_s/\|K_s\|}$ under $\widehat{a}_1$-similarity, and $r_s \propto \sqrt{c_s/(\|K_s\|/\sigma_{\min}^+(K_s))}$ under $\{(a_s, K_s)\}_{s\in[S]}$-similarity. Similar results hold for $\Pi(\frac{1}{N+1}\sum_{k=0}^{N}\widehat{X}^k)$ to be an $(\epsilon, 0)$-solution.

Bounds using the function similarity. In reality, sometimes the functions $\{f_v\}_{v\in V}$ exhibit similarity. For instance, in the extreme case $f_v = f_{v'}$ for all $v, v'\in V$, and thus communication is not needed at all! In that case, the bound on $\sup_{\widetilde{X}\in\mathcal{X},\ K\widetilde{X}=\mathbf{0}}\|(I-\Pi)\nabla F(\widetilde{X})\|$ (and other terms using Π_s) using the Lipschitz constant M_f is too loose: in fact, one can choose $\widehat{a}_1 = a_s = \epsilon_0$ for all s for arbitrarily small $\epsilon_0 > 0$, then when R_s are set according to (28) or (21) with constant ξ, one only needs $N = O(\frac{\epsilon_0}{\epsilon}\sum_{s=1}^{S}r_s)$, which can be arbitrarily small.

More generally, choosing $\xi = 1$ and setting R_s according to (28) or (21), we obtain the following bound on the rounds of communication following under $\widehat{a}_1$-similarity (N_3) and $\{(a_s, K_s)\}_{s\in[S]}$-similarity (N_4):

$$N_3 = O(\frac{\overline{r}\sqrt{D^X}}{\epsilon}\cdot\frac{\sum_{s=1}^{S}\|K_s\|}{\sigma_{\min}^+(K)}\cdot\widehat{a}_1),\quad N_4 = O(\frac{\overline{r}\cdot\sqrt{D^X}}{\epsilon}\cdot(\sum_{s=1}^{S}\frac{a_s\|K_s\|}{\sigma_{\min}^+(K_s)})).$$

Importantly, the number of rounds needed now depends on the *function similarity* instead of crude quantities such as Lipschitz constants.

In fact, when $S = 1$, such dependency is optimal. Indeed, [4] designs a pair of "chain like" functions $\{F_1, F_2\}$, such that for any $\gamma \ge 0$, $\{\gamma F_1, \gamma F_2\}$ is $\sqrt{1.5}\gamma$-similar. In addition, when $m/2$ agents are given γF_1 and the rest are give γF_2, finding an ϵ suboptimal x (in terms of the objective value) in the l_2 unit ball requires $\Omega(\frac{\gamma}{\epsilon/m})$ rounds of communication (see Theorem 2 and the discussions after it in [4]). For our algorithm, with $\widehat{a}_1 = O(\sqrt{m}\gamma)$, $D^x = 1/2$, and $K = I - \Pi$ (and so $\|K\| = \sigma_{\min}^+(K)$), we have $N_3 = O(\frac{r_1\gamma}{\epsilon/m})$. Thus, y_1 is updated only $N_3/r_1 = O(\frac{\gamma}{\epsilon/m})$ times, which is also the number of actual communication rounds needed. This achieves the theoretical lower bound, and so is optimal.

The hierarchical setting and function similarity at different scales. In addition, we provide results when function variations could be different along the span of K_s for different $s\in[S]$. As an example, consider the hierarchical setting discussed in Section 4.2, with the additional assumption that for each non-leaf layer of the tree, all dual variables in that layer have the same number of child nodes. Then it can be

shown that $\|K_s\| = \sigma^+_{\min}(K_s) = \sqrt{|\mathsf{Chi}(s)|/|\mathsf{Des}(s)|}$ (by (32) in the proof of Lemma 4.1), so the above bound N_4 can be simplified as

$$N_4' = O(\frac{\overline{r} \cdot (\sum_{s=1}^S a_s) \cdot \sqrt{D^X}}{\epsilon}), \quad \rho_s \propto a_s \sqrt{|\mathsf{Des}(s)|}.$$

As discussed in Section 4.3.2, a_s measures the function variation along the span of K_s, i.e. variation in $\{f_v\}_{v \in \mathsf{Des}(s)}$ *not taken care of* by $\mathsf{Agent}(y_{s'})$ in the subtree rooted at $\mathsf{Agent}(y_s)$. In addition, (26) shows that $a_s^2 = |\mathsf{Des}(s)| \cdot \sup_{x \in \mathcal{X}} \mathsf{Var}_{i \sim \mu_s}(\overline{f'}_i(x))$, and so $\rho_s \propto |\mathsf{Des}(s)| \cdot \sqrt{\sup_{x \in \mathcal{X}} \mathsf{Var}_{i \sim \mu_s}(\overline{f'}_i(x))}$.

Thus, from the cost-minimization perspective in the discussion in Section 3.2, denoting the cost of updating y_s as c_s, one should choose $r_s \propto \sqrt{\frac{c_s/|\mathsf{Des}(s)|}{\sqrt{\sup_{x \in \mathcal{X}} \mathsf{Var}_{i \sim \mu_s}(\overline{f'}_i(x))}}}$. This corroborates the intuition that if along some K_s the function does not vary by too much ($\mathsf{Var}_{i \sim \mu_s}(\overline{f'}_i)$ is small), then $\mathsf{Agent}(y_s)$ does not need to update y_s very frequently (can use larger r_s).

4.6 Convergence for strongly convex objectives

Good initialization for $(\mathcal{P}_d^{lift})$. In Corollary 3.4, assuming that $\mathcal{X}$ is compact, then one can always use $D^X \geq \sup_{X \in \mathcal{X}} D_{w^X}(X, X^{init})$, suggesting that X^{init} should be chosen as the "center" of $\mathcal{X}$, and D^X measures the (squared) radius of $\mathcal{X}$. The resulting N, then, depends on D^X. However, such dependence on the size of $\mathcal{X}$ could be suboptimal, especially when local objectives are similar. Indeed, in the extreme case where all local functions are the same, then primal agents can optimize their local objectives without communication at all.

To take advantage of potential similarities in the local functions, we propose initializing the primal variables at (approximate) local optimal solutions, which has the following guarantee on $D(X^*, X^{init})$.

Lemma 4.4. *Assume that all norms are the l_2 norm, and for some $\epsilon_0 \geq 0$, $\widehat{X} = (\widehat{x}_v)_{v \in V} \in \mathcal{X}$ satisfies the following condition*

$$F(\widehat{X}) \leq \min_{X \in \mathcal{X}} F(X) + \epsilon_0.$$

Assume that (17) *holds for some $\mu > 0$ and $\{f_v\}_{v \in V}$ is $\{(a_s, K_s)\}_{s \in [S]}$-similar, then*

$$\|\widehat{X} - X^*\| \leq \frac{(\sum_{s=1}^S a_s^2)^{1/2}}{\mu} + \sqrt{\frac{\sum_{s=1}^S a_s^2}{\mu^2} + \frac{2\epsilon_0}{\mu}}.$$

Proof of Lemma 4.4. By the suboptimality condition for $\widehat{X}$ and (17), we get

$$\frac{\mu}{2}\|\widehat{X} - X^*\|^2 \leq F(\widehat{X}) - F(X^*) - \langle \nabla F(X^*), \widehat{X} - X^* \rangle \leq -\langle \nabla F(X^*), \widehat{X} - X^* \rangle + \epsilon_0.$$

Notice that by the first-order optimality condition of X^*, we get

$$\langle \nabla F(X^*), \Pi(\widehat{X} - X^*) \rangle \geq 0.$$

Combining the above two results, we get

$$\begin{aligned}
\frac{\mu}{2}\|\widehat{X} - X^*\|^2 &\leq -\langle \nabla F(X^*), (I - \Pi)(\widehat{X} - X^*) \rangle + \epsilon_0 \\
&= -\sum_{s=1}^S \langle \Pi_s \nabla F(X^*), \Pi_s(\widehat{X} - X^*) \rangle + \epsilon_0 \\
&\leq \sum_{s=1}^S \|\Pi_s \nabla F(X^*)\|_* \cdot \|\Pi_s(\widehat{X} - X^*)\| + \epsilon_0 \\
&\leq (\sum_{s=1}^S \|\Pi_s \nabla F(X^*)\|_*^2)^{1/2} \cdot (\sum_{s=1}^S \|\Pi_s(\widehat{X} - X^*)\|^2)^{1/2} + \epsilon_0 \\
&\leq (\sum_{s=1}^S a_s^2)^{1/2} \cdot \|\widehat{X} - X^*\| + \epsilon_0,
\end{aligned}$$

where the last $\leq$ is because of the assumption that $\{f_v\}_{v\in V}$ are $\{(a_s, K_s)\}_{s\in[S]}$-similar, and all norms are l_2 norm. The above inequality is quadratic in $\|\widehat{X} - X^*\|$, and the result follows. $\square$

The above Lemma 4.4 shows that if $\{\widehat{x}_v\}_{v\in V}$ are all approximately optimal to local objectives, then $D(X^*, \widehat{X}) \sim \frac{\sum_{s=1}^S a_s^2}{\mu^2}$. To find such initialization, one can apply the GS procedure.

Corollary 4.3. *Assume that all norms are l_2 norms and $w_x(x) = \frac{1}{2}\|x\|^2$, that (17) holds with some $\mu > 0$. For each $v \in V$, assume that* $\mathsf{Agent}(x_v)$ *is given some $\underline{x}_v^0 \in \mathcal{X}$ such that $\sup_{x\in\mathcal{X}} D(x, \underline{x}_v^0) \leq \underline{D}^x < \infty$*

$$(\_, x_v^{init}) = GS(f_v, \mathcal{X}, D, \underline{T}, \underline{\eta}, \mathbf{0}, \underline{x}_v^0, \underline{x}_v^0),$$

where the GS procedure uses λ_t, β_t according to Corollary A.1 (for $\mu > 0$), then with $\epsilon_0 = \widetilde{a}^2/\mu$, $\underline{T} \geq \frac{8CM_f^2 m}{\epsilon_0 \mu}$ and $\underline{\eta} = \frac{\epsilon_0/2}{m\underline{D}^x}$

$$D(X^*, X^{init}) \leq \frac{4\widetilde{a}^2}{\mu^2},$$

where $\widetilde{a} = \widehat{a}_1$ if $\{f_v\}_{v\in V}$ is $\widehat{a}_1$-similar for some $\widehat{a}_1 > 0$, and $\widetilde{a} = (\sum_{s=1}^S a_s^2)^{1/2}$ if $\{f_v\}_{v\in V}$ is $\{(a_s, K_s)\}_{s\in[S]}$-similar such that $(\sum_{s=1}^S a_s^2)^{1/2} > 0$.

Complexities for $(\mathcal{P}_d)$. Combining Theorem 3.5, Corollary 4.3, Lemma 4.3, and Corollary 4.1, we get the following results.

Corollary 4.4. *Assume that all norms are the l_2 norm, and take $y_s^0 = \mathbf{0}$, $w_{y_s}(y_s) = \frac{1}{2}\|y_s\|^2$, $w_x(x) = \frac{1}{2}\|x\|^2$, and $w_X(X) = \sum_{v\in V} w_x(x_v)$. Assume that conditions of Corollaries 3.5 and 4.3 hold and X^{init} is initialized according to Corollary 4.3, and the following holds for A_0, A_1 specified below*

$$N \geq \frac{2\overline{r}A}{\sqrt{\mu_f \epsilon}}, \quad A = \sqrt{\overline{r^3}/(\overline{r})^3 + 7(\overline{r^2}/(\overline{r})^2)^2} \cdot A_1 + A_0.$$

1. *If $\{f_v\}_{v\in V}$ is $\widehat{a}_1$-similar, then take $A_1 = \widehat{a}_1$ and $\rho_s = \frac{\|K_s\|}{\sum_{s'=1}^S \|K_{s'}\|}$*
   - *(a) $\frac{\sum_{k=0}^N \theta_k \widehat{X}^k}{\sum_{k=0}^N \theta_k}$ is an $(\epsilon, \epsilon/\xi)$-solution if $R_s = \widehat{R}_s^{ccv}$ as defined in (27) (then $\sqrt{2D_s^y} = \frac{\xi + \widehat{a}_1}{\sigma_{\min}^+(K)}$) and $A_0 = \frac{(\sum_{s=1}^S \|K_s\|)}{\sigma_{\min}^+(K)} \cdot (\xi + \widehat{a}_1)$;*
   - *(b) $\Pi(\frac{\sum_{k=0}^N \theta_k \widehat{X}^k}{\sum_{k=0}^N \theta_k})$ is an $(\epsilon(1+1/\xi), 0)$-solution if $R_s = \widehat{R}_s^{prj}$ as defined in (28) (then $\sqrt{2D_s^y} = \frac{(1+\xi)\widehat{a}_1}{\sigma_{\min}^+(K)}$) and $A_0 = \frac{(1+\xi)(\sum_{s=1}^S \|K_s\|)\cdot \widehat{a}_1}{\sigma_{\min}^+(K)}$.*
2. *If $\{f_v\}_{v\in V}$ is $\{(a_s, K_s)\}_{s\in[S]}$-similar where $a_s > 0$ for all s, then take $A_1 = (\sum_{s=1}^S a_s^2)^{1/2}$,*
   - *(a) $\frac{\sum_{k=0}^N \theta_k \widehat{X}^k}{\sum_{k=0}^N \theta_k}$ is an $(\epsilon, \epsilon/\xi)$-solution if $R_s = R_s^{ccv}$ as defined in (20) (then $\sqrt{2D_s^y} = \frac{\xi + a_s}{\sigma_{\min}^+(K_s)}$), $\rho_s = (\frac{\xi + a_s}{\sigma_{\min}^+(K_s)})/(\sum_{s'=1}^S \frac{\xi + a_{s'}}{\sigma_{\min}^+(K_{s'})^+})$, and $A_0 = \sum_{s=1}^S (\xi + a_s) \cdot \frac{\|K_s\|}{\sigma_{\min}^+(K_s)}$;*
   - *(b) $\Pi(\frac{\sum_{k=0}^N \theta_k \widehat{X}^k}{\sum_{k=0}^N \theta_k})$ is an $(\epsilon(1+1/\xi), 0)$-solution if $R_s = R_s^{prj}$ satisfies (21) (then $\sqrt{2D_s^y} = \frac{(1+\xi)a_s}{\sigma_{\min}^+(K_s)}$), $\rho_s = (\frac{a_s}{\sigma_{\min}^+(K_s)})/(\sum_{s'=1}^S \frac{a_{s'}}{\sigma_{\min}^+(K_{s'})^+})$, and $A_0 = (1+\xi)(\sum_{s=1}^S a_s \cdot \frac{\|K_s\|}{\sigma_{\min}^+(K_s)})$.*

Subgradient oracle complexities. With the initialization in Corollary 4.3 and $C = 1$, the number of subgradient steps needed to find X^{init} is $\underline{T} \geq \frac{8mM_f^2}{\widetilde{a}^2}$, constant in ϵ.

In addition, in Corollary 3.4, we can take $D_0 = \frac{4\widetilde{a}^2}{\mu_f^2}$, and so $D_1 = \frac{2\widetilde{a}^2(\overline{r^2}/\overline{r})^2}{M_f^2}$. Thus, the requirement on T becomes $T/N \geq \max(\frac{5}{\sqrt{D_1}}, \frac{64\overline{r}}{D_1})$, i.e. $T/N = \Omega(\max(\frac{M_f/\widetilde{a}}{\overline{r^2}/\overline{r}}, (\frac{M_f/\widetilde{a}}{\overline{r^2}/\overline{r}})^2 \cdot \overline{r}))$, and so the total subgradient steps needed (for each agent) is

$$N^2 \cdot O(\max(\frac{M_f/\widetilde{a}}{\overline{r^2}/\overline{r}}, (\frac{M_f/\widetilde{a}}{\overline{r^2}/\overline{r}})^2 \cdot \overline{r})) = O(\frac{\overline{r}^2 A^2}{\mu_f \epsilon} \cdot \max(\frac{M_f/\widetilde{a}}{\overline{r^2}/\overline{r}}, (\frac{M_f/\widetilde{a}}{\overline{r^2}/\overline{r}})^2 \cdot \overline{r}))$$

In the special case where $S = 1$ and $\|K\| = O(\sigma^+_{\min}(K))$, we have $\widetilde{a} = \Omega(A)$. Further assuming that $A = O(\sqrt{m}M_f)$ (which holds when $\|f'_v\| \le M_f$ for all v), the above can be simplified as $\frac{\overline{r}\sqrt{m}M_f^2}{\mu_f\epsilon}$.

Communication rounds complexities. From Corollary 4.4, the communication rounds needed is $N = O(\frac{\overline{r}A}{\sqrt{\mu_f\epsilon}})$, where A depends on function similarities and higher moments of $\{r_s\}_{s\in[S]}$. In terms of N and T's dependency on ϵ, μ_f, $O(1/\sqrt{\mu_f\epsilon})$ communication rounds and $O(1/(\mu_f\epsilon))$ gradient steps are needed.

Now, consider the special case where $\overline{r^2} = O((\overline{r})^2)$ and $\overline{r^3} = O((\overline{r})^3)$, i.e. r_s has small variation, then $\sqrt{\overline{r^3}/(\overline{r})^3 + 7(\overline{r^2}/(\overline{r})^2)^2} = O(1)$, and so $A = O(\widetilde{a} + A_0)$, then $\xi = 1$ with $\widehat{R}_s^{prj}$ and R_s^{prj} give the following N_1 and N_2 respectively

$$N_1 = O\left(\frac{\overline{r}\widehat{a}_1}{\sqrt{\epsilon\mu_f}} \cdot \frac{\sum_{s=1}^S \|K_s\|}{\sigma^+_{\min}(K)}\right), \quad N_2 = O\left(\frac{\overline{r}}{\sqrt{\epsilon\mu_f}} \cdot \left(\sum_{s=1}^S a_s \cdot \frac{\|K_s\|}{\sigma^+_{\min}(K_s)}\right)\right),$$

both have linear dependence on $\overline{r}$ and function similarities. (For N_2, we use $(\sum_{s=1}^S a_s^2)^{1/2} \le \sum_{s=1}^S a_s$ and $\|K_s\| \ge \sigma^+_{\min}(K_s)$.)

Comparison with communication lower bounds. When $S = 1$ (and so $\overline{r^2} = (\overline{r})^2$), $K = I - \Pi$ (and so $\|K\| = \sigma^+_{\min}(K)$), with $w_x(x) = \frac{1}{2}\|x\|^2$ and $w_{y_s}(y_s) = \frac{1}{2}\|y_s\|^2$, assuming that $\|f'_v\| \le M_f$, $a_1^2 \le m\gamma^2$ for some $\gamma \le M_f$, we have $N = O(\frac{\overline{r}\cdot\sqrt{m}\gamma}{\sqrt{\epsilon\mu_f}})$. Since communication is only needed when $\mathsf{Agent}(y_1)$ updates, the total number of communication rounds is $N/r_s = O(\frac{\sqrt{m}\gamma}{\sqrt{\epsilon\mu_f}})$, which achieves the theoretical lower bound (Theorem 2 and discussion after in [4]) on the communication round complexity for μ-strongly convex, $\sqrt{m}\gamma$-similar functions, and so is optimal[6].

5 Numerical experiments

Below, we present numerical experiments applying MT-PDHG to linear programming problems (Section 5.1), and (A)MT-PDHG to distributed Support Vector Machine problems (Section 5.2). All experiments are implemented using Python and run on MacBook Air with the M3 chip and 8 cores.

5.1 Experiment: linear programming

In this set of experiments, we apply our MT-PDHG and the vanilla PDHG to simulated linear programming problems of the form

$$\min_{X\in\mathbb{R}^n} c^T X, \quad s.t.\ AX = b,\ X \ge \mathbf{0},$$

where $A \in \mathbb{R}^{m\times n}$ and the rows are divided evenly into $S = 6$ blocks, with the associated dual blocks updated in parallel:

$$\min_{X\in\mathbb{R}^n} \max_{Y=(y_s)_{s\in[S]}\in\mathbb{R}^m} c^T X - \sum_{s=1}^S (y_s^T A_s X - y_s^T b), \quad s.t.\ X \ge \mathbf{0}.$$

Problem simulation. We simulate $c_i \sim \mathcal{N}(0,1)$ for $i \in [n]$, $A_{i,j} \sim \text{Uniform}([0,1))$ for $i \in [m], j \in [n]$, all independently. To ensure the problem is feasible, we simulate $X'_i \sim \text{Uniform}([0,1))$ for $i \in [n]$ independent of c, A, and take $b = AX'$.

Choice of rates r_s. We consider 4 combinations of the updating rates for the dual blocks: 1. $r_s = 1$ for all s; 2. $r_1 = r_2 = r_3 = 1$ and $r_4 = r_5 = r_6 = 10$; 3. $r_s = 10$ for all s; 4. $r_s = 50$ for all s. These are denoted in different colors in Figures 3 and 4.

Algorithm setups. We use $w_X(X) = \frac{1}{2}\|X\|^2$ and $w_{y_s}(y_s) = \frac{1}{2}\|y_s\|^2$. We benchmark our MT-PDHG against the vanilla PDHG, where at each global iteration k, X is updated as the minimizer to $\langle c - A^T\overline{Y}^k, X\rangle + \frac{\eta}{2}\|X - X^{k-1}\|^2$, with $\eta = \|A\|$, and at $k = i_s r_s$, the dual block y_s is updated as the minimizer to $\langle K_s\widetilde{X}_s^{i_s} -$

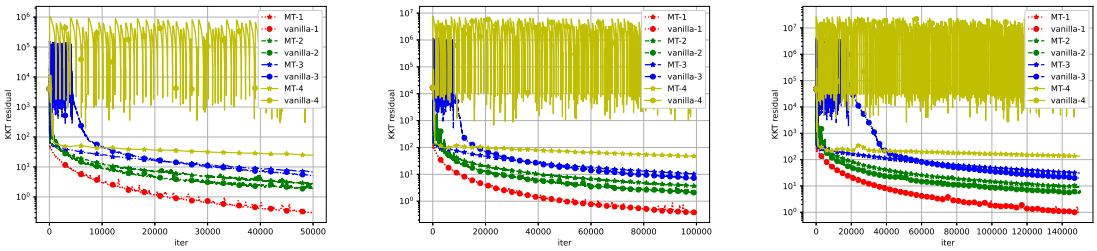
[6] [4] constructs a pair of "chain like" functions $\{\gamma F_1, \gamma F_2\}$ which are $\Theta(\gamma)$-similar and μ-strongly convex. In addition, when $m/2$ agents are given γF_1 and the rest are given γF_2 (and so this set of m functions is $\Theta(\sqrt{m}\gamma)$-similar), the number of rounds of communication needed is $\Omega(\gamma\sqrt{\frac{1}{\mu_f\epsilon/m}})$.

$b, y_s\rangle + \frac{\tau_s}{2}\|y_s - y_s^{i_s-1}\|^2$, where $\widetilde{X}_s^{i_s} = 2X^{k-1} - X^{k-2}$ and $\tau_s = 2S\|A_s\|^2/\eta$. Thus, there is no mixture of Bregman divergence and multi-timescale extrapolation. For our MT-PDHG, we use $\rho_s = 1/S$ and the same η, τ_s as above. We initialize $X = \mathbf{0}$ and $Y = \mathbf{0}$.

Results. We present the KKT residual $(\|AX - b\|^2 + \|[A^TY - c]_+\|^2 + [c^TX - b^TY]_+)^{1/2}$ for MT-PDHG and vanilla PDHG, under different combinations of r_s.

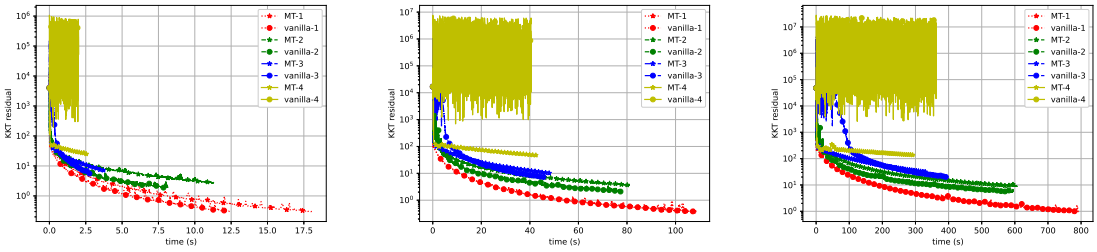
In Figure 3, we present the residual as a function of global iteration. As can be seen, our MT-PDHG is stable under various combinations of updating rates. Interestingly, the vanilla PDHG, even without mixture of Bregman divergence and multi-timescale extrapolation, still converges when the rates are small (rate combinations 1, 2, and 3). However, as the yellow lines suggest, when the rates ($r_s = 50$) are large, our explicit control through the Bregman divergence and the extrapolation helps stabilize the performance. Moreover, comparing the green and blue curves, which have the same $\max_s r_s = 10$ but different average r_s, which is 5.5 for green but 10 for blue, we see that smaller $\overline{r}$ indeed corresponds to faster convergence rate (as a function of iteration), demonstrating that our MT-PDHG are robust to extreme values in r_s.

In Figure 4, we rescale the x-axis of Figure 3 by the total runtime of each configuration. Comparing MT-PDHG with vanilla PDHG, we observe an additional overhead arising from the computation of the Bregman mixture and multi-timescale extrapolation, which involves evaluating $\sum_s \rho_s X^{k-r_s}$ and $\sum_{i=1}^{r_s} X^{k-i}$. However, as m and n grow, this overhead becomes negligible since the runtime is increasingly dominated by matrix–vector multiplications. We also observe that, approximately, the wall-clock time required to reach a target accuracy scales with $\max_s r_s$ (instead of $\overline{r}$ for the iteration). We note, however, that this behavior may depend on implementation details and the underlying computing hardware.



(a) $(m, n) = (500, 1000)$ (b) $(m, n) = (1000, 3000)$ (c) $(m, n) = (3000, 5000)$

Figure 3: KKT residual as a function of global iteration for different (m, n), under 4 different combinations of the updating rates for the dual blocks: 1. $r_s = 1$ for all s; 2. $r_1 = r_2 = r_3 = 1$ and $r_4 = r_5 = r_6 = 10$; 3. $r_s = 10$ for all s; 4. $r_s = 50$ for all s.



(a) $(m, n) = (500, 1000)$ (b) $(m, n) = (1000, 3000)$ (c) $(m, n) = (3000, 5000)$

Figure 4: KKT residual as a function of running time in seconds for different (m, n), under 4 different combinations of the updating rates for the dual blocks: 1. $r_s = 1$ for all s; 2. $r_1 = r_2 = r_3 = 1$ and $r_4 = r_5 = r_6 = 10$; 3. $r_s = 10$ for all s; 4. $r_s = 50$ for all s.

5.2 Experiment: distributed Support Vector Machine

We consider the Support Vector Machine (SVM) problem with hinge loss and additional regularization. More precisely, each primal $\mathsf{Agent}(x_v)$ is given m_s pairs $\{(b_v^l, y_v^l)\}_{l\in[m_s]}$ such that $b_v^l \in \mathbb{R}^{\overline{d}}$ is a feature vector satisfying $\|b_v^l\| = 1$, and $y_v^l \in \{\pm 1\}$ is the label. The goal of SVM is to find a weight vector $x \in \mathbb{R}^{\overline{d}}$ such that the linear classifier $b \to \mathsf{sign}(\langle b, x\rangle)$ agrees with most pairs (b_v^l, y_v^l) in the dataset. To achieve this, one common approach is to solve the following (regularized) hinge loss minimization problem (in a distributed fashion)[28, 19]:

$$\min_{x\in\overline{\mathcal{X}}} \sum_{v\in V} f_v(x), \quad f_v(x) = \frac{1}{m_s}\sum_{l=1}^{m_s}[1 - y_v^l\langle b_v^l, x\rangle]_+ + \frac{\mu}{2}\|x\|^2,\ v \in V, \tag{31}$$

In this experiment, we use the w8a dataset in LIBSVM [16], which consists of 49749 samples, and the feature dimension is $\overline{d} = 300$. We first normalize the features $\|b_v^l\| = 1$ for all data, and take $\overline{\mathcal{X}} = \{x \in \mathbb{R}^{\overline{d}} | \|x\| \le 5\}$. When $\mu = 0$, (31) is the classical hinge loss minimization problem, and when $\mu > 0$, the local objectives are μ-strongly convex.

Setup. We take $w_x(x) = \frac{1}{2}\|x\|^2$ and $w_{y_s}(y_s) = \frac{1}{2}\|y_s\|^2$, and

$$f_v'(x) = -\frac{1}{m_s}\sum_{l=1}^{m_s} y_v^l b_v^l \cdot \mathbf{1}[1 > y_v^l\langle b_v^l, x\rangle] + \mu x.$$

Since $\|b_v^l\| \le 1$ and $\|x\| \le 5$ (since $x \in \overline{\mathcal{X}}$), we have $\|f_v'\| \le 1 + 5\mu$ and so we can take $M = 2(1 + 5\mu)$.

Below, we look at the suboptimality of $F(\Pi\underline{X}^k)$ where $\underline{X}^k := \frac{\sum_{k'=0}^k \theta_{k'}\widehat{X}^{k'}}{\sum_{k'=0}^k \theta_{k'}}$.

5.2.1 Suboptimality and $k, \overline{r}$

In this experiment, we investigate the suboptimality of $F(\Pi\underline{X}^k)$ as a function of the iteration number k and the mean updating rates for the dual $\overline{r}$.

Communication setup. We consider the hierarchical setup, with 3 layers of dual agents and 1 layer of primal agents, where each non-leaf node has 5 child nodes. Thus, there are $m = 125$ primal agents and $S = 31$ dual agents, and $|\mathsf{Chi}(s)| = 5$ for each non-leaf node. We assume that the dual agents at layer i are updated with rate r_i for $i = 1, 2, 3$, and test with various (r_1, r_2, r_3). We divide the data set evenly among the primal agents.

Algorithm setup. Notice that the μx part in f_v' is common to all v so we can take $a_s = 2\sqrt{|\mathsf{Des}(s)|}$ for all s. We use R_s^{prj} as defined in (21) with $\xi = 1$, and set $x_v^{init} = \mathbf{0}$ and $y_s^{init} = \mathbf{0}$. We test MT-PDHG for $N + 1 = 3000$ and $\mu = 0$. All parameters are set according to Theorem 3.1. We test AMT-PDHG for $N + 1 = 500$ and $\mu = 0.01$. We set $T = N + 1$ for simplicity, and set all other parameters according to Theorem 3.2.

Results. In Figures 5, we present $F(\Pi\underline{X}^k) - F^*$ as a function of k for MT-PDHG and AMT-PDHG respectively, where F^* is the minimum value among all iterations of all algorithm configurations plus 0.001 (as the y-axis is in the log scale).

Different lines correspond to different $(r_1, r_2, r_3, \overline{r})$, with the line colors indicating $\overline{r}$, and the line with starred markers can serve as the benchmark: when all $r_s = 1$, our (A)MT-PDHG has the same updating rules as the classical PDHG.

From the figures, we see that under all settings of $(r_1, r_2, r_3, \overline{r})$, our (A)MT-PDHG converge or show trend of convergence, and the convergence is faster for smaller $\overline{r}$. In addition, comparing the two figures in Figure 5, we see that strong convexity (with AMT-PDHG) indeed accelerates the convergence.

5.2.2 Suboptimality and costs of communication

To further demonstrate the potential benefit of using different updating rates, we rescale the lines in the left plots of Figure 5 based on Amortized Costs (AC): since each of the 5^{i-1} dual agents at layer i is updated every r_i global iteration and each iteration costs c_i, the average cost per global iteration becomes $AC := c_1/r_1 + 5c_2/r_2 + 25c_3/r_3$. In Figures 6 and 7, we rescale the x-axis in Figure 5 using AC for each combinations of r_s, and the resulting plots reflect the objective values as a function of the costs. In addition,

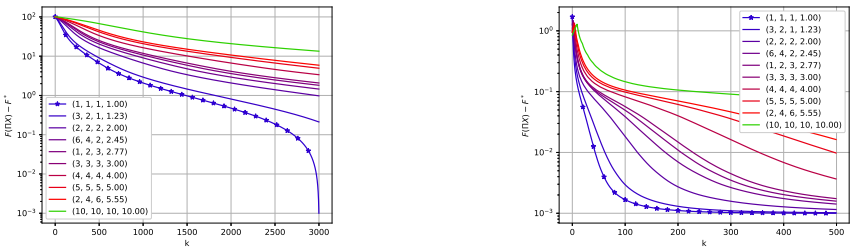


Figure 5: Dependence of $F(\Pi \underline{X}^k)$ on the iteration number k and the mean updating rate $\overline{r}$ for MT-PDHG ($\mu = 0$) (left) and AMT-PDHG ($\mu = 0.01$) with communication sliding. Legends represent $(r_1, r_2, r_3, \overline{r})$ and line colors represent $\overline{r}$.

we use different markers to represent the ratio $r_1 : r_2 : r_3$: circle for $1:1:1$, triangle for $3:2:1$, and star for $1:2:3$.

From Figure 6, we see that the convergence rate for fixed (c_1, c_2, c_3) depends on the ratios of (r_1, r_2, r_3), as the lines with the same ratio (marker type) coincide. In addition, when the costs vary significantly, there is significant benefit of choosing the rates adaptive to the costs. For instance, in the middle two figures, with large c_1 or c_2, MT-PDHG is more cost-efficient when r_1, r_2 are large (as then dual agents in top two layers of the tree are updated less frequently). Indeed, lines with star markers show a faster rate of convergence.

For AMT-PDHG, as suggested by our convergence results in Corollary 4.4, the convergence rates have a more complicated dependency on r_s. From Figure 7, the relative level of (r_1, r_2, r_3) does not determine the convergence rates any more. However, the results still agree with the general intuition that with larger c_1 or c_2, it's more cost-efficient to choose relatively large r_1 or r_2.

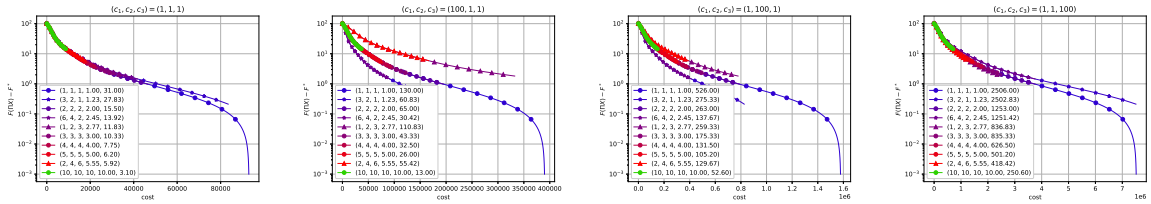


Figure 6: MT-PDHG ($\mu = 0$), dependence of $F(\Pi \underline{X}^k)$ on the costs. Legend represents $(r_1, r_2, r_3, \overline{r}, AC)$ where the amortized cost $AC := c_1/r_1 + 5c_2/r_2 + 25c_3/r_3$. The marker indicates the ratio $r_1 : r_2 : r_3$: circle for $1:1:1$, triangle for $3:2:1$, and star for $1:2:3$.

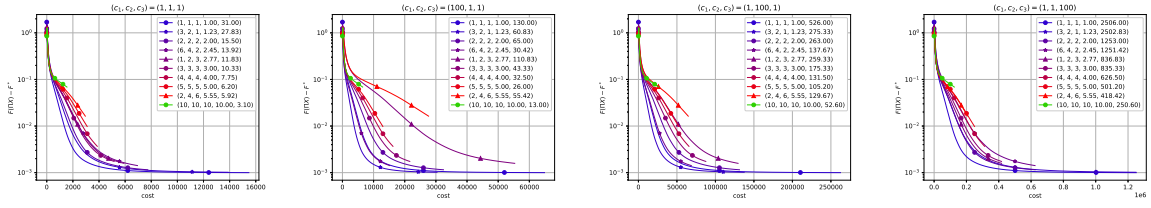


Figure 7: AMT-PDHG ($\mu = 0.01$), dependence of $F(\Pi \underline{X}^k)$ on the costs. Legend represents $(r_1, r_2, r_3, \overline{r}, AC)$ where the amortized cost $AC := c_1/r_1 + 5c_2/r_2 + 25c_3/r_3$. The marker indicates the ratio $r_1 : r_2 : r_3$: circle for $1:1:1$, triangle for $3:2:1$, and star for $1:2:3$.

5.2.3 Suboptimality and k, A

In this experiment, we use 1% of the w8a dataset. We assume that there is only one dual agent (thus $A = \Theta(a_1)$) who updates at rate $r_1 = 1$, and there are $m = 10$ primal agents. We focus on (normalized) $F(\Pi \underline{X}^k)$ as a function of k and function similarities a_1.

Dataset induced function similarities. In this experiment, similarities between $\{f'_v\}_{v\in V}$ are inherited from similarities in the local datasets. More precisely, there is a global dataset $\{(b^l_{global}, y^l_{global})\}_{l\in[m_{global}]}$ consisting of m_{global} pairs of data. In addition, each agent has m_{local} pairs of private data $\{(b^l_{local,v}, y^l_{local,v})\}_{l\in[m_{local}]}$. Thus, $\mathsf{Agent}(x_v)$ has access to $\{(b^l_{global}, y^l_{global})\}_{l\in[m_{global}]} \cup \{(b^l_{local,v}, y^l_{local,v})\}_{l\in[m_{local}]}$, a total of $m_s = m_{global} + m_{local}$ pairs of data.

More concretely, the global dataset consists of $\frac{1-\gamma}{1+(m-1)\gamma}$ fraction of the data, and the rest of the data is divided evenly among m primal agents as local data. Thus $m_{local} \approx \gamma m_s$. We test $\gamma = 0.1, 0.2, \ldots, 0.8$. Due to the global dataset, we have $\|f'_v - f'_{v'}\| \le \frac{m_{local}}{m_s}$ for any $v, v' \in V$, and so we take $a_1 = 2\gamma\sqrt{m}$.

Algorithm setup. We consider two setups.

1. Type-0 setup. We use $R_s = R^{prj}_s$ as defined in (21) with $\xi = 1$, and $y^{init}_s = \mathbf{0}$. We set the parameter $T = N + 1$ and all other parameters are set according to Theorems 3.1 and 3.2. For the initialization, for MT-PDHG, we use $x^{init}_v = \mathbf{0}$ and for AMT-PDHG, we use $\underline{x}^0_v = \mathbf{0}$ and construct x^{init}_v according to Corollary 4.3.

2. Type-1 setup. In addition to the above γ-aware setup, we also test our algorithms for $R_s = 10000 R^{prj}_s$, $a_1 = 2\sqrt{m}$, and $x^{init}_v = \mathbf{0}$, which we denote as type-1 setup. Compared to type-0, type-1 has larger dual domain size and ignores the function similarities. Thus, it can serve as an approximation to the DCS algorithms in [28].

For both types of setups, we test MT-PDHG for $N + 1 = 500$ and $\mu = 0$ and AMT-PDHG for $N + 1 = 200$ and $\mu = 0.01$. Since the datasets are different for different γ, below, for each γ, we normalize $F(\Pi \underline{X}^k)$ such that $F(\mathbf{0}) = m = 500$ is normalized to 1, and the minimum (over k and two types) of $F(\Pi \underline{X}^k)$ is normalized to 0.

Results. In Figure 8, we present the normalized $F(\Pi \underline{X}^k)$ as a function of k for MT-PDHG and AMT-PDHG respectively. Different lines correspond to different types of setup and different γ, with the line colors indicating γ. As can be seen, our MT-PDHG and AMT-PDHG converge in all the tested settings, and strong convexity (with AMT-PDHG) accelerates the convergence.

Moreover, for MT-PDHG, from the left figure in Figure 8 solid curves – representing γ-aware setup – are converging faster than the dotted curves, In addition, as γ decreases – global dataset takes a larger fraction – the algorithm converges faster. These demonstrate that our proposed function similarity dependent penalties indeed take advantage of the function similarities to speed up the convergence.

For the AMT-PDHG, from the right figure in Figure 8, one can see that similarity helps speed up the convergence for both types of setups. We leave it to future works to investigate if the conditions on the penalty levels are necessary for function-similarity dependent convergence rates when the local objectives are strongly convex.

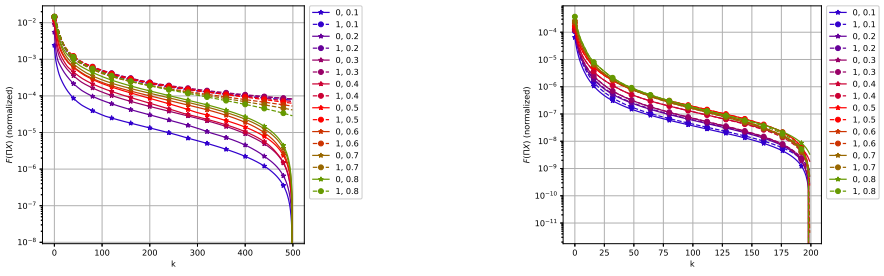


Figure 8: Dependence of normalized $F(\Pi \underline{X}^k)$ on the iteration number k and function similarities γ. Left: MT-PDHG ($\mu = 0$). Right: AMT-PDHG ($\mu = 0.01$). Legends represent (type of setup, γ) and line colors represent γ.

6 Conclusion and future direction

In this work, we propose the (accelerated) multi-timescale PDHG algorithms for saddle point problems with block-decomposable duals. Our (A)MT-PDHG allows arbitrary updating rates for dual blocks while remaining fully deterministic and robust to extreme delays in dual updates. We further apply (A)MT-PDHG to distributed optimization and demonstrate how the flexibility in choosing the updating rates could help improve the overall algorithm efficiencies in heterogeneous environments.

To make the algorithms more practical, one direction of future work is to develop more space-efficient algorithms: currently each primal agents need to store $O(r_{\max})$ vectors in $\mathbb{R}^d$, which are used as mixtures of proximal centers and when calculating messages to dual agents. It is an interesting question whether one can achieve similar convergence guarantees with smaller memory requirement. Another promising direction is to extend the multi-timescale update mechanism to a broader class of algorithms, including those for saddle point and more general optimization problems.

Acknowledgements

This work was funded by the Office of Naval Research grant N00014-24-1-2470.

References

[1] Ahmet Alacaoglu, Olivier Fercoq, and Volkan Cevher. "On the Convergence of Stochastic Primal-Dual Hybrid Gradient". In: *SIAM Journal on Optimization* 32.2 (2022), pp. 1288–1318.

[2] Zeyuan Allen-Zhu, Yang Yuan, and Karthik Sridharan. "Exploiting the structure: stochastic Gradient methods using raw clusters". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2016, pp. 1650–1658.

[3] David Applegate, Oliver Hinder, Haihao Lu, and Miles Lubin. "Faster first-order primal-dual methods for linear programming using restarts and sharpness". In: *Mathematical Programming* 201.1 (Sept. 2023), pp. 133–184.

[4] Yossi Arjevani and Ohad Shamir. "Communication complexity of distributed convex learning and optimization". In: *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 1756–1764.

[5] By Mahmoud Assran, Arda Aytekin, Hamid Reza Feyzmahdavian, Mikael Johansson, and Michael G. Rabbat. "Advances in Asynchronous Parallel and Distributed Optimization". In: *Proceedings of the IEEE* 108.11 (2020), pp. 2013–2031.

[6] Amit Attia, Ofir Gaash, and Tomer Koren. "Faster Stochastic Optimization with Arbitrary Delays via Adaptive Asynchronous Mini-Batching". In: *Proceedings of the 42nd International Conference on Machine Learning*. Ed. by Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu. Vol. 267. Proceedings of Machine Learning Research. PMLR, 13–19 Jul 2025, pp. 1931–1949.

[7] N. S. Aybat, Z. Wang, T. Lin, and S. Ma. "Distributed Linearized Alternating Direction Method of Multipliers for Composite Convex Consensus Optimization". In: *IEEE Transactions on Automatic Control* 63.1 (2018), pp. 5–20.

[8] Albert S. Berahas, Raghu Bollapragada, Nitish Shirish Keskar, and Ermin Wei. "Balancing Communication and Computation in Distributed Optimization". In: *IEEE Transactions on Automatic Control* 64.8 (2019), pp. 3141–3155.

[9] Dimitri P. Bertsekas. "Incremental proximal methods for large scale convex optimization". In: *Mathematical Programming* 129.2 (Oct. 2011), pp. 163–195.

[10] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. USA: Prentice-Hall, Inc., 1989.

[11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. 2011.

[12] Francesco Bullo, Jorge Cortés, and Sonia Martínez. *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. 2009.

[13] Antonin Chambolle, Matthias J. Ehrhardt, Peter Richtárik, and Carola-Bibiane Schönlieb. "Stochastic Primal-Dual Hybrid Gradient Algorithm with Arbitrary Sampling and Imaging Applications". In: *SIAM Journal on Optimization* 28.4 (2018), pp. 2783–2808.

[14] Antonin Chambolle and Thomas Pock. "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging". In: *Journal of Mathematical Imaging and Vision* 40.1 (May 2011), pp. 120–145.

[15] Antonin Chambolle and Thomas Pock. "On the ergodic convergence rates of a first-order primal–dual algorithm". In: *Mathematical Programming* 159.1 (Sept. 2016), pp. 253–287.

[16] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011). Software available at `http://www.csie.ntu.edu.tw/~c` 27:1–27:27.

[17] Alon Cohen, Amit Daniely, Yoel Drori, Tomer Koren, and Mariano Schain. "Asynchronous Stochastic Optimization Robust to Arbitrary Delays". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 9024–9035.

[18] Dominik Csiba and Peter Richtárik. "Importance Sampling for Minibatches". In: *Journal of Machine Learning Research* 19.27 (2018), pp. 1–21.

[19] John C. Duchi, Alekh Agarwal, and Martin J. Wainwright. "Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling". In: *IEEE Transactions on Automatic Control* 57.3 (2012), pp. 592–606.

[20] Saeed Ghadimi and Guanghui Lan. "Optimal Stochastic Approximation Algorithms for Strongly Convex Stochastic Composite Optimization I: A Generic Algorithmic Framework". In: *SIAM Journal on Optimization* 22.4 (2012), pp. 1469–1492.

[21] Hassan Jaleel and Jeff S. Shamma. "Distributed Optimization for Robot Networks: From Real-Time Convex Optimization to Game-Theoretic Self-Organization". In: *Proceedings of the IEEE* 108.11 (2020), pp. 1953–1967.

[22] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. virtual conference: PMLR, 13–18 Jul 2020, pp. 5132–5143.

[23] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. "A Unified Theory of Decentralized SGD with Changing Topology and Local Updates". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. virtual conference: PMLR, 13–18 Jul 2020, pp. 5381–5393.

[24] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. *Federated Optimization: Distributed Machine Learning for On-Device Intelligence*. 2016.

[25] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. "Federated Learning: Strategies for Improving Communication Efficiency". In: *NIPS Workshop on Private Multi-Party Machine Learning*. 2016.

[26] Dmitry Kovalev, Aleksandr Beznosikov, Ekaterina Borodich, Alexander Gasnikov, and Gesualdo Scutari. "Optimal gradient sliding and its application to distributed optimization under similarity". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS '22. New Orleans, LA, USA: Curran Associates Inc., 2022.

[27] Guanghui Lan. "Gradient sliding for composite optimization". In: *Mathematical Programming* 159.1 (Sept. 2016), pp. 201–235.

[28] Guanghui Lan, Soomin Lee, and Yi Zhou. "Communication-efficient algorithms for decentralized and stochastic optimization". In: *Mathematical Programming* 180.1 (Mar. 2020), pp. 237–284.

[29] Dan Li, K.D. Wong, Yu Hen Hu, and A.M. Sayeed. "Detection, classification, and tracking of targets". In: *IEEE Signal Processing Magazine* 19.2 (2002), pp. 17–29.

[30] Ji Liu and Stephen J. Wright. "Asynchronous Stochastic Coordinate Descent: Parallelism and Convergence Properties". In: *SIAM Journal on Optimization* 25.1 (2015), pp. 351–376.

[31] Carl D. Meyer. "Generalized Inversion of Modified Matrices". In: *SIAM Journal on Applied Mathematics* 24.3 (1973), pp. 315–323.

[32] Daniel K. Molzahn, Florian Dörfler, Henrik Sandberg, Steven H. Low, Sambuddha Chakrabarti, Ross Baldick, and Javad Lavaei. "A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems". In: *IEEE Transactions on Smart Grid* 8.6 (2017), pp. 2941–2962.

[33] Angelia Nedić and Ji Liu. "Distributed Optimization for Control". In: *Annual Review of Control, Robotics, and Autonomous Systems* 1.Volume 1, 2018 (2018), pp. 77–103.

[34] Angelia Nedić and Alex Olshevsky. "Distributed Optimization Over Time-Varying Directed Graphs". In: *IEEE Transactions on Automatic Control* 60.3 (2015), pp. 601–615.

[35] Angelia Nedić, Alex Olshevsky, and Michael G. Rabbat. "Network Topology and Communication-Computation Tradeoffs in Decentralized Optimization". In: *Proceedings of the IEEE* 106.5 (2018), pp. 953–976.

[36] Angelia Nedic and Asuman Ozdaglar. "Distributed Subgradient Methods for Multi-Agent Optimization". In: *IEEE Transactions on Automatic Control* 54.1 (2009), pp. 48–61.

[37] Feng Niu, Benjamin Recht, Christopher Re, and Stephen J. Wright. "HOGWILD! a lock-free approach to parallelizing stochastic gradient descent". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems*. NIPS'11. Granada, Spain: Curran Associates Inc., 2011, pp. 693–701.

[38] Zhimin Peng, Yangyang Xu, Ming Yan, and Wotao Yin. "ARock: An Algorithmic Framework for Asynchronous Parallel Coordinate Updates". In: *SIAM Journal on Scientific Computing* 38.5 (2016), A2851–A2879.

[39] Roger J. B. Wets R. Tyrrell Rockafellar. *Variational Analysis*. Berlin, Germany: Springer Science & Business Media, 2009.

[40] M. Rabbat and R. Nowak. "Distributed optimization in sensor networks". In: *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004.* 2004, pp. 20–27.

[41] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. "Optimal Convergence Rates for Convex Distributed Optimization in Networks". In: *Journal of Machine Learning Research* 20.159 (2019), pp. 1–31.

[42] Shai Shalev-Shwartz and Tong Zhang. "Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization". In: *Journal of Machine Learning Research* 14.16 (2013), pp. 567–599.

[43] Ohad Shamir, Nati Srebro, and Tong Zhang. "Communication-Efficient Distributed Optimization using an Approximate Newton-type Method". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1000–1008.

[44] Ola Shorinwa, Trevor Halsted, Javier Yu, and Mac Schwager. "Distributed Optimization Methods for Multi-Robot Systems: Part 1—A Tutorial [Tutorial]". In: *IEEE Robotics & Automation Magazine* 31.3 (2024), pp. 121–138.

[45] Boyd Stephen, Parikh Neal, Chu Eric, Peleato Borja, and Eckstein Jonathan. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends in Information Retrieval* 3.1 (July 2011), pp. 1–122.

[46] Sebastian Urban Stich. "Local SGD Converges Fast and Communicates Little". In: *ICLR 2019-International Conference on Learning Representations*. 2019.

[47] Tao Sun, Robert Hannah, and Wotao Yin. "Asynchronous coordinate descent under more realistic assumption". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6183–6191.

[48] Ye Tian, Gesualdo Scutari, Tianyu Cao, and Alexander Gasnikov. "Acceleration in Distributed Optimization under Similarity". In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Ed. by Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera. Vol. 151. Proceedings of Machine Learning Research. virtual conference: PMLR, 28–30 Mar 2022, pp. 5721–5756.

[49] J. Tsitsiklis, D. Bertsekas, and M. Athans. "Distributed asynchronous deterministic and stochastic gradient optimization algorithms". In: *IEEE Transactions on Automatic Control* 31.9 (1986), pp. 803–812.

[50] Alexander Tyurin and Peter Richtárik. "On the Optimal Time Complexities in Decentralized Stochastic Asynchronous Optimization". In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang. Vol. 37. Red Hook,NY: Curran Associates, Inc., 2024, pp. 122652–122705.

[51] Santosh S. Vempala, Ruosong Wang, and David P. Woodruff. "The Communication Complexity of Optimization". In: *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2020, pp. 1733–1752.

[52] Ermin Wei and Asuman Ozdaglar. *On the O(1/k) Convergence of Asynchronous Distributed Alternating Direction Method of Multipliers*. 2013.

[53] Blake E Woodworth, Kumar Kshitij Patel, and Nati Srebro. "Minibatch vs Local SGD for Heterogeneous Distributed Learning". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Red Hook, NY: Curran Associates, Inc., 2020, pp. 6281–6292.

[54] Stephen J. Wright. "Coordinate descent algorithms". In: *Mathematical Programming* 151.1 (June 2015), pp. 3–34.

[55] Qunsong Zeng, Yuqing Du, Kaibin Huang, and Kin K. Leung. "Energy-Efficient Resource Management for Federated Edge Learning With CPU-GPU Heterogeneous Computing". In: *IEEE Transactions on Wireless Communications* 20.12 (2021), pp. 7947–7962.

[56] Yuchen Zhang and Xiao Lin. "DiSCO: Distributed Optimization for Self-Concordant Empirical Loss". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 362–370.

[57] Yuchen Zhang and Xiao Lin. "Stochastic Primal-Dual Coordinate Method for Regularized Empirical Risk Minimization". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 353–361.

[58] Martin A. Zinkevich, Markus Weimer, Alex Smola, and Lihong Li. "Parallelized stochastic gradient descent". In: *Proceedings of the 24th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'10. Vancouver, British Columbia, Canada: Curran Associates Inc., 2010, pp. 2595–2603.

A Generalized gradient sliding procedure

In Algorithm 3, we provide the generalized gradient sliding procedure. Then we provide the proofs of its properties.

Algorithm 3 Generalized gradient sliding procedure

Input: The sequences $\{\beta_t\}$ and $\{\lambda_t\}$, $\phi' : U \to \mathbb{R}^{d_0}$ a subgradient oracle for ϕ.
Output: $(u^T, \widehat{u}^T) = GS(\phi, U, D, T, (\eta_i)_{i\in\mathcal{I}}, v, (x_i)_{i\in\mathcal{I}}, x^{init})$, an approximate solution to

$$\min_{u\in U} \Phi(u) := \langle v, u\rangle + \phi(u) + \sum_{i\in\mathcal{I}} \eta_i D(u, x_i)$$

$(u^0, \widehat{u}^0) \leftarrow (x^{init}, x^{init})$, $\eta \leftarrow \sum_{i\in\mathcal{I}} \eta_i$
for $t = 1, \ldots, T$ **do**

$$u^t = \operatorname*{argmin}_{u\in U} \langle v + \phi'(u^{t-1}), u\rangle + \sum_{i\in\mathcal{I}} \eta_i D(u, x_i) + \eta\beta_t D(u, u^{t-1})$$

end for

$$\widehat{u}^T = (\sum_{t=1}^{T} \lambda_t)^{-1} \sum_{t=1}^{T} \lambda_t u^t.$$

Lemma A.1 (generalized lemma 5 in [28]). *Let the convex function $q : U \to \mathbb{R}$, and $\mathcal{I}$ an arbitrary finite index set. Assume that the points $x_i \in U$ and the numbers $\eta_i \geq 0$ for $i \in \mathcal{I}$. Let $w : U \to \mathbb{R}$ be a distance generating function and*

$$u^* \in \operatorname*{argmin}_{u\in U} q(u) + \sum_{i\in\mathcal{I}} \eta_i D(u, x_i).$$

Then for any $u \in U$, we have

$$q(u^*) + \sum_{i\in\mathcal{I}} \eta_i D(u^*, x_i) \leq q(u) + \sum_{i\in\mathcal{I}} \eta_i D(u, x_i) - \sum_{i\in\mathcal{I}} \eta_i D(u, u^*).$$

Proof of Lemma A.1. First, by the optimality condition for u^*, there exists $q'(u^*) \in \partial q(u^*)$ such that

$$\langle q'(u^*) + \sum_{i\in\mathcal{I}} \eta_i \nabla D(u^*, x_i), u - u^*\rangle \geq 0, \quad \forall u \in U.$$

By definition, we have for each $i \in \mathcal{I}$ that

$$D(u, x_i) - D(u^*, x_i) - D(u, u^*) = \langle \nabla w(x_i) - \nabla w(u^*), u - u^*\rangle = -\langle \nabla D(u^*, x_i), u - u^*\rangle$$

Thus, we have for any $u \in U$,

$$\begin{aligned}
&q(u) + \sum_{i\in\mathcal{I}} \eta_i D(u, x_i) \\
&\geq q(u^*) + \langle q'(u^*), u - u^*\rangle + \sum_{i\in\mathcal{I}} \eta_i \left(D(u^*, x_i) + D_w(u, u^*) - \langle \nabla D(u^*, x_i), u - u^*\rangle\right) \\
&\geq q(u^*) + \sum_{i\in\mathcal{I}} \eta_i D(u^*, x_i) + \sum_{i\in\mathcal{I}} \eta_i D(u, u^*).
\end{aligned}$$

□

Lemma A.2. *Assume that $U \subset \mathbb{R}^{d_0}$ is a convex set, and $\phi : U \to \mathbb{R}$ is a convex function such that*

$$\frac{\mu}{2}\|x - y\|^2 \leq \phi(x) - \phi(y) - \langle \phi'(y), x - y\rangle \leq M\|x - y\|, \quad \forall x, y \in U,$$

where $\phi' : U \to \mathbb{R}^{d_0}$ is a subgradient oracle, i.e. for each $y \in U$, $\phi'(y) \in \partial\phi(y)$ is a subgradient. In addition, $D_{w^x}(x, x') \leq \frac{C}{2}\|x - x'\|^2$ for some $C \in [0, \infty]$. If $\{\beta_t\}$ and $\{\lambda_t\}$ in Algorithm 3 satisfies that

$$\lambda_{t+1}(\eta\beta_{t+1} - \mu/C) \leq \lambda_t(1 + \beta_t)\eta, \quad \forall t \geq 1,$$

then for $t \geq 1$ and $u \in U$

$$(\sum_{t=1}^{T} \lambda_t) \cdot (\Phi(\widehat{u}^T) - \Phi(u)) \leq (\eta\beta_1 - \mu/C)\lambda_1 D(u, u^0) - \eta(1+\beta_T)\lambda_T D(u, u^T) + \sum_{t=1}^{T} \frac{M^2\lambda_t}{2\eta\beta_t}.$$

Proof of Lemma A.2. Applying Lemma A.1, and using $\sum_{i\in\mathcal{I}} \eta_i = \eta$, we have

$$\begin{aligned}&\langle v + \phi'(u^{t-1}), u^t - u\rangle + \sum_{i\in\mathcal{I}} \eta_i D(u^t, x_i) - \sum_{i\in\mathcal{I}} \eta_i D(u, x_i)\\ &\leq \eta\beta_t D(u, u^{t-1}) - \eta\beta_t D(u^t, u^{t-1}) - (1+\beta_t)\eta D(u, u^t)\end{aligned}$$

The rest follows a similar argument as in the proof of Proposition 2 [28]. □

As a corollary to Lemma A.2, we have the following performance guarantee.

Corollary A.1. *Assume that $U \subset \mathbb{R}^{d_0}$ is a convex set, and $\phi : U \to \mathbb{R}$ is a convex function such that*

$$\frac{\mu}{2}\|x-y\|^2 \leq \phi(x) - \phi(y) - \langle \phi'(y), x-y\rangle \leq M\|x-y\|, \quad \forall x, y \in U,$$

where $\phi' : U \to \mathbb{R}^{d_0}$ is a subgradient oracle, i.e. for each $y \in U$, $\phi'(y) \in \partial\phi(y)$ is a subgradient. With $\lambda_t = t+1$ and $\beta_t = \frac{t}{2}$ for $t \geq 1$, we have for any $u \in U$

$$\begin{aligned}\langle v, \widehat{u}^T - u\rangle + \phi(\widehat{u}^T) - \phi(u) \leq{}& \frac{2\eta}{T(T+3)} D(u, x^{init}) + \sum_{i\in\mathcal{I}} \eta_i D(u, x_i)\\ &- \frac{(T+1)(T+2)}{T(T+3)}\eta D(u, u^T) - \sum_{i\in\mathcal{I}} \eta_i D(\widehat{u}^T, x_i) + \frac{4M^2}{\eta(T+3)}.\end{aligned}$$

Further, if $\mu > 0$, and $D_{w^x}(x, x') \leq \frac{C}{2}\|x-x'\|^2$ for some $C < \infty$, then denoting $\eta = \sum_{i\in\mathcal{I}} \eta_i$, setting $\lambda_t = t$ and $\beta_t = \frac{(t+1)\mu}{2\eta C} + \frac{t-1}{2}$, we have for any $u \in U$,

$$\begin{aligned}\langle v, \widehat{u}^T - u\rangle + \phi(\widehat{u}^T) - \phi(u) \leq{}& \sum_{i\in\mathcal{I}} \eta_i D(u, x_i) - \sum_{i\in\mathcal{I}} \eta_i D(\widehat{u}^T, x_i)\\ &- (\frac{\mu}{C} + \eta) D(u, u^T) + \frac{2M^2/\eta}{T(T+1)} \sum_{t=1}^{T} \frac{\lambda_t}{\beta_t},\end{aligned}$$

and $\frac{2M^2/\eta}{T(T+1)} \sum_{t=1}^{T} \frac{\lambda_t}{\beta_t} \leq \frac{4CM^2}{\mu(T+1)}$.

B Proof for Section 4.2

Proof of Lemma 4.1. For $\overline{d} = 1$, the matrix representation of $K_s \in \mathbb{R}^{|\mathsf{Chi}(s)|\times m}$ is $K_s = (I - \mathbf{1}(\frac{|\mathsf{Des}(j)|}{|\mathsf{Des}(s)|})^T_{j\in\mathsf{Chi}(s)})P_s$ where $P_s \in \mathbb{R}^{|\mathsf{Chi}(s)|\times m}$, and $P_s(i,j) = |\mathsf{Des}(i)|^{-1}$ if $j \in \mathsf{Des}(i)$ and $P_s(i,j) = 0$ otherwise. Notice that

$$K_s K^*_{s'} = (I - \mathbf{1}(\frac{|\mathsf{Des}(j)|}{|\mathsf{Des}(s)|})^T_{j\in\mathsf{Chi}(s)}) P_s P^T_{s'} (I - \mathbf{1}(\frac{|\mathsf{Des}(j)|}{|\mathsf{Des}(s')|})^T_{j\in\mathsf{Chi}(s')})^T.$$

If s is not in the subtree rooted at s' and s' is not in the subtree rooted at s, then $\mathsf{Des}(s) \cap \mathsf{Des}(s') = \emptyset$, and so $P_s P^T_{s'} = \mathbf{0}$. If s is in the subtree rooted at s', then s is in the subtree rooted at some $\widehat{s} \in \mathsf{Chi}(s')$. In particular, $(P_s P^T_{s'})(i,j) = 0$ for all $j \neq \widehat{s}$ and $(P_s P^T_{s'})(i,\widehat{s}) = |\mathsf{Des}(\widehat{s})|^{-1}$, and thus $(I - \mathbf{1}(\frac{|\mathsf{Des}(j)|}{|\mathsf{Des}(s)|})^T_{j\in\mathsf{Chi}(s)}) P_s P^T_{s'} = \mathbf{0}$. Similarly for the case when s' is in the subtree rooted at s. The case when $\overline{d} > 1$ follows by applying the above argument coordinate-wise.

For the second claim, consider the case $\overline{d} = 1$, denoting

$$D = \mathsf{diag}((|\mathsf{Des}(j)|)_{j\in\mathsf{Chi}(s)}), \quad v = (|\mathsf{Des}(j)|)_{j\in\mathsf{Chi}(s)}/\|(|\mathsf{Des}(j)|)_{j\in\mathsf{Chi}(s)}\|_2,$$

where the norm in the denominator in the definition of v is the l_2 norm. Then when $\overline{d} = 1$, we have (applying Theorem 6 in [31])

$$K_s K_s^* = D^{-1} - \frac{1}{|\mathsf{Des}(s)|}\mathbf{1}\mathbf{1}^T, \quad (K_s K_s^*)^\dagger = D - vv^T D - Dvv^T + (v^T Dv)vv^T. \tag{32}$$

Thus, noticing that $v^T K_s = \mathbf{0}$, we have

$$\Pi_s = K_s^*(K_s K_s^*)^\dagger K_s = K_s^T D K_s.$$

Thus, for any $\widetilde{X}, \widehat{X} \in \mathbb{R}^m$

$$\langle \widehat{X}, \Pi_s \widetilde{X}\rangle = \langle \Pi_s \widehat{X}, \Pi_s \widetilde{X}\rangle = (K_s\widehat{X})^T D(K_s\widetilde{X}) = \sum_{i\in\mathsf{Chi}(s)} |\mathsf{Des}(i)| \cdot \langle (K_s\widehat{X})_i, (K_s\widetilde{X})_i\rangle.$$

The above argument can be applied coordinate-wise, and so extend to $\overline{d} \geq 1$. □