Decentralized Multi-Task Stochastic Optimization With Compressed Communications ⋆

Navjot Singh [a], Xuanyu Cao [b], Suhas Diggavi [a], Tamer Başar [c]

[a] *Department of Electrical and Computer Engineering, University of California Los Angeles, USA*

[b] *Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong*

[c] *Coordinated Science Laboratory, University of Illinois Urbana-Champaign, USA*

Abstract

We consider a multi-agent network where each node has a stochastic (local) cost function that depends on the decision variable of that node and a random variable, and further, the decision variables of neighboring nodes are pairwise constrained. There is an aggregated objective function for the network, composed additively of the expected values of the local cost functions at the nodes, and the overall goal of the network is to obtain the minimizing solution to this aggregate objective function subject to all the pairwise constraints. This is to be achieved at the level of the nodes using decentralized information and local computation, with exchanges of only compressed information allowed by neighboring nodes. The paper develops algorithms and obtains performance bounds for two different models of local information availability at the nodes: (i) sample feedback, where each node has direct access to samples of the local random variable to evaluate its local cost, and (ii) bandit feedback, where samples of the random variables are not available, but only the values of the local cost functions at two random points close to the decision are available to each node. For both models, with compressed communication between neighbors, we have developed decentralized saddle-point algorithms that deliver performances no different (in order sense) from those without communication compression; specifically, we show that deviation from the global minimum value and violations of the constraints are upper-bounded by $O(T^{-\frac{1}{2}})$ and $O(T^{-\frac{1}{4}})$, respectively, where T is the number of iterations. Numerical examples provided in the paper corroborate these bounds.

Key words: Decentralized stochastic optimization, saddle-point algorithm, compression, sample feedback, bandit feedback

1 Introduction

The emergence of multi-agent networks and the need to distribute computation across different nodes which have access to only piece of the network-wide data but are allowed to exchange information under some resource constraints, have accelerated research efforts on decentralized and distributed optimization in multiple communities, particularly during the last 10-15 years. Spearheading this activity has been decentralized consensus optimization in static settings, where the goal is to minimize the sum of local cost functions, toward which [1] proposed a decentralized sub-gradient algorithm, whose convergence was further studied in [2]. Following this initial work, several other consensus algorithms were introduced and studied, including alternating direction method of multipliers (ADMM) [4], exact first-order algorithm [5], stochastic consensus optimization [6,7], and online consensus optimization with time-varying cost functions [8,9].

Consensus modeling framework requires, in essence, all nodes to converge to the same value. This however may not be appropriate in many network scenarios, where different nodes, even neighboring ones, may ultimately end up with different decision (or action) values. Such a scenario arises in, for example, distributed multitask adaptive signal processing, where the weight vectors at neighboring nodes are not the same [10,11]. One of the first papers that has analyzed such departure from consensus optimization is [12], where the formulation included proximity constraints between neighboring nodes, which were handled through construction of Lagrangians and

⋆ The work of NS and SD was supported in part by NSF grants 2007714, 2139304, and the Army Research Laboratory (ARL) grant W911NF-17-2-0196.

Email addresses: navjotsingh@ucla.edu (Navjot Singh), eexcao@ust.hk (Xuanyu Cao), suhas@ee.ucla.edu (Suhas Diggavi), basar1@illinois.edu (Tamer Başar).

using saddle-point algorithms, and extended to the asynchronous setting in [13].

Decentralized algorithms are built on the assumption that there is some exchange of information among the nodes (at least among the neighboring nodes) which then propagates across the network towards achieving the global optimum in the limit. Extensive and frequent exchange of such information is generally practically impossible (due to bandwidth constraints on the edges of the underlying network which constitute the communication links, and computation and storage limitations, among many others), which inherently brings in a restriction on the amount and timing of the exchange of relevant current data. In the literature several studies have addressed these limitations through quantization of information or actions [14–18], by using only sign information on some differences [19,20], by controlling the timing of transmissions through event triggering [21, 46, 52], or by sparsification [22, 23, 29]. Quantization in the context of decentralized optimization (and not consensus problems) has also been studied, with some of the algorithms leading to nonzero errors in convergence (see the early work [25, 26]) and others to exact convergence [27]; see also [28] for quantized stochastic optimization. Some recent work has also used error-compensated compression in decentralized optimization, such as [29,30,51,52]. Recently, error-compensated compressed decentralized training for *online* convex optimization was considered in [53].

Most of the existing works on decentralized optimization with quantized/compressed communications are, as discussed above, focused on either *consensus* optimization or *unconstrained* optimization. Research departing from that trend was initiated in [32], which addressed the problem of multitask learning (or distributed optimization with pairwise constraints) using quantized communications. More specifically, the model adopted in that paper (with an underlying network topology) associated with each node a stochastic (local, individual) cost and with each pair of neighbors an inequality constraint, e.g., proximity constraint. Note that in such a formulation, different from consensus problems, each node has its own decision variable, but these cannot be picked independently because of the pairwise constraints. Further, the distribution of the random variable in the stochastic local cost function of each node is unknown and each node operates based on sequential feedback information, rendering the formulation distinct from deterministic optimization. The paper developed stochastic saddle-point algorithms with quantized communications between neighbors, and studied the impact of quantization on the optimization performance. One shortcoming of the result of [32] is that the scheme developed led to nonzero convergence error; said differently, the algorithm in that work does not lead to convergence to the exact optimal solution as the number of iterations grow. This is precisely the issue we address in this paper, and achieve exact convergence by employing a saddle-point algorithm along with an approach based on error-compensated communication compression. Before further discussing the contents and contributions of this paper, let us point out that saddle-point algorithms (a.k.a. primal-dual algorithms) have been extensively used in literature on constrained optimization, such as deterministic centralized optimization [33, 34], decentralized optimization [35], stochastic optimization [12, 13, 36], and online optimization [37, 38].

1.1 Contributions

In this paper, we address the problem of decentralized multi-agent stochastic optimization on a network, where each agent has a local stochastic convex cost function and each pair of neighbors is associated with an inequality constraint. The overall goal is to minimize the total (additive) expected cost of all agents subject to all the constraints on all edges, with all computation carried out at the nodes and with information exchanged among the nodes using compressed communication. We consider two scenarios of interest based on the sample information available locally at the nodes:

- *Sample Feedback:* Each node has access to the local samples of the random variable affecting its local cost function drawn from its distribution at any time instance during the optimization process, and can thus evaluate its cost function and its gradient.
- *Bandit Feedback:* Nodes do not have access to the samples, but rather only observe values of the corresponding local cost functions at two points sufficiently close to the original node parameter. For references on bandit feedback in context of optimization (a.k.a. zeroth-order optimization), see [39–44].

Under both scenarios, the paper develops a decentralized saddle-point algorithm which leads to zero convergence error, even with a *finite* number of bits for each iteration. Note that previous works in this topic [32] required the number of bits to be unbounded for the error to diminish. Specifically, under some standard assumptions, we show that the expected sub-optimality and the expected constraint violations are upper bounded by $O(T^{-\frac{1}{2}})$ and $O(T^{-\frac{1}{4}})$, respectively, where T is the number of iterations, despite the proposed algorithm using a finite number of bits. These bounds match, in order sense, the bounds for algorithms without communication compression. Hence, we get near optimal optimization performance even with finite number of bits under both scenarios. The paper also provides results of numerical experiments, which corroborate these bounds.

Accordingly, the main contributions of this paper are:

- Using finite bit compressed sample feedback, with T being the horizon of the optimization problem,

achieving $O(1/\sqrt{T})$ closeness to optimum value of the objective function, and achieving $O(T^{-\frac{1}{4}})$ constraint violation—both being the same as in the case without compression.
- Obtaining the same order bounds with bandit feedback, using only two-point feedback values.

1.2 Paper Organization

The rest of the paper is organized as follows: Section 2 provides a precise formulation of the problem under consideration. Section 3 develops the saddle-point algorithm (Algorithm 1) under sample feedback, and provides convergence results and performance bounds (Theorem 1) along with essential points of the analyses and proofs. Section 4 presents the counterpart of Section 3 for bandit feedback, with the corresponding algorithm (Algorithm 2) and corresponding main result on convergence and performance bounds (Theorem 2). Section 5 discusses results of some numerical experiments. Section 6 provides some concluding remarks. Omitted technical details can be found in the arXiv version [59].

2 Problem Formulation

We consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ nodes and $|\mathcal{E}| = m$ connected edges. We assume that each connected node pair $(i, j) \in \mathcal{E}$ allows for bi-directional communication from i to j and j to i. The neighbor set of the node i is denoted by $\mathcal{N}_i$.

Associated with each node $i \in [n] := \{1, 2, \ldots, n\}$ is an unknown data distribution which we denote by $\mathcal{P}_i$. The samples generated from the distribution are denoted by $\xi_i \sim \mathcal{P}_i$ where $\xi_i \in \Xi_i$. Each node also has a local cost function $f_i : \mathcal{X} \times \Xi_i \to \mathbb{R}^+$ which takes as input a sample $\xi_i \in \Xi_i$ and a local parameter $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ to yield the sample cost $f_i(\mathbf{x}_i, \xi_i)$. Here, the set $\mathcal{X}$ corresponds to the set of feasible parameters the node can choose from, which is the same across all nodes. As an example, for supervised image recognition tasks, the sample ξ_i for a node i may correspond to an image-label pair with the set $\mathcal{X}$ being the set of all neural networks with a width of 2 layers and $\mathbf{x}_i$ a particular 2-layer neural network. The local objective f_i in this case may denote a cross-entropy loss function evaluated using the given image-label pair and the neural network. The expected cost for a node i for parameter $\mathbf{x}_i \in \mathcal{X}$ is denoted by $F_i(\mathbf{x}_i) = \mathbb{E}_{\xi_i \sim \mathcal{P}_i}[f_i(\mathbf{x}_i, \xi_i)]$. In general, we are interested in minimizing the expected cost for all the nodes $i \in [n]$. That is, we are interesting in finding node parameters $\{\mathbf{x}_i\}_{i=1}^n$ that minimize the cost $F(\mathbf{x}) := \sum_{i=1}^n F_i(\mathbf{x}_i)$ where $F_i(\mathbf{x}_i)$ denotes the expected cost of the node i evaluated using parameter $\mathbf{x}_i$ and $\mathbf{x} \in \mathcal{X}^n$ denotes stacking of all the individual node parameters $\{\mathbf{x}_i\}_{i=1}^n$. Further, we assume that the node parameters are related via pairwise constraints on the connected nodes in the graph. Specifically, for any $i \in [n]$ and $j \in \mathcal{N}_i$, there is a function $g_{ij} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that the inequality $g_{ij}(\mathbf{x}_i, \mathbf{x}_j) \leq 0$ should be satisfied. This may, for example, encode a proximity constraint on the node parameters by having $g_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 - c_{ij}$ where $\|.\|_2$ denotes the ℓ_2 norm and $c_{ij} \geq 0$ is a constant. In this paper, we assume that the constraint functions $g_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ are symmetric in their parameters, i.e., $g_{ij}(\mathbf{x}_i, \mathbf{x}_j) = g_{ji}(\mathbf{x}_j, \mathbf{x}_i)$ for all $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ and connected node pairs $(i, j) \in \mathcal{E}$, which leads to m number of distinct pairwise constraints for all the parameters. With the notation now in place, we state the learning objective for the multi-task problem can be stated as follows:

$$\begin{aligned} \min_{\mathbf{x} \in \mathcal{X}^n} F(\mathbf{x}) &= \sum_{i=1}^n \mathbb{E}_{\boldsymbol{\xi}_i}[f_i(\mathbf{x}_i, \boldsymbol{\xi}_i)] \\ \text{subject to} \quad & g_{ij}(\mathbf{x}_i, \mathbf{x}_j) \leq 0, \quad \forall i \in [n], j \in \mathcal{N}_i \end{aligned} \tag{1}$$

To solve the problem in (1) in a decentralized manner, the nodes need to communicate during the optimization procedure which can be prohibitive for low bandwidth links or when the exchanged information updates among the nodes are large. To this end, in this paper we consider compression of the information exchanges among the nodes to make the communication efficient. We employ the notion of the compression operator proposed in [23]:

Definition 1. A (possibly randomized) function $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is called a compression operator, if there exists a constant $\omega \in (0, 1)$, such that for every $\mathbf{x} \in \mathbb{R}^d$:

$$\mathbb{E}\|\mathbf{x} - \mathcal{C}(\mathbf{x})\|_2^2 \leq (1 - \omega)\|\mathbf{x}\|_2^2 \tag{2}$$

where expectation is taken over the randomness of $\mathcal{C}$. We assume $\mathcal{C}(\mathbf{0}) = \mathbf{0}$.

Many important sparsifiers and quantizers in the literature satisfy the above definition, few of them being: (i) Top_k and $Rand_k$ sparsifiers [23] (where only k entries out of d are non-zero) with $\omega = \frac{k}{d}$, (ii) Stochastic quantizer QSGD [28] with $\omega = (1 - \beta_{d,s})$ for $\beta_{d,s} := \min\left(\frac{d}{s^2}, \frac{\sqrt{d}}{s}\right) < 1$, (iii) The scaled $Sign$ quantizer [45] with $\omega = \frac{\|\mathbf{x}\|_1^2}{d\|\mathbf{x}\|_2^2}$ for vector $\mathbf{x} \in \mathbb{R}^d$, and (iv) composed quantization and sparsification operators in [31] with $\omega = \left(1 - \frac{k}{d(1+\beta_{k,s})}\right)$.

We consider two scenarios of interest based on the sampled information available locally at the nodes:

(i) Sample Feedback: In this scenario we assume that each node i has access to the local samples ξ_i drawn from $\mathcal{P}_i$ at any time instance during the optimization procedure and can thus evaluate the cost function and its derivative.

(ii) Bandit Feedback: In this scenario, nodes do not have a direct access to the samples, but rather can only observe values of the local cost function at two perturbations from the original node parameter.

We focus on these scenarios separately in Section 3 and Section 4 respectively, where we develop a compressed decentralized algorithm for optimizing (1) for each, and present our theoretical convergence results.

3 Decentralized compressed optimization with Sample feedback

In this section we describe our approach for optimizing the objective in (1) for the case of *sample feedback*. In this setting, each node $i \in [n]$ has access to the sampled instance ξ_i at any stage of the optimization procedure, and thus can evaluate the local objective $f_i(\mathbf{x}_i, \xi_i)$ based on its local parameter $\mathbf{x}_i$.

3.1 Algorithm: with Sample Feedback

We develop a stochastic saddle-point algorithm for solving (1) in a decentralized manner with compressed parameter exchanges. Our proposed scheme is presented in Algorithm 1 and is based on finding a saddle point of the modified Lagrangian for the optimization problem in (1). For a given sample ξ_i, we define this modified Lagrangian as follows:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{n} \Big[f_i(\mathbf{x}_i, \xi_i) + \sum_{j \in \mathcal{N}_i} \Big(\lambda_{ij} g_{ij}(\mathbf{x}_i, \mathbf{x}_j) - \frac{\delta\eta}{2} \lambda_{ij}^2 \Big) \Big] \tag{3}$$

On the L.H.S. of (3), $\mathbf{x}$ denotes the concatenation of all the model parameters $\{\mathbf{x}_i\}_{i=1}^n$, each of which is in $\mathbb{R}^d$, leading to $\mathbf{x} \in \mathbb{R}^{nd}$. For $i \in [n]$ and $j \in \mathcal{N}_i$, $\lambda_{ij} \geq 0$ denotes the Lagrangian multiplier associated with the constraint $g_{ij}(\mathbf{x}_i, \mathbf{x}_j) \leq 0$. Similarly, $\boldsymbol{\lambda}$ on the L.H.S. denotes the concatenation of all λ_{ij} for $i \in [n]$ and $j \in \mathcal{N}_i$, thus $\boldsymbol{\lambda} \in \mathbb{R}^m$, where m is twice the number of edges in the underlying undirected graph. The last term on the R.H.S. of (3) corresponds to a regularizer which mitigates the growth of the Lagrangian multiplier $\boldsymbol{\lambda}$ during the saddle-point algorithm updates. In this term, $\eta > 0$ corresponds to the learning rate of the algorithm and $\delta > 0$ is a control parameter.

To find the saddle point of the Lagrangian in (3), we utilize alternating gradient updates of the primal variables concatenated in $\mathbf{x}$, and the dual variables in $\boldsymbol{\lambda}$. For any $i \in [n]$, the gradient of the modified Lagrangian with respect to (w.r.t.) the model parameter $\mathbf{x}_i$ is given by:

$$\nabla_{\mathbf{x}_i} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{j \in \mathcal{N}_i} \left[\lambda_{ij} \nabla_{\mathbf{x}_i} g_{ij}(\mathbf{x}_i, \mathbf{x}_j) + \lambda_{ji} \nabla_{\mathbf{x}_i} g_{ij}(\mathbf{x}_j, \mathbf{x}_i) \right] + \nabla_{\mathbf{x}_i} f_i(\mathbf{x}_i, \xi_i) \tag{4}$$

The gradient w.r.t. the Lagrangian multiplier λ_{ij} for $i \in [n]$, $j \in \mathcal{N}_i$ is similarly given by:

$$\frac{\partial}{\partial \lambda_{ij}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = g_{ij}(\mathbf{x}_i, \mathbf{x}_j) - \delta\eta\lambda_{ij} \tag{5}$$

The stochastic algorithm developed for updating the primal and dual variables via equations (4) and (5) is presented in Algorithm 1, which is described below.

Algorithm 1 Compressed Decentralized Optimization with Sample Feedback

Initialize: Random raw parameters $\widetilde{\mathbf{x}}_i^{(1)} \in \mathcal{X}$, $\lambda_{ij}^{(1)} = 0$ for each $i \in [n], j \in \mathcal{N}_i$, $\hat{\mathbf{x}}_i^{(0)} = \mathbf{0}$ for each $i \in [n]$, number of iterations T, learning rate η, parameter $\delta > 0$. (Communicate in the first iteration without compression to ensure that $\widetilde{\mathbf{x}}^{(1)} = \hat{\mathbf{x}}^{(1)}$)

1: **for** $t = 1$ **to** T in parallel for $i \in [n]$ **do**
2: Compute $\mathbf{q}_i^{(t)} = \mathcal{C}(\widetilde{\mathbf{x}}_i^{(t)} - \hat{\mathbf{x}}_i^{(t-1)})$
3: **for** nodes $k \in \mathcal{N}_i \cup \{i\}$ **do**
4: Send $\mathbf{q}_i^{(t)}$ and receive $\mathbf{q}_k^{(t)}$
5: Update $\hat{\mathbf{x}}_k^{(t)} = \hat{\mathbf{x}}_k^{(t-1)} + \mathbf{q}_k^{(t)}$
6: Compute $\mathbf{x}_k^{(t)} = \Pi_{\mathcal{X}}(\hat{\mathbf{x}}_k^{(t)})$
7: **end for**
8: Update running average for local parameter: $\overline{\mathbf{x}}_{i,avg}^{(t)} = \frac{1}{t}\mathbf{x}_i^{(t)} + \frac{t-1}{t}\overline{\mathbf{x}}_{i,avg}^{(t-1)}$
9: Sample $\boldsymbol{\xi}_i^{(t)} \sim \mathcal{P}_i$ and compute $\nabla_{\mathbf{x}_i} f_i(\mathbf{x}_i^{(t)}, \boldsymbol{\xi}_i^{(t)})$
10: For all $j \in \mathcal{N}_i$ compute $\nabla_{\mathbf{x}_i} g_{ij}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)})$
11: Update the primal variable by gradient descent:

$$\widetilde{\mathbf{x}}_i^{(t+1)} = \Pi_{\mathcal{X}} \Big(\widetilde{\mathbf{x}}_i^{(t)} - \eta \nabla_{\mathbf{x}_i} f_i(\mathbf{x}_i^{(t)}, \boldsymbol{\xi}_i^{(t)}) - 2\eta \sum_{j \in \mathcal{N}_i} \lambda_{ij}^{(t)} \nabla_{\mathbf{x}_i} g_{ij}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}) \Big)$$

12: For $j \in \mathcal{N}_i$, update the dual variables through gradient ascent:

$$\lambda_{ij}^{(t+1)} = \Big[\lambda_{ij}^{(t)} + \eta \Big(g_{ij}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}) - \delta\eta\lambda_{ij}^{(t)} \Big) \Big]^+$$

13: **end for**

Output: Time averaged parameters $\overline{\mathbf{x}}_{i,avg}^{(T)}$ for all $i \in [n]$.

Our proposed scheme in Algorithm 1 is a stochastic saddle-point algorithm to minimize the objective in (1) by finding a saddle point of the modified Lagrangian in (3) in a communication efficient manner. Each node is allowed to exchange with its neighboring nodes only compressed parameters, via the compression operator in (2). To realize exchange of compressed parameters between workers, for node $i \in [n]$ and its associated *raw* parameter $\widetilde{\mathbf{x}}_i$, all nodes $j \in \mathcal{N}_i$ maintain an estimate $\hat{\mathbf{x}}_i$ of $\widetilde{\mathbf{x}}_i$,

so, each node $i \in [n]$ has access to $\hat{\mathbf{x}}_j$ for all $j \in \mathcal{N}_i$. The parameter $\widetilde{\mathbf{x}}_i$ is called raw as it corresponds to the model parameter before any compression in our algorithm. We refer to $\hat{\mathbf{x}}_i$ as the *copy* parameter of the node i.

We first initialize the regularization parameter δ (see Theorem 1 for definition) and learning rate η. We initialize the parameter copies of all the nodes as $\hat{\mathbf{x}}_i = \mathbf{0}$ for all $i \in [n]$ and allow each node to communicate with its neighbors in the first round without any compression. This is to ensure that $\widetilde{\mathbf{x}}_i^{(1)} = \hat{\mathbf{x}}_i^{(1)}$ for all the nodes (this is a requirement to control the error encountered via compression, c.f. Lemma 2). At any time step $t \in [T]$ of the algorithm, node i first computes the compressed update to its copy parameter, given by $\mathbf{q}_i^{(t)}$ (line 2) and then sends and receives these copy parameter updates from its neighbor nodes in $\mathcal{N}_i$ (line 3). Importantly, these copy parameter updates are compressed using the operator in (2), and thus the communication is efficient. After receiving the copy updates from its neighbors, each node updates the locally available copy parameters of its neighbors and its own copy parameters (line 5) and ensures that these lie in the set $\mathcal{X}$ by taking a projection [1] to form the local node parameter $\mathbf{x}_i^{(t)}$ (line 6). As the node i has access to the updated copy parameters of its neighbors, it also has access to $\mathbf{x}_j^{(t)}$ for all $j \in \mathcal{N}_i$. With the local node parameter evaluated, the node can update its running average of parameters (line 8).

For the stochastic saddle-point update with sample feedback, at time t , the node $i \in [n]$ can sample a data point $\boldsymbol{\xi}_i^{(t)}$ and evaluate the gradient using the previously computed node parameter $\mathbf{x}_i^{(t)}$ (line 9). Since the node also has access to the parameters $\mathbf{x}_j^{(t)}$ for neighbors $j \in \mathcal{N}_i$, it can compute the gradient w.r.t. the pairwise constraint function g_{ij} evaluated at $\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}$ (line 10). Thus, the node can evaluate the gradient of the modified Lagrangian w.r.t. the primal local node parameters as in (4) and take a gradient descent step to update the raw node parameter $\widetilde{\mathbf{x}}_i^{(t)}$. Similarly, the dual variables $\lambda_{ij}^{(t)}$ are also updated via a gradient ascent step (line 12) following (5) and then projected on the positive real space.

Symmetry of dual updates: Note that the derived expression for the gradient $\nabla_{\mathbf{x}_i}\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$, consists of the dual parameters λ_{ij} and λ_{ji}. Meanwhile, the update in line 11 of Algorithm 1 considers these parameters to be the same for all time $t \in [T]$. We describe the reasoning behind this update in the following induction argument. The dual variables are initialized to 0, that is, $\lambda_{ij}^{(1)} = 0$ for all $i \in [n]$ and $j \in \mathcal{N}_i$. Thus for any connected nodes i, j, for $t = 1$, the condition $\lambda_{ij}^{(t)} = \lambda_{ji}^{(t)}$ holds. Next, we assume that for any arbitrary $\tau \in [T]$, $\tau \neq 1$, it is the case that $\lambda_{ij}^{(\tau)} = \lambda_{ji}^{(\tau)}$. Thus for the time step $t = \tau + 1$, by the update given in line 12 of Algorithm 1, we have:

$$\begin{aligned}\lambda_{ij}^{(\tau+1)} &= \left[\lambda_{ij}^{(\tau)} + \eta\left(g_{ij}(\mathbf{x}_i^{(\tau)}, \mathbf{x}_j^{(\tau)}) - \delta\eta\lambda_{ij}^{(\tau)}\right)\right]^+ \\ &\overset{(a)}{=} \left[\lambda_{ji}^{(\tau)} + \eta\left(g_{ji}(\mathbf{x}_j^{(\tau)}, \mathbf{x}_i^{(\tau)}) - \delta\eta\lambda_{ji}^{(\tau)}\right)\right]^+ \\ &= \lambda_{ji}^{(\tau+1)}\end{aligned}$$

where (a) follows from the fact that $\lambda_{ij}^{(\tau)} = \lambda_{ji}^{(\tau)}$ and the symmetry of the pairwise constraints g_{ij} for connected nodes i, j. Thus, as the induction step holds for arbitrary $\tau \in [T]$ and for the base case $t = 1$, it follows that $\lambda_{ij}^{(t)} = \lambda_{ji}^{(t)}$ for all $t \in [T]$ for all $i \in [n], j \in \mathcal{N}_i$.

Justification for raw parameter updates: Note that in the steps given in lines (9-11) in Algorithm 1, the gradients are evaluated at the node parameters $\{\mathbf{x}_i^{(t)}\}_{i=1}^n$, while the updates are made to the raw parameters $\{\widetilde{\mathbf{x}}_i^{(t)}\}_{i=1}^n$ via gradient descent. The reason for this is that in our scheme, the raw parameters effectively play the role of a *virtual* parameter, which mimic SGD-like updates (c.f. line 11), with the gradients evaluated at a different (perturbed) parameter. The notion of such virtual parameters to analyze convergence has been promising lately in stochastic optimization within the perturbed iterate analysis framework, see [23,31,46,47]. The key idea to analyze convergence in such settings is to control the difference of the iterates $\|\mathbf{x}_i^{(t)} - \widetilde{\mathbf{x}}_i^{(t)}\|_2$ for all $i \in [n]$. Controlling this difference is one key contribution of our work, c.f. Lemma 2.

3.2 Main Result: Sample Feedback

We now present our theoretical result on the convergence rate of Algorithm 1 for decentralized optimization for the case with sample feedback. We first present and discuss the set of assumptions our result is based on.

A. 1. The set of admissible model parameters $\mathcal{X}$, is closed, convex and bounded, i.e., there exists a constant $R > 0$ such that $\|\widetilde{\mathbf{x}}\|_2 \leq \frac{R}{\sqrt{n}}$, for all $\widetilde{\mathbf{x}} \in \mathcal{X}$.

A. 2. For any $i \in [n]$, the local objective $f_i(\mathbf{x}_i, \boldsymbol{\xi}_i)$ is convex in $\mathbf{x}_i$ for any $\boldsymbol{\xi}_i \in \Xi_i$. The pairwise constraint function $g_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ is (jointly) convex in $\mathbf{x}_i$ and $\mathbf{x}_j$, for any pair $i \in [n], j \in \mathcal{N}_i$.

A. 3. For $i \in [n]$ and $\mathbf{x}_i \in \mathcal{X}$, $\exists G_i > 0$ such that:

$$\mathbb{E}_{\boldsymbol{\xi}_i \sim \mathcal{P}_i}\left[\|\nabla_{\mathbf{x}_i} f_i(\mathbf{x}_i, \boldsymbol{\xi}_i)\|_2^2\right] \leq G_i^2. \tag{6}$$

[1] It can be checked that the computational complexities for projection of all the primal node parameters and the dual parameters are $\mathcal{O}(nd)$ and $\mathcal{O}(m)$, respectively, per iteration.

To simplify the notation, we also define $G := \sqrt{\sum_{i=1}^{n} G_i^2}$. Additionally, for any $i \in [n], j \in \mathcal{N}_i$, we assume that there exists a constant $G_{ij} > 0$ such that $\forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$:

$$\left\| \left[\nabla_{\mathbf{x}_i} g_{ij}(\mathbf{x}_i, \mathbf{x}_j)^\mathsf{T}, \nabla_{\mathbf{x}_j} g_{ij}(\mathbf{x}_i, \mathbf{x}_j)^\mathsf{T} \right]^\mathsf{T} \right\|_2 \leq G_{ij}. \quad (7)$$

We define $\tilde{G} := \max_{i \in [n], j \in \mathcal{N}_i} G_{ij}$.

A. 4. For any $i \in [n], j \in \mathcal{N}_i$, the pairwise constraint function g_{ij} is bounded. That is, there exists a constant $C_{ij} > 0$ such that $|g_{ij}(\mathbf{x}_i, \mathbf{x}_j)| \leq C_{ij}$, $\forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$. We define $C^2 := \sqrt{\sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} C_{ij}^2}$.

Assumptions A.1-A.4 are frequently used in convergence rate analysis of convex optimization algorithms, even without compression. The assumption on a bounded parameter space $\mathcal{X}$ and the bounded constraint functions have been made earlier in [32, 48]. The assumption on boundedness of the gradient of the objectives (Equation (6)) has also been made earlier in [1, 32, 48] and boundedness of gradients of the constraint functions (Equation (7)) have been assumed in [32, 49, 50] [2] .

With these assumptions in place, we now present our main theoretical result in Theorem 1 below for the convergence rate of Algorithm 1. The result is stated in terms of the stacked vector $\mathbf{x}$, which corresponds to the concatenation of the parameters $\{\mathbf{x}_i\}_{i=1}^n$, and thus is $n \times d$ dimensional. The vector $\mathbf{x}^*$ represents the stacked optimal parameters which is the solution of the optimization problem (1). The proof details for Theorem 1 are presented in Section 3.3.

Theorem 1. Consider running Algorithm 1 for T iterations with fixed step size $\eta = \frac{a}{\sqrt{T}}$ for positive constant a and regularization parameter $\delta = \frac{1 - \sqrt{1 - \frac{64\eta^2(1+m)\tilde{G}^2}{\omega^2}}}{4\eta^2}$ where $\omega \in (0, 1)$ is the compression factor. Then, under assumptions A.1 - A.4, for $T \geq \frac{64a^2(1+m)\tilde{G}^2}{\omega^2}$, the expected value of F evaluated at the stacked time-averaged vector $\overline{\mathbf{x}}_{avg}^{(T)} := \frac{1}{T} \sum_{t=1}^{T} \mathbf{x}^{(t)}$ satisfies:

$$\mathbb{E}[F(\overline{\mathbf{x}}_{avg}^{(T)})] - F(\mathbf{x}^*) \leq \frac{2R^2}{a\sqrt{T}} + \frac{a}{\sqrt{T}} \left(\frac{4}{\omega^2}(1+m)G^2 + 2C^2 \right) \quad (8)$$

[2] Assumption A.3 for compressed decentralized optimization has been relaxed in one of our previous works [46]. The arguments for relaxing this assumption can similarly be extended to the analysis in this paper, a technicality which we omit in interest of keeping the analysis relatively cleaner, and to focus on the main novelty of analyzing compressed communication in the pairwise multi-task setting.

For $i \in [n]$, $j \in \mathcal{N}_i$, the constraint function g_{ij} satisfies:

$$\mathbb{E}\left[g_{ij}(\overline{\mathbf{x}}_{i,avg}^{(T)}, \overline{\mathbf{x}}_{j,avg}^{(T)}) \right] \leq \frac{1}{T^{\frac{1}{4}}} \left(\sqrt{\frac{8GR}{a}} + \sqrt{8\delta a G R} \right) + \frac{1}{\sqrt{T}} \sqrt{2 \left(2R^2\delta + \frac{4}{\omega^2}(1+m)G^2 + C^2 \right)} + \frac{1}{\sqrt{T}} \sqrt{2\delta a^2 \left(\frac{4}{\omega^2}(1+m)G^2 + C^2 \right)} + \frac{2R}{a\sqrt{T}} \quad (9)$$

where the d-dimensional vector $\overline{\mathbf{x}}_{k,avg}^{(T)}$ denotes the time averaged parameter for node $k \in [n]$ in $\overline{\mathbf{x}}_{avg}^{(T)}$.

Theorem 1 establishes that for any given compression requirement $\omega \in (0, 1)$, the sub-optimality of the objective, $\mathbb{E}[F(\overline{\mathbf{x}}^{(T)})] - F(\mathbf{x}^*)$, is $\mathcal{O}\left(\frac{1}{T^{1/2}}\right)$, and the expected constraint violation $\mathbb{E}[g_{ij}(\overline{\mathbf{x}}_i^{(T)}, \overline{\mathbf{x}}_j^{(T)})]$ for any connected node pair (i, j) is $\mathcal{O}\left(\frac{1}{T^{1/4}}\right)$. Thus, the difference between the attained objective and the global minimum of (1), as well as the constraint violations can be made arbitrarily small by increasing the number of iterations the algorithm is run for.

3.3 Convergence analysis

We first introduce a compact vector notation which we will use throughout the proof. Consider the stacked (concatenated) vector of the node parameter vectors $\{\mathbf{x}_i\}_{i=1}^n$ which we denote by $\mathbf{x}$, and thus is nd-dimensional. Similarly, we define the vector $\boldsymbol{\lambda}$ of size m which stacks together the dual variables λ_{ij} for $i \in [n]$ and $j \in \mathcal{N}_i$. The vector $\mathbf{g}(\mathbf{x})$ represents the the stacked vector of constraint values $g_{ij}(\mathbf{x}_j, \mathbf{x}_j)$, and is also m-dimensional. Finally, $\boldsymbol{\xi}$ denotes the concatenated vector of samples across the nodes. The projection $\Pi_{\mathcal{X}^n}(\mathbf{x})$ refers to projection of $\mathbf{x}$ on the space $\mathcal{X}^n$ where each individual node parameter comprising $\mathbf{x}$ is projected onto $\mathcal{X}$. Under this compact notation, the modified Lagrangian presented in (3) can be re-written as:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}, \boldsymbol{\xi}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}) - \frac{\delta\eta}{2} \|\boldsymbol{\lambda}\|^2 \quad (10)$$

We now present a few auxiliary results which we use through the course of the proof. Some of these can be derived from the assumptions made in A.1-A.4.

Fact 1. *Suppose $\mathcal{A} \subset \mathbb{R}^l$ is closed and convex. Then, for any $\mathbf{y} \in \mathbb{R}^l$ and $\mathbf{x} \in \mathcal{A}$, we have:*

$$\|\mathbf{x} - \Pi_{\mathcal{A}}(\mathbf{y})\|_2 \leq \|\mathbf{x} - \mathbf{y}\|_2$$

where $\Pi_{\mathcal{A}}(\mathbf{y})$ denotes the projection of $\mathbf{y}$ on the set $\mathcal{A}$.

Fact 2. *(Bound on gradients of the Lagrangian) Consider the Lagrangian function over the primal and dual variables defined in* (10)*. We have the following bounds:*

(a) $\mathbb{E}\|\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\mathbf{x}^{(t)},\boldsymbol{\lambda}^{(t)})\|^2 \leq 2C^2 + 2\delta^2\eta^2\mathbb{E}\|\boldsymbol{\lambda}^{(t)}\|^2$

(b) $\mathbb{E}\|\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^{(t)},\boldsymbol{\lambda}^{(t)})\|^2 \leq (1+m)\left(G^2 + \widetilde{G}^2\mathbb{E}\|\boldsymbol{\lambda}\|^2\right)$

where $C^2, \tilde{G}$ *and* G *are as defined in Assumptions A.3 and A.4. Proof of this fact can be found in [59].*

Fact 3. *For all* $\mathbf{x} \in \mathcal{X}^n$*, we have:*

$$\mathbb{E}[F(\mathbf{x})] - F(\mathbf{x}^*) > -4GR$$

where $\mathbf{x}^*$ *is an optimal solution of* (1)*, and* R, G *are as defined in Assumptions A.1 and A.4, respectively. We provide a proof for Fact 3 in [59].*

3.3.1 Proof of Theorem 1

We first consider the following lemma which establishes a relationship between the Lagrangian function and the primal, dual variables in Algorithm 1. The proof for the lemma, provided in the [59], relies on considering the update steps of the primal and dual variables in Algorithm 1 and invoking convexity/concavity arguments for the Lagrangian function.

Lemma 1. Consider the update steps in Algorithm 1 with learning rate η and parameter $\delta \geq 0$. Under assumptions A.1-A.4, for $\mathbf{x} \in \mathcal{X}^n$ and $\boldsymbol{\lambda} \in \mathbb{R}^m$ with $\boldsymbol{\lambda} \succeq \mathbf{0}$, the summation of the Lagrangian function satisfies:

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}\left(\mathcal{L}(\mathbf{x}^{(t)},\boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x},\boldsymbol{\lambda}^{(t)})\right) &\leq \frac{1}{2\eta}\left(\|\boldsymbol{\lambda}\|^2 + 4R^2\right)\\ &+ \eta T\left((1+m)G^2 + C^2\right) + \frac{1}{2\eta}\sum_{t=1}^{T}\mathbb{E}\|\mathbf{x}^{(t)} - \tilde{\mathbf{x}}^{(t)}\|^2\\ &+ \eta\left((1+m)\tilde{G}^2 + \delta^2\eta^2\right)\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{\lambda}^{(t)}\|^2]\end{aligned}$$

where $G, C, \tilde{G}$, R are defined in assumptions A.1-A.4.

Using the definition of Lagrangian from (10) and $\mathbb{E}[f(\mathbf{x}^{(t)},\boldsymbol{\xi}^{(t)})] = F(\mathbf{x}^{(t)})$, the L.H.S. of the result in Lemma 1 can also be written as following for any $\boldsymbol{\lambda} \succeq \mathbf{0}$:

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T}\left(\mathcal{L}(\mathbf{x}^{(t)},\boldsymbol{\lambda}) - \mathcal{L}(\mathbf{x},\boldsymbol{\lambda}^{(t)})\right)\right]\\ &= \sum_{t=1}^{T}(\mathbb{E}[F(\mathbf{x}^{(t)})] - F(\mathbf{x}^*)) + \left\langle\boldsymbol{\lambda}, \sum_{t=1}^{T}\mathbb{E}[\mathbf{g}(\mathbf{x}^{(t)})]\right\rangle - \frac{\delta\eta T}{2}\|\boldsymbol{\lambda}\|^2\\ &\quad - \mathbb{E}\left[\sum_{t=1}^{T}\langle\boldsymbol{\lambda}^{(t)}, \mathbf{g}(\mathbf{x}^*)\rangle\right] + \frac{\delta\eta}{2}\mathbb{E}\left[\sum_{t=1}^{T}\|\boldsymbol{\lambda}^{(t)}\|^2\right]\end{aligned}$$

Rearranging the terms and employing the bound from Lemma 1, for any $\boldsymbol{\lambda} \succeq \mathbf{0}$, we thus have:

$$\begin{aligned}&\sum_{t=1}^{T}\left(\mathbb{E}[F(\mathbf{x}^{(t)})] - F(\mathbf{x}^*)\right) + \left\langle\boldsymbol{\lambda}, \sum_{t=1}^{T}\mathbb{E}[\mathbf{g}(\mathbf{x}^{(t)})]\right\rangle\\ &\quad - \frac{\delta\eta T}{2}\|\boldsymbol{\lambda}\|^2 - \mathbb{E}\left[\sum_{t=1}^{T}\langle\boldsymbol{\lambda}^{(t)}, \mathbf{g}(\mathbf{x}^*)\rangle\right]\\ &\leq \frac{1}{2\eta}\left(\|\boldsymbol{\lambda}\|^2 + 4R^2\right) + \frac{1}{2\eta}\sum_{t=1}^{T}\mathbb{E}\|\mathbf{e}^{(t)}\|^2 + \eta T\left((1+m)G^2 + C^2\right)\\ &\quad + \eta\left((1+m)\tilde{G}^2 + \delta^2\eta^2 - \frac{\delta}{2}\right)\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{\lambda}^{(t)}\|^2] \quad (11)\end{aligned}$$

where we have defined $\mathbb{E}\|\mathbf{e}^{(t)}\|^2 := \mathbb{E}\|\widetilde{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}\|^2$ on the R.H.S. of (11). This term relates to the error between the copies of the parameters at time t (denoted by $\widetilde{\mathbf{x}}^{(t)}$) and the true parameters of the nodes (given by $\mathbf{x}^{(t)}$). We provide a bound for this term in Lemma 2 stated below, the proof of which is provided in the arXiv version of the paper [59].

Lemma 2. For the update steps in Algorithm 1, the norm of expected error $\mathbb{E}\|\mathbf{e}^{(t)}\|$ for $t \in [T]$ is bounded as:

$$\begin{aligned}&\mathbb{E}\|\mathbf{e}^{(t)}\|^2 \leq\\ &\frac{2\eta^2}{\omega}\sum_{k=0}^{t-2}\left(1 - \frac{\omega}{2}\right)^k\mathbb{E}\|\nabla_{\mathbf{x}}\mathcal{L}_{t-1-k}(\mathbf{x}^{(t-1-k)},\boldsymbol{\lambda}^{(t-1-k)})\|^2\end{aligned}$$

Plugging the bound for $\mathbb{E}\|\mathbf{e}^{(t)}\|^2$ from Lemma 2 into (11):

$$\begin{aligned}&\sum_{t=1}^{T}\left(\mathbb{E}[F(\mathbf{x}^{(t)})] - F(\mathbf{x}^*)\right) + \left\langle\boldsymbol{\lambda}, \sum_{t=1}^{T}\mathbb{E}[\mathbf{g}(\mathbf{x}^{(t)})]\right\rangle\\ &\quad - \frac{\delta\eta T}{2}\|\boldsymbol{\lambda}\|^2 - \mathbb{E}\left[\sum_{t=1}^{T}\langle\boldsymbol{\lambda}^{(t)}, \mathbf{g}(\mathbf{x}^*)\rangle\right]\\ &\leq \frac{1}{2\eta}\left(\|\boldsymbol{\lambda}\|^2 + 4R^2\right) + \eta T\left((1+m)G^2 + C^2\right)\\ &+ \frac{\eta}{\omega}\sum_{t=1}^{T}\sum_{k=0}^{t-2}\left(1 - \frac{\omega}{2}\right)^k\mathbb{E}\|\nabla_{\mathbf{x}}\mathcal{L}_{t-1-k}(\mathbf{x}^{(t-1-k)},\boldsymbol{\lambda}^{(t-1-k)})\|^2\\ &\quad + \eta\left((1+m)\tilde{G}^2 + \delta^2\eta^2 - \frac{\delta}{2}\right)\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{\lambda}^{(t)}\|^2] \quad (12)\\ &= \frac{1}{2\eta}\left(\|\boldsymbol{\lambda}\|^2 + 4R^2\right) + \eta T\left((1+m)G^2 + C^2\right)\\ &\quad + \eta\left((1+m)\tilde{G}^2 + \delta^2\eta^2 - \frac{\delta}{2}\right)\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{\lambda}^{(t)}\|^2]\end{aligned}$$

$$+\frac{\eta}{\omega}\sum_{k=1}^{T-1}\sum_{t=k+1}^{T}\left(1-\frac{\omega}{2}\right)^{(t-1-k)}\mathbb{E}\|\nabla_{\mathbf{x}}\mathcal{L}_k(\mathbf{x}^{(k)},\boldsymbol{\lambda}^{(k)})\|^2$$

where the equality follows from rewriting the double-sum of the second term. Using $\sum_{t=k+1}^{T}\left(1-\frac{\omega}{2}\right)^{(t-1-k)} \leq \sum_{t=0}^{\infty}\left(1-\frac{\omega}{2}\right)^{(t)} = \frac{2}{\omega}$, we get:

$$\begin{aligned}
&\sum_{t=1}^{T}(\mathbb{E}[F(\mathbf{x}^{(t)})]-F(\mathbf{x}^*)) + \left\langle \boldsymbol{\lambda}, \sum_{t=1}^{T}\mathbb{E}[\mathbf{g}(\mathbf{x}^{(t)})]\right\rangle \\
&\quad - \frac{\delta\eta T}{2}\|\boldsymbol{\lambda}\|^2 - \mathbb{E}\left[\sum_{t=1}^{T}\langle\boldsymbol{\lambda}^{(t)},\mathbf{g}(\mathbf{x}^*)\rangle\right] \\
&\leq \frac{1}{2\eta}\left(\|\boldsymbol{\lambda}\|^2+4R^2\right)+\eta T\left((1+m)G^2+C^2\right) \\
&\qquad + \eta\left((1+m)\tilde{G}^2+\delta^2\eta^2-\frac{\delta}{2}\right)\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{\lambda}^{(t)}\|^2] \\
&\qquad + \frac{2\eta}{\omega^2}\sum_{t=1}^{T-1}\mathbb{E}\|\nabla_{\mathbf{x}}\mathcal{L}_t(\mathbf{x}^{(t)},\boldsymbol{\lambda}^{(t)})\|^2 \qquad (13)
\end{aligned}$$

Using the bound from (b) in Fact 2 for the last term in above, and noting that $\frac{2}{\omega^2} > 1$ gives us:

$$\begin{aligned}
&\sum_{t=1}^{T}(\mathbb{E}[F(\mathbf{x}^{(t)})]-F(\mathbf{x}^*)) + \left\langle \boldsymbol{\lambda}, \sum_{t=1}^{T}\mathbb{E}[\mathbf{g}(\mathbf{x}^{(t)})]\right\rangle \\
&\quad - \frac{\delta\eta T}{2}\|\boldsymbol{\lambda}\|^2 - \mathbb{E}\left[\sum_{t=1}^{T}\langle\boldsymbol{\lambda}^{(t)},\mathbf{g}(\mathbf{x}^*)\rangle\right] \\
&\leq \frac{1}{2\eta}\left(\|\boldsymbol{\lambda}\|^2+4R^2\right)+\eta T\left(\frac{4}{\omega^2}(1+m)G^2+C^2\right) \\
&+ \eta\left(\frac{4}{\omega^2}(1+m)\tilde{G}^2+\delta^2\eta^2-\frac{\delta}{2}\right)\sum_{t=1}^{T}\mathbb{E}[\|\boldsymbol{\lambda}^{(t)}\|^2] \qquad (14)
\end{aligned}$$

We now focus on the last term in the above equation, which has a coefficient of $\left(\frac{4}{\omega^2}(1+m)\tilde{G}^2+\delta^2\eta^2-\frac{\delta}{2}\right)$. To get rid of the last term in the upper bound, we choose the value of δ such that this coefficient is negative. It can be easily checked that the following value of δ satisfies this requirement:

$$\delta = \frac{1-\sqrt{1-\frac{64\eta^2(1+m)\tilde{G}^2}{\omega^2}}}{4\eta^2}$$

Note that we require running the algorithm for $T \geq \frac{64a^2(1+m)\tilde{G}^2}{\omega^2}$ for the choice $\eta = \frac{a}{\sqrt{T}}$. For $T \to \infty$ (i.e., $\eta \to 0$), it can be verified that the value of δ converges to a positive constant. Using the above value of δ, the fact $\mathbb{E}\left[\sum_{t=1}^{T}\langle\boldsymbol{\lambda}^{(t)},\mathbf{g}(\mathbf{x}^*)\rangle\right] \leq 0$ since $\boldsymbol{\lambda}^{(t)} \succeq \mathbf{0}$ for $t \in [T]$ and $\mathbf{g}(\mathbf{x}^*) \preceq \mathbf{0}$ and rearranging the terms, we get:

$$\begin{aligned}
&\sum_{t=1}^{T}\left(\mathbb{E}[F(\mathbf{x}^{(t)})]-F(\mathbf{x}^*)\right) + \left\langle \boldsymbol{\lambda}, \sum_{t=1}^{T}\mathbb{E}[\mathbf{g}(\mathbf{x}^{(t)})]\right\rangle \\
&\quad - \left(\frac{\delta\eta T}{2}+\frac{1}{2\eta}\right)\|\boldsymbol{\lambda}\|^2 \\
&\leq \frac{2R^2}{\eta}+\eta T\left(\frac{4}{\omega^2}(1+m)G^2+C^2\right) \qquad (15)
\end{aligned}$$

Recall that $\boldsymbol{\lambda}$ can be any non-negative vector. We set it as $\boldsymbol{\lambda} = \frac{\left[\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{g}(\mathbf{x}^{(t)})\right]\right]^+}{\delta\eta T+\frac{1}{\eta}}$. Plugging this in (15) yields:

$$\begin{aligned}
&\sum_{t=1}^{T}\left(\mathbb{E}[F(\mathbf{x}^{(t)})]-F(\mathbf{x}^*)\right) \\
&\qquad + \sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}\frac{\left(\left[\mathbb{E}\left[\sum_{t=1}^{T}g_{ij}(\mathbf{x}_i^{(t)},\mathbf{x}_j^{(t)})\right]\right]^+\right)^2}{2\left(\delta\eta T+\frac{1}{\eta}\right)} \\
&\leq \frac{2R^2}{\eta}+\eta T\left(\frac{4}{\omega^2}(1+m)G^2+C^2\right) \qquad (16)
\end{aligned}$$

Dividing both sides of (16) by T and noting that the second term on the L.H.S. of (16) is positive, we can bound the time-average sub-optimality of F as:

$$\sum_{t=1}^{T}\frac{\left(\mathbb{E}[F(\mathbf{x}^{(t)})]-F(\mathbf{x}^*)\right)}{T} \leq \frac{2R^2}{\eta T}+\eta\left(\frac{4}{\omega^2}(1+m)G^2+C^2\right)$$

Using the convexity of F and setting $\eta = \frac{a}{\sqrt{T}}$ for some positive constant a, concludes the proof of the convergence rate for the objective sub-optimality given in (8) in Theorem 1. We now prove our result for the pairwise constraint functions. From Fact 3, $\forall \mathbf{x} \in \mathcal{X}^n$, we have $\mathbb{E}[F(\mathbf{x})]-F(\mathbf{x}^*) > -4GR$. Using this inequality in (16):

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}\left(\left[\mathbb{E}\left[\sum_{t=1}^{T}g_{ij}(\mathbf{x}_i^{(t)},\mathbf{x}_j^{(t)})\right]\right]^+\right)^2 \\
&\leq \frac{4R^2}{\eta^2}+T\left(4R^2\delta+\frac{8}{\omega^2}(1+m)G^2+2C^2+\frac{8GR}{\eta}\right) \\
&\qquad + T^2\left(2\delta\eta^2\left(\frac{4}{\omega^2}(1+m)G^2+C^2\right)+8\delta\eta GR\right)
\end{aligned}$$

Note that the above bound also holds for a given $i \in [n]$ and $j \in \mathcal{N}_i$, that is, the R.H.S. of the above equation is also a bound for the term $\left(\left[\mathbb{E}\left[\sum_{t=1}^{T}g_{ij}(\mathbf{x}_i^{(t)},\mathbf{x}_j^{(t)})\right]\right]^+\right)^2$. Taking square root on both sides and using the fact that

$\sqrt{\sum_{i=1}^n p_i} \leq \sum_{i=1}^n \sqrt{p_i}$ for positive $p_1, \ldots, p_n$ yields:

$$\mathbb{E}\left[\sum_{t=1}^{T} g_{ij}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)})\right] \leq \frac{2R}{\eta} + \sqrt{2T}\sqrt{\left(2R^2\delta + \frac{4}{\omega^2}(1+m)G^2 + C^2 + \frac{4GR}{\eta}\right)} + \sqrt{2}T\sqrt{\left(\delta\eta^2\left(\frac{4}{\omega^2}(1+m)G^2 + C^2\right) + 4\delta\eta GR\right)}$$

Dividing both sides of above by T, using the convexity of constraint function g_{ij} and substituting $\eta = \frac{a}{\sqrt{T}}$ concludes the proof of (9) in Theorem 1. □

4 Decentralized compressed optimization with Bandit feedback

In this section, we focus on the *bandit feedback* scenario where the nodes do not have direct access to samples drawn from their local data distributions. This could, for example, arise in situations where the samples are high dimensional and thus can be hard to observe or measure. For the model we work with in this paper, we now assume that the nodes instead can query the value of the local objective function $f_i(\mathbf{x}_i, \xi_i)$ for some particular choices of the parameter $\mathbf{x}_i$. We first formally define the objective query process for the nodes and then describe how this model can be used to develop a stochastic gradient method for optimizing the overall objective (1).

Let $\mathbb{S} := \{\mathbf{u} \in \mathbb{R}^d | \|\mathbf{u}\|_2 = 1\}$ and $\mathbb{B} := \{\mathbf{u} \in \mathbb{R}^d | \|\mathbf{u}\|_2 \leq 1\}$ be the unit sphere, ball in d-dimensions, respectively. For each node $i \in [n]$, and at any stage in the optimization process, we assume access to two local objective values $f_i(\mathbf{x}_i \pm \zeta \mathbf{u}_i, \xi_i)$ where $\mathbf{u}_i$ is sampled uniformly at random over the unit sphere $\mathbb{S}$ (independent of $\mathbf{x}_i$ or ξ_i), ζ is a small positive constant, and $\mathbf{x}_i$ is the local model parameter. To evaluate the gradient using these objective values, we make use of the following fact from [39]:

Fact 4. *Consider a function $\phi : \mathbb{R}^d \to \mathbb{R}$, and let $\zeta > 0$. Define $\tilde{\phi}(\mathbf{x}) = \mathbb{E}_{\mathbf{u}\sim\mathcal{U}(\mathbb{B})}[\phi(\mathbf{x} + \zeta\mathbf{u})]$ where $\mathcal{U}(\mathbb{B})$ denotes uniform distribution over the unit ball $\mathbb{B} \subset \mathbb{R}^d$. Then:*

(i) If ϕ is convex, then $\tilde{\phi}$ is also convex.
(ii) For any $\mathbf{x} \in \mathbb{R}^d$, $\nabla_{\mathbf{x}}\tilde{\phi}(\mathbf{x}) = \frac{d}{\zeta}\mathbb{E}_{\mathbf{u}\sim\mathcal{U}(\mathbb{S})}[\phi(\mathbf{x} + \zeta\mathbf{u})\mathbf{u}]$ where $\mathcal{U}(\mathbb{S})$ denotes the uniform distribution over the unit sphere $\mathbb{S} \subset \mathbb{R}^d$.

For the node $i \in [n]$, the above fact can be used to estimate the gradient of the local objective function using the values $f_i(\mathbf{x}_i \pm \zeta\mathbf{u}_i, \xi_i)$ where $\mathbf{u}_i \sim \mathcal{U}(\mathbb{S})$. For a given ξ_i, we define $\tilde{f}_i(\mathbf{x}_i, \xi_i) = \mathbb{E}_{\mathbf{v}_i\sim\mathcal{U}(\mathbb{B})}[f_i(\mathbf{x}_i + \zeta\mathbf{v}_i, \xi_i)]$. From the above fact, $\tilde{f}(\mathbf{x}_i, \xi_i)$ is convex in $\mathbf{x}_i$ for a given ξ_i.

Note that as stated, the parameter vector $\mathbf{x}_i \pm \zeta\mathbf{u}_i$ may not lie in the feasible set $\mathcal{X}$ for all range of values of ζ. Thus, we need some restriction on the range of values ζ can take. In the following, we make this argument precise. We first introduce an additional mild assumption on the topology of the set $\mathcal{X}$:

A.5. The set $\mathcal{X}$ has a non-empty interior, that is, $\exists \mathbf{y}_0 \in \mathcal{X}$, $r > 0$, s.t. $\mathcal{B}(\mathbf{y}_0, r) \subset \mathcal{X}$. Here, $\mathcal{B}(\mathbf{y}_0, r)$ denotes the open ball of radius r centered at $\mathbf{y}_0$, i.e., $\mathcal{B}(\mathbf{y}_0, r) = \{\mathbf{x} | \|\mathbf{x} - \mathbf{y}_0\|_2 \leq r\}$.

From the above assumption, by the convexity of $\mathcal{X}$, it can also be concluded that for any $\alpha \in (0, 1)$ and $\mathbf{x} \in \mathcal{X}$, we have $\mathcal{B}((1-\alpha)\mathbf{x} + \alpha\mathbf{y}_0, \alpha r) \subset \mathcal{X}$. We further define the set $\widetilde{\mathcal{X}} = \{(1 - \frac{\zeta}{r})\mathbf{x} + \frac{\zeta}{r}\mathbf{y}_0 | \mathbf{x} \in \mathcal{X}\}$. It can now be readily checked that if $\mathbf{x}_i \in \widetilde{\mathcal{X}}$ for the node i, then $\mathbf{x}_i \pm \zeta\mathbf{u}_i \in \mathcal{X}$, where $\mathbf{u}_i$ is any point on the unit sphere $\mathbb{S}$. Thus in the development of the algorithm below, we project the parameters onto the space $\widetilde{\mathcal{X}}$ to ensure that during the bandit feedback, the evaluated parameter $\mathbf{x}_i \pm \zeta\mathbf{u}_i$ for any node i lies in the space $\mathcal{X}$.

4.1 Algorithm: Bandit Feedback

We develop an algorithm for the bandit feedback scenario to find a saddle-point of the modified Lagrangian:

$$\tilde{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{n}\left[\tilde{f}_i(\mathbf{x}_i, \xi_i) + \sum_{j\in\mathcal{N}_i}\left(\lambda_{ij}g_{ij}(\mathbf{x}_i, \mathbf{x}_j) - \frac{\delta\eta}{2}\lambda_{ij}^2\right)\right] \tag{17}$$

The vector $\mathbf{x} \in \widetilde{\mathcal{X}}^n$ represents the stacked node parameters and $\boldsymbol{\lambda}$ represents the stacked dual variables. Here, the main difference from the modified Lagrangian in sample feedback case presented in (3) is that the objectives $\{f_i\}_{i=1}^n$ of the nodes are now replaced by the functions $\{\tilde{f}_i\}_{i=1}^n$. Importantly, the gradient of these functions can be computed via the result of Fact 4 which enables us to develop a primal-dual gradient algorithm to find the saddle point of (17). The gradient w.r.t. the primal variable $\mathbf{x}$ is given by:

$$\nabla_{\mathbf{x}_i}\tilde{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{j\in\mathcal{N}_i}\left[\lambda_{ij}\nabla_{\mathbf{x}_i}g_{ij}(\mathbf{x}_i, \mathbf{x}_j) + \lambda_{ji}\nabla_{\mathbf{x}_i}g_{ij}(\mathbf{x}_j, \mathbf{x}_i)\right] + \nabla_{\mathbf{x}_i}\tilde{f}_i(\mathbf{x}_i, \xi_i) \tag{18}$$

Using the result from Fact 4, for any $i \in [n]$ we have:

$$\nabla_{\mathbf{x}}\tilde{f}_i(\mathbf{x}_i, \xi_i) = \frac{d}{2\zeta}\mathbb{E}_{\mathbf{u}_i\sim\mathcal{U}(\mathbb{S})}[f(\mathbf{x}_i + \zeta\mathbf{u}_i, \xi_i) - f(\mathbf{x}_i - \zeta\mathbf{u}_i, \xi_i)]\mathbf{u}_i$$

As the node has access to the values of the local objective function in the bandit feedback scenario, the quantity $\frac{d}{2\zeta}[f(\mathbf{x}_i + \zeta\mathbf{u}_i, \xi_i) - f(\mathbf{x}_i - \zeta\mathbf{u}_i, \xi_i)]$ for a given $\mathbf{u}_i \sim \mathcal{U}(\mathbb{S})$,

$\mathbf{x}_i, \xi_i$, serves as an unbiased estimate of $\nabla_{\mathbf{x}} \tilde{f}_i(\mathbf{x}_i, \xi_i)$. We note that such an approximation for the gradient is common in the stochastic optimization literature, e.g. [54, 55]. In contrast to the uniform perturbation we consider in Fact 4, one can possibly use perturbations arising from distributions such as Gaussian, symmetric Bernoulli distributions as in [56, 58]. Using this, we can construct the following estimate for the primal gradient $\nabla_{\mathbf{x}_i} \tilde{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda})$:

$$\mathbf{p}_i^{(t)} := \frac{d}{2\xi}\left[f_i(\mathbf{x}_i^{(t)}+\zeta\mathbf{u}_i^{(t)}, \boldsymbol{\xi}_i^{(t)}) - f_i(\mathbf{x}_i^{(t)}-\zeta\mathbf{u}_i^{(t)}, \boldsymbol{\xi}_i^{(t)})\right]\mathbf{u}_i^{(t)} + 2\sum_{j\in\mathcal{N}_i}\lambda_{ij}^{(t)}\nabla_{\mathbf{x}_i} g_{ij}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}) \tag{19}$$

The gradient of the Lagrangian in (17) w.r.t. the dual parameter λ_{ij} for $i \in [n]$ and $j \in \mathcal{N}_i$ is the same as in the sample feedback scenario and is given in (4).

The development of Algorithm 2 is similar to that of Algorithm 1. The main difference is that we now find the saddle point of (17) via alternating primal and dual variable gradient updates given in equations (19) and (5) and project onto the space $\widetilde{\mathcal{X}}$ to ensure that the perturbed parameters lie in $\mathcal{X}$. As before, for a node $i \in [n]$, $\widetilde{\mathbf{x}}_i$ refers to its raw parameter, $\mathbf{x}_i$ as its local parameter, and $\hat{\mathbf{x}}_i$ is the copy parameter.

We initialize the raw parameters $\{\widetilde{\mathbf{x}}_i^{(1)}\}_{i=1}^n$ inside the set $\widetilde{\mathcal{X}}$. During the first round, we assume the communication without compression to ensure that $\widetilde{\mathbf{x}}_i^{(1)} = \hat{\mathbf{x}}_i^{(1)}$ for all $i \in [n]$. At time step $t \in [T]$, the node $i \in [n]$ computes and exchanges its copy parameters and constructs the local node parameter $\mathbf{x}_i^{(t)}$ for which we track the running average (lines 2-8). As samples from the underlying distribution $\mathcal{P}_i$ are not directly revealed to the node in case of bandit feedback; instead it queries the value of the local objective $f_i(., \xi_i)$ at parameters $\mathbf{x}_i^{(t)} + \zeta\mathbf{u}_i^{(t)}$ and $\mathbf{x}_i^{(t)} - \zeta\mathbf{u}_i^{(t)}$ where $\mathbf{u}_i^{(t)}$ is uniformly sampled over the d-dimensional unit sphere $\mathbb{S}$ (lines 9-10). These values are then used to construct an unbiased estimate of $\nabla_{\mathbf{x}_i}\tilde{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda})$ using (19), and then to update the raw parameter $\widetilde{\mathbf{x}}_i^{(t)}$ along with a projection operation back to the set $\widetilde{\mathcal{X}}$ (lines 11-13). Finally, the dual variables are also updated via gradient descent along with the projection to the positive real space to ensure feasibility (line 13). As in the case of sample feedback, the update of the dual steps in line 13 and the initialization $\lambda_{ij}^{(1)} = 0$ ensures that $\lambda_{ij}^{(t)} = \lambda_{ji}^{(t)}$ for all $t \in [T]$, and for all $i \in [n], j \in \mathcal{N}_i$.

4.2 Main Result: Bandit Feedback

We now present the convergence result rate for Algorithm 2 which optimizes (1) in the bandit feedback scenario. The proof details are provided in Section 4.3.

Algorithm 2 Compressed Decentralized Optimization with Bandit Feedback

Initialize: Random $\widetilde{\mathbf{x}}_i^{(1)} \in \widetilde{\mathcal{X}}$ individually for each $i \in [n]$ and $\lambda_{ij}^{(1)} = 0$ for each $j \in \mathcal{N}_i$. $\hat{\mathbf{x}}_i^{(0)} = \mathbf{0}$ for each $i \in [n]$, number of iterations T, learning rate η, parameters $\zeta, \delta > 0$.
(Communicate in the first iteration without compression to ensure that $\widetilde{\mathbf{x}}^{(1)} = \hat{\mathbf{x}}^{(1)}$.)

1: **for** $t = 1$ **to** T in parallel for $i \in [n]$ **do**
2: Compute $\mathbf{q}_i^{(t)} = \mathcal{C}(\widetilde{\mathbf{x}}_i^{(t)} - \hat{\mathbf{x}}_i^{(t-1)})$
3: **for** nodes $k \in \mathcal{N}_i \cup \{i\}$ **do**
4: Send $\mathbf{q}_i^{(t)}$ and receive $\mathbf{q}_k^{(t)}$
5: Update $\hat{\mathbf{x}}_k^{(t)} = \hat{\mathbf{x}}_k^{(t-1)} + \mathbf{q}_k^{(t)}$
6: Compute $\mathbf{x}_k^{(t)} = \Pi_{\widetilde{\mathcal{X}}}(\hat{\mathbf{x}}_k^{(t)})$
7: **end for**
8: Update running average for local parameter: $\overline{\mathbf{x}}_{i,avg}^{(t)} = \frac{1}{t}\mathbf{x}_i^{(t)} + \frac{t-1}{t}\overline{\mathbf{x}}_{i,avg}^{(t-1)}$
9: Sample $\mathbf{u}_i^{(t)} \sim \mathcal{U}(\mathbb{S})$
10: Query the two values: $f_i(\mathbf{x}_i^{(t)} \pm \zeta\mathbf{u}_i^{(t)}, \boldsymbol{\xi}_i^{(t)})$
11: Compute the Lagrangian primal gradient estimate:

$$\mathbf{p}_i^{(t)} := 2\sum_{j\in\mathcal{N}_i}\lambda_{ij}^{(t)}\nabla_{\mathbf{x}_i} g_{ij}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}) + \frac{d}{2\xi}\left[f_i(\mathbf{x}_i^{(t)}+\zeta\mathbf{u}_i^{(t)}, \boldsymbol{\xi}_i^{(t)}) - f_i(\mathbf{x}_i^{(t)}-\zeta\mathbf{u}_i^{(t)}, \boldsymbol{\xi}_i^{(t)})\right]\mathbf{u}_i^{(t)}$$

12: Update the primal variable via gradient descent:

$$\widetilde{\mathbf{x}}_i^{(t+1)} = \Pi_{\widetilde{\mathcal{X}}}\left(\widetilde{\mathbf{x}}_i^{(t)} - \eta\mathbf{p}_i^{(t)}\right)$$

13: For all $j \in \mathcal{N}_i$, update the dual variables via gradient ascent:

$$\lambda_{ij}^{(t+1)} = \left[\lambda_{ij}^{(t)} + \eta\left(g_{ij}(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}) - \delta\eta\lambda_{ij}^{(t)}\right)\right]^+$$

14: **end for**

Output: Time averaged parameters $\overline{\mathbf{x}}_{i,avg}^{(T)}$ for all $i \in [n]$.

Theorem 2. Consider running Algorithm 2 for T iterations with fixed step size $\eta = \frac{a}{\sqrt{T}}$ for positive constant a, with perturbation constant $\zeta = \frac{1}{T}$, and regularization parameter $\delta = \frac{1-\sqrt{1-\frac{256\eta^2(1+m)\tilde{G}^2}{\omega^2}}}{4\eta^2}$, where $\omega \in (0,1)$ is the compression factor. Under Assumptions A.1-A.5, for $T \geq \frac{256a^2(1+m)\tilde{G}^2}{\omega^2}$, the expected value of F evaluated at the time averaged vector $\overline{\mathbf{x}}_{avg}^{(T)} := \frac{1}{T}\sum_{t=1}^T \mathbf{x}^{(t)}$ satisfies:

$$\mathbb{E}[F(\overline{\mathbf{x}}_{avg}^{(T)})] - F(\mathbf{x}^*) \leq \frac{2R^2}{a\sqrt{T}} + \frac{a}{\sqrt{T}}\left[\frac{16}{\omega^2}d^2(1+m)G^2 + C^2\right]$$

$$+\frac{2\sqrt{m}\tilde{G}RC}{\delta ar\sqrt{T}}+\frac{4RG}{rT}+\frac{4\sqrt{n}G}{T} \quad (20)$$

where $r \leq \frac{R}{\sqrt{n}}$. For any $i \in [n]$, $j \in \mathcal{N}_i$, we have:

$$\begin{aligned}
&\mathbb{E}\left[g_{ij}(\overline{\mathbf{x}}_{i,avg}^{(T)}, \overline{\mathbf{x}}_{j,avg}^{(T)})\right] \\
&\leq \frac{1}{T^{1/4}}\left[\sqrt{\frac{8GR}{a}}+\sqrt{\frac{8\delta a(R+r\sqrt{n})G}{r}+8GR\delta a}\right] \\
&+\frac{1}{\sqrt{T}}\sqrt{\left(\frac{32}{\omega^2}d^2(1+m)G^2+2C^2\right)+\frac{4\sqrt{m}\tilde{G}RC}{r\delta a^2}+4R^2\delta} \\
&+\frac{1}{\sqrt{T}}\sqrt{\delta a^2\left(\frac{32}{\omega^2}d^2(1+m)G^2+2C^2\right)+\frac{4\sqrt{m}\tilde{G}RC}{r}} \\
&+\frac{1}{T^{3/4}}\sqrt{\frac{8(R+r\sqrt{n})G}{ra}} \qquad (21)
\end{aligned}$$

where $\overline{\mathbf{x}}_{k,avg}^{(T)}$ is time averaged parameter of node k.

The above result establishes that for a given compression requirement $\omega \in (0,1)$, the sub-optimality of the objective $\mathbb{E}[F(\overline{\mathbf{x}}_{avg}^{(T)})] - F(\mathbf{x}^*)$ is $\mathcal{O}\left(\frac{1}{T^{1/2}}\right)$. Similarly, the expected constraint violation for $i \in [n]$ and $j \in \mathcal{N}_i$ given by $\mathbb{E}\left[g_{ij}(\overline{\mathbf{x}}_{i,avg}^{(T)}, \overline{\mathbf{x}}_{j,avg}^{(T)})\right]$ is $\mathcal{O}\left(\frac{1}{T^{1/4}}\right)$. Thus, in effect by choosing a large enough value of T, the number of iterations Algorithm 2 is run for, the obtained stacked parameter $\overline{\mathbf{x}}_{avg}^{(T)}$ is a good estimate of the optimal solution of the overall objective (1). Moreover, the result obtained matches the rate that was obtained for the sample feedback case in Theorem 1, where the nodes had access to the samples at every stage. Theorem 2 thus establishes that even when node access to samples is not assumed, but rather only to a pair of values of the local objectives, the derived convergence rate suffers no degradation.

4.3 Convergence analysis

As done earlier for proof of bandit feedback, we use a compact notation by stacking together the parameters across the nodes. The modified Lagrangian in (17) for a time step $t \in [T]$ in this notation is given as:

$$\tilde{\mathcal{L}}(\mathbf{x}^{(t)}, \boldsymbol{\lambda}^{(t)}) = \tilde{f}(\mathbf{x}^{(t)}, \boldsymbol{\xi}^{(t)}) + \langle \boldsymbol{\lambda}^{(t)}, \mathbf{g}(\mathbf{x}^{(t)})\rangle - \frac{\delta\eta}{2}\|\boldsymbol{\lambda}^{(t)}\|^2 \quad (22)$$

where $\mathbf{x}^{(t)}$, is of size nd, $\boldsymbol{\lambda}^{(t)}$ is of size m, and $\boldsymbol{\xi}^{(t)}$ is collection of samples across all the nodes at time t. We construct another quantity of interest:

$$\mathcal{H}(\mathbf{x}^{(t)}, \boldsymbol{\lambda}^{(t)}) = \widetilde{\mathcal{L}}(\mathbf{x}^{(t)}, \boldsymbol{\lambda}^{(t)}) + \langle \mathbf{p}^{(t)} - \widetilde{\mathcal{L}}(\mathbf{x}^{(t)}, \boldsymbol{\lambda}^{(t)}), \mathbf{x}^{(t)}\rangle \quad (23)$$

It can be seen that $\mathcal{H}(\mathbf{x}^{(t)}), \boldsymbol{\lambda}^{(t)}$ is convex in the parameter $\mathbf{x}^{(t)}$ and concave in $\boldsymbol{\lambda}^{(t)}$ for any t. Further, the gradients of the function $\mathcal{H}(\mathbf{x}^{(t)}), \boldsymbol{\lambda}^{(t)}$ satisfy:

$$\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x}^{(t)}, \boldsymbol{\lambda}^{(t)}) = \mathbf{p}^{(t)}, \quad \nabla_{\boldsymbol{\lambda}}\mathcal{H}(\mathbf{x}^{(t)}, \boldsymbol{\lambda}^{(t)}) = \nabla_{\boldsymbol{\lambda}}\widetilde{\mathcal{L}}(\mathbf{x}^{(t)}, \boldsymbol{\lambda}^{(t)})$$

To derive our results, we consider another auxiliary result along the ones stated earlier in Section 3.3.

Fact 5. *Under Assumptions A.2 and A.3, for all $t \in [T]$, $i \in [n]$ and any $\mathbf{u}, \mathbf{v} \in \mathcal{X}$, we have:*

$$\mathbb{E}_{\boldsymbol{\xi}_i^{(t)}}[f_i(\mathbf{u}, \boldsymbol{\xi}_i^{(t)}) - f_i(\mathbf{v}, \boldsymbol{\xi}_i^{(t)})]^2 \leq 4G_i^2\|\mathbf{u}-\mathbf{v}\|^2$$

where $\mathbb{E}_{\boldsymbol{\xi}_i^{(t)}}[.]$ denotes expectation w.r.t. sampling at time-step t for the node i. See [59] for proof.

4.3.1 Proof of Theorem 2

We first establish a relationship between the primal, dual variables in Algorithm 2 and the function $\mathcal{H}$ defined in (23). This following lemma can be seen as a counterpart of Lemma 1 in the bandit feedback case.

Lemma 3. Consider the update steps in Algorithm 2 with learning rate η. Under assumptions A.1-A.4, for any $\mathbf{x} \in \widetilde{\mathcal{X}}^n$ and $\boldsymbol{\lambda} \in \mathbb{R}^m$ with $\boldsymbol{\lambda} \succeq \mathbf{0}$, the summation of the function $\mathcal{H}$ (defined in (23)) satisfies:

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\left[\mathcal{H}(\mathbf{x}^{(t)}, \boldsymbol{\lambda}) - \mathcal{H}(\mathbf{x}, \boldsymbol{\lambda}^{(t)})\right] \\
&\leq \frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}\left(2\|\mathbf{p}^{(t)}\|^2 + \|\nabla_{\boldsymbol{\lambda}}\widetilde{\mathcal{L}}_t(\mathbf{x}^{(t)}, \boldsymbol{\lambda}^{(t)})\|^2\right) \\
&+\frac{1}{2\eta}\sum_{t=1}^{T}\|\mathbf{x}^{(t)}-\widetilde{\mathbf{x}}^{(t)}\|^2 + \frac{1}{2\eta}\sum_{t=1}^{T}\left(\|\boldsymbol{\lambda}\|^2 + 4R^2\right)
\end{aligned}$$

Now consider $\mathbf{x}^* \in \mathcal{X}^n$, then by definition of $\widetilde{\mathcal{X}}$, we have $(1-\alpha)\mathbf{x}^* + \alpha\tilde{\mathbf{y}}_0 \in \widetilde{\mathcal{X}}^n$ for $\alpha = \frac{\zeta}{r}$ where $\tilde{\mathbf{y}}_0$ and r are defined in Assumption A.5 [3]. Substituting $\mathbf{x} = (1-\alpha)\mathbf{x}^* + \alpha\tilde{\mathbf{y}}_0$ in the result from Lemma 3 gives us:

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\left[\mathcal{H}(\mathbf{x}^{(t)}, \boldsymbol{\lambda}) - \mathcal{H}((1-\alpha)\mathbf{x}^* + \alpha\tilde{\mathbf{y}}_0, \boldsymbol{\lambda}^{(t)})\right] \\
&\leq \frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}\left(2\|\mathbf{p}^{(t)}\|^2 + \|\nabla_{\boldsymbol{\lambda}}\widetilde{\mathcal{L}}_t(\mathbf{x}^{(t)}, \boldsymbol{\lambda}^{(t)})\|^2\right)
\end{aligned}$$

[3] Here, $\tilde{\mathbf{y}}_0 \in \mathbb{R}^{nd}$ denotes the stacking of the d dimensional vector $\mathbf{y}_0$ defined in Assumption A.5

$$+\frac{1}{2\eta}\sum_{t=1}^{T}\|\mathbf{x}^{(t)}-\widetilde{\mathbf{x}}^{(t)}\|^2+\frac{1}{2\eta}\sum_{t=1}^{T}\left(\|\boldsymbol{\lambda}\|^2+4R^2\right) \quad (24)$$

The following result bounds the error $\mathbb{E}\|\mathbf{e}^{(t)}\|^2 := \mathbb{E}\|\mathbf{x}^{(t)}-\widetilde{\mathbf{x}}^{(t)}\|^2$ for any time t in terms of the summation of $\mathbb{E}\|\mathbf{p}^{(t)}\|$; see [59] for proof.

Lemma 4. Consider the error $\mathbf{e}^{(t)} := \mathbf{x}^{(t)}-\widetilde{\mathbf{x}}^{(t)}$ for any $t \in [T]$. We have:

$$\mathbb{E}\|\mathbf{e}^{(t)}\|^2 \leq \frac{2\eta^2}{\omega}\sum_{k=0}^{t-2}\left(1-\frac{\omega}{2}\right)^k\mathbb{E}\|\mathbf{p}^{(t-k-1)}\|^2$$

Using the result from Lemma 4 in (24) and the double sum trick similar to the updates from (12) to (13) yields:

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}\left[\mathcal{H}(\mathbf{x}^{(t)},\boldsymbol{\lambda})-\mathcal{H}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0,\boldsymbol{\lambda}^{(t)})\right]\\&\leq\frac{\eta}{2}\sum_{t=1}^{T}\left(\left(2+\frac{4}{\omega^2}\right)\|\mathbf{p}^{(t)}\|^2+\|\nabla_{\boldsymbol{\lambda}}\widetilde{\mathcal{L}}_t(\mathbf{x}^{(t)},\boldsymbol{\lambda}^{(t)})\|^2\right)\\&\quad+\frac{1}{2\eta}\mathbb{E}\left(4R^2+\|\boldsymbol{\lambda}\|^2\right) \quad (25)\end{aligned}$$

We now provide bounds for the first and second terms on the R.H.S. of (25) in Proposition 6 below. The proof of this proposition is provided in [59].

Proposition 6. *For the update steps given in Algorithm 2, under Assumptions A.2-A.4, for any $t \in [T]$, we have:*

(i) $\mathbb{E}\|\mathbf{p}^{(t)}\|^2 \leq 4d^2(1+m)G^2+4(1+m)\tilde{G}^2\mathbb{E}\|\boldsymbol{\lambda}^{(t)}\|^2$
(ii) $\mathbb{E}\|\nabla_{\boldsymbol{\lambda}}\widetilde{\mathcal{L}}_t(\mathbf{x}^{(t)},\boldsymbol{\lambda}^{(t)})\|^2 \leq 2C^2+2\delta^2\eta^2\mathbb{E}\|\boldsymbol{\lambda}^{(t)}\|^2$

Substituting the bounds from Proposition 6 in (25) and using that fact $\frac{2}{\omega^2} > 1$, we have:

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}\left[\mathcal{H}(\mathbf{x}^{(t)},\boldsymbol{\lambda})-\mathcal{H}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0,\boldsymbol{\lambda}^{(t)})\right]\\&\leq\frac{1}{2\eta}\left(4R^2+\|\boldsymbol{\lambda}\|^2\right)+\eta T\left[\frac{16}{\omega^2}d^2(1+m)G^2+C^2\right]\\&\quad+\eta\left[\frac{16}{\omega^2}(1+m)\tilde{G}^2+\delta^2\eta^2\right]\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{\lambda}^{(t)}\|^2 \quad (26)\end{aligned}$$

We now express the L.H.S. of (26) in terms of the Lagrangian $\widetilde{\mathcal{L}}$. This relation is provided in Proposition 7 below, which is proved in [59].

Proposition 7. *For any $\boldsymbol{\lambda} \in \mathbb{R}^m$ with $\boldsymbol{\lambda} \succeq \mathbf{0}$, the updates of Algorithm 2 satisfy:*

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}\left[\mathcal{H}(\mathbf{x}^{(t)},\boldsymbol{\lambda})-\mathcal{H}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0,\boldsymbol{\lambda}^{(t)})\right]\\&=\sum_{t=1}^{T}\mathbb{E}\left[\widetilde{\mathcal{L}}_t(\mathbf{x}^{(t)},\boldsymbol{\lambda})-\widetilde{\mathcal{L}}_t((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0,\boldsymbol{\lambda}^{(t)})\right]\end{aligned}$$

where $\mathbf{x}^*$ is the optimal parameter value for the objective (1), and $\mathcal{H}$, $\widetilde{\mathcal{L}}$ are defined in (23) and (22), respectively.

Proposition 7 implies the following for (26):

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}\left[\widetilde{\mathcal{L}}_t(\mathbf{x}^{(t)},\boldsymbol{\lambda})-\widetilde{\mathcal{L}}_t((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0,\boldsymbol{\lambda}^{(t)})\right]\\&\leq\frac{1}{2\eta}\left(4R^2+\|\boldsymbol{\lambda}\|^2\right)+\eta T\left[\frac{16}{\omega^2}d^2(1+m)G^2+C^2\right]\\&\quad+\eta\left[\frac{16}{\omega^2}(1+m)\tilde{G}^2+\delta^2\eta^2\right]\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{\lambda}^{(t)}\|^2\end{aligned}$$

Using the definition of $\widetilde{\mathcal{L}}$ from (22) on the L.H.S. of the above, and rearranging the terms, we have:

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}\left[\widetilde{f}(\mathbf{x}^{(t)},\boldsymbol{\xi}^{(t)})-\widetilde{f}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0,\boldsymbol{\xi}^{(t)})\right]-\frac{\delta\eta T}{2}\|\boldsymbol{\lambda}\|^2\\&+\left\langle\boldsymbol{\lambda},\mathbb{E}\sum_{t=1}^{T}\mathbf{g}(\mathbf{x}^{(t)})\right\rangle-\mathbb{E}\sum_{t=1}^{T}\left\langle\boldsymbol{\lambda}^{(t)},\mathbf{g}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0)\right\rangle\\&\leq\frac{1}{2\eta}\left(4R^2+\|\boldsymbol{\lambda}\|^2\right)+\eta T\left[\frac{16}{\omega^2}d^2(1+m)G^2+C^2\right]\\&\quad+\eta\left[\frac{16}{\omega^2}(1+m)\tilde{G}^2+\delta^2\eta^2-\frac{\delta}{2}\right]\sum_{t=1}^{T}\mathbb{E}\|\boldsymbol{\lambda}^{(t)}\|^2 \quad (27)\end{aligned}$$

Similar to what we did for the sample feedback case in (14), we choose the following value of δ to make the coefficient of the last term in (27) negative:

$$\delta=\frac{1-\sqrt{1-\frac{256\eta^2(1+m)\tilde{G}^2}{\omega^2}}}{4\eta^2}$$

As before, we require running the algorithm for $T \geq \frac{256a^2(1+m)\tilde{G}^2}{\omega^2}$ for the choice $\eta=\frac{a}{\sqrt{T}}$, and for $T\to\infty$ (i.e., $\eta\to 0$), the above value of δ converges to a positive constant. Plugging the value of δ in (27) yields:

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}\left[\widetilde{f}(\mathbf{x}^{(t)},\boldsymbol{\xi}^{(t)})-\widetilde{f}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0,\boldsymbol{\xi}^{(t)})\right]-\frac{\delta\eta T}{2}\|\boldsymbol{\lambda}\|^2\\&+\left\langle\boldsymbol{\lambda},\mathbb{E}\sum_{t=1}^{T}\mathbf{g}(\mathbf{x}^{(t)})\right\rangle-\mathbb{E}\left[\sum_{t=1}^{T}\left\langle\boldsymbol{\lambda}^{(t)},\mathbf{g}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0)\right\rangle\right]\end{aligned}$$

$$\leq \frac{1}{2\eta}\left(4R^2+\|\boldsymbol{\lambda}\|^2\right)+\eta T\left[\frac{16}{\omega^2}d^2(1+m)G^2+C^2\right] \quad (28)$$

Our goal is to derive a bound for the sub-optimality of the function $F(\mathbf{x}^{(t)})$. To this end, we will now bound the terms on the L.H.S. of (28) in terms of the function F. We first consider the first term on the L.H.S. of (28). From the definitions of f and $\widetilde{f}$ provided in Fact 4:

$$\begin{aligned}
&\mathbb{E}\left[|\widetilde{f}(\mathbf{x}^{(t)},\boldsymbol{\xi}^{(t)})-f(\mathbf{x}^{(t)},\boldsymbol{\xi}^{(t)})|\right]\\
&\overset{(a)}{=}\mathbb{E}\left[\left|\sum_{i=1}^{n}f_i(\mathbf{x}_i^{(t)}+\zeta\mathbf{u}_i^{(t)},\boldsymbol{\xi}_i^{(t)})-f_i(\mathbf{x}_i^{(t)},\boldsymbol{\xi}_i^{(t)})\right|\right]\\
&\overset{(b)}{\leq}\mathbb{E}\left[\sum_{i=1}^{n}\left|f_i(\mathbf{x}_i^{(t)}+\zeta\mathbf{u}_i^{(t)},\boldsymbol{\xi}_i^{(t)})-f_i(\mathbf{x}_i^{(t)},\boldsymbol{\xi}_i^{(t)})\right|\right]\\
&\overset{(c)}{\leq}\mathbb{E}\left[\sum_{i=1}^{n}\sqrt{\mathbb{E}_{\boldsymbol{\xi}_i^{(t)}}\left[f_i(\mathbf{x}_i^{(t)}+\zeta\mathbf{u}_i^{(t)},\boldsymbol{\xi}_i^{(t)})-f_i(\mathbf{x}_i^{(t)},\boldsymbol{\xi}_i^{(t)})\right]^2}\right]
\end{aligned} \quad (29)$$

where in (a), $\{\mathbf{u}_i^{(t)}\}_{i=1}^n$ denote random vectors uniformly distributed over $\mathbb{B}$, (b) uses the triangle inequality, (c) uses the fact $\mathbb{E}[A]\leq\sqrt{\mathbb{E}[A^2]}$ via Jensen's inequality. From Proposition 5 and the fact $\|\mathbf{u}_i^{(t)}\|^2=1$ for all $i\in[n]$ (as they lie on the unit sphere $\mathbb{S}$), we have:

$$\mathbb{E}_{\boldsymbol{\xi}_i^{(t)}}[f_i(\mathbf{x}_i^{(t)}+\zeta\mathbf{u}_i^{(t)},\boldsymbol{\xi}_i^{(t)})-f_i(\mathbf{x}_i^{(t)},\boldsymbol{\xi}_i^{(t)})]^2\leq 4G_i^2\zeta^2 \quad (30)$$

Plugging the bound from (30) in (29) and noting that $\sum_{i=1}^n G_i\leq\sqrt{n}G$ (using the fact that $G^2=\sum_{i=1}^n G_i^2$):

$$\mathbb{E}\left[|\widetilde{f}(\mathbf{x}^{(t)},\boldsymbol{\xi}^{(t)})-f(\mathbf{x}^{(t)},\boldsymbol{\xi}^{(t)})|\right]\leq 2\zeta\sqrt{n}G$$

Using Jensen's inequality for the L.H.S. of above equation and rearranging the terms finally yields:

$$\mathbb{E}[\widetilde{f}(\mathbf{x}^{(t)},\boldsymbol{\xi}^{(t)})]\geq\mathbb{E}[F(\mathbf{x}^{(t)})]-2\zeta\sqrt{n}G \quad (31)$$

The steps to bound the second term on the L.H.S. of (28) are similar. We note that:

$$\begin{aligned}
&\mathbb{E}\left[|\widetilde{f}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0,\boldsymbol{\xi}^{(t)})-f(\mathbf{x}^*,\boldsymbol{\xi}^{(t)})|\right]\\
&\leq\mathbb{E}\sum_{i=1}^{n}\Big(\mathbb{E}_{\boldsymbol{\xi}_i^{(t)}}\Big[f_i((1-\alpha)\mathbf{x}_i^*+\alpha\mathbf{y}_0+\zeta\mathbf{u}_i^{(t)},\boldsymbol{\xi}^{(t)})\\
&\qquad -f_i(\mathbf{x}_i^*,\boldsymbol{\xi}_i^{(t)})\Big]^2\Big)^{1/2}
\end{aligned} \quad (32)$$

where the inequality follows the same arguments we used for arriving at (29). Further using Proposition 5, we have:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi}_i^{(t)}}\left[f_i((1-\alpha)\mathbf{x}_i^*+\alpha\mathbf{y}_0+\zeta\mathbf{u}_i^{(t)},\boldsymbol{\xi}^{(t)})-f_i(\mathbf{x}_i^*,\boldsymbol{\xi}_i^{(t)})\right]^2\\
&\leq 4G_i^2\|-\alpha\mathbf{x}_i^*+\alpha\mathbf{y}_0+\zeta\mathbf{u}_i^{(t)}\|^2
\end{aligned} \quad (33)$$

Plugging in the bound from (33) into (32), using Fact 1 for $\mathbf{x}_i^*,\mathbf{y}_0\in\mathcal{X}$ along with $\|\mathbf{u}_i^{(t)}\|=1$ for all $i\in[n]$, and Jensen's inequality, we have:

$$\mathbb{E}[\widetilde{f}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0,\boldsymbol{\xi}^{(t)})]\leq F(\mathbf{x}^*)+4G\alpha R+2\zeta G\sqrt{n} \quad (34)$$

Further, we can also simplify other terms on the L.H.S. of (28). We note that:

$$\begin{aligned}
\sum_{t=1}^{T}\left\langle\boldsymbol{\lambda}^{(t)},\mathbf{g}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0)\right\rangle&=\sum_{t=1}^{T}\left\langle\boldsymbol{\lambda}^{(t)},\mathbf{g}(\mathbf{x}^*)\right\rangle\\
&+\sum_{t=1}^{T}\left\langle\boldsymbol{\lambda}^{(t)},\mathbf{g}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0)-\mathbf{g}(\mathbf{x}^*)\right\rangle\\
&\leq\sum_{t=1}^{T}\|\boldsymbol{\lambda}^{(t)}\|\|\mathbf{g}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0)-\mathbf{g}(\mathbf{x}^*)\|
\end{aligned} \quad (35)$$

where to obtain the last inequality we have used the fact that $\langle\boldsymbol{\lambda}^{(t)},\mathbf{g}(\mathbf{x}^*)\rangle\leq 0$ for all $t\in[T]$ and the Cauchy-Schwarz inequality. For the second term in the product on the R.H.S. in (35), using (7) in Assumption 3 $g(\mathbf{x}_i,\mathbf{x}_j)$ are G_{ij}-Lipschitz for all $i,j\in[n]$, we have:

$$\begin{aligned}
&\|\mathbf{g}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0)-\mathbf{g}(\mathbf{x}^*)\|^2\\
&\leq\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}G_{ij}^2\|-\alpha\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0\|^2\leq 4\alpha^2R^2m\tilde{G}^2
\end{aligned} \quad (36)$$

where $\tilde{G}:=\max_{i\in[n],j\in\mathcal{N}_i}G_{ij}$ and the last inequality follows from noting that $\mathbf{x}^*,\tilde{\mathbf{y}}_0\in\mathcal{X}^n$ and using Fact 1. We now bound the first term in the product on the R.H.S. in (35). From the update equation of $\boldsymbol{\lambda}^{(t)}$ in line 13 of Algorithm 2, we have:

$$\begin{aligned}
\|\boldsymbol{\lambda}^{(t+1)}\|&\leq\|\boldsymbol{\lambda}^{(t)}+\eta\nabla_{\boldsymbol{\lambda}}\widetilde{\mathcal{L}}_t(\mathbf{x}^{(t)},\boldsymbol{\lambda}^{(t)})\|\\
&\leq(1-\delta\eta^2)\|\boldsymbol{\lambda}^{(t)}\|+\eta C
\end{aligned}$$

where the second inequality follows from the gradient update for the dual variable (5), the triangle inequality, the fact that $\delta\eta^2\leq 1$ (since an upper bound for δ is $\frac{1}{4\eta^2}$) and Assumption 4 to bound $\|\mathbf{g}(\mathbf{x}^{(t)})\|_2$. Continuing the recursion till $t=1$, it can be shown that $\|\boldsymbol{\lambda}^{(t)}\|\leq\frac{C}{\delta\eta}$, $\forall t\in[T]$. Using this bound, and (36) in (35) leads to:

$$\sum_{t=1}^{T}\left\langle\boldsymbol{\lambda}^{(t)},\mathbf{g}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0)\right\rangle\leq\frac{2\alpha RC\sqrt{m}\tilde{G}T}{\delta\eta} \quad (37)$$

Finally, using bounds from (31), (34), (37) in (28) yields:

$$\sum_{t=1}^{T}\mathbb{E}\left[F(\mathbf{x}^{(t)})-F(\mathbf{x}^*)\right]-\frac{\delta\eta T}{2}\|\boldsymbol{\lambda}\|^2 + \left\langle\boldsymbol{\lambda},\mathbb{E}\sum_{t=1}^{T}\mathbf{g}(\mathbf{x}^{(t)})\right\rangle-\mathbb{E}\left[\sum_{t=1}^{T}\left\langle\boldsymbol{\lambda}^{(t)},\mathbf{g}((1-\alpha)\mathbf{x}^*+\alpha\tilde{\mathbf{y}}_0)\right\rangle\right] \leq \frac{1}{2\eta}\left(4R^2+\|\boldsymbol{\lambda}\|^2\right)+\eta T\left[\frac{16}{\omega^2}d^2(1+m)G^2+C^2\right] + 4G\alpha RT+4\zeta G\sqrt{n}T+\frac{2\alpha RC\sqrt{m}\tilde{G}T}{\delta\eta} \quad (38)$$

Setting $\boldsymbol{\lambda}=\frac{\left[\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{g}(\mathbf{x}^{(t)})\right]\right]^+}{\delta\eta T+\frac{1}{\eta}}$ in (38) gives:

$$\sum_{t=1}^{T}\left(\mathbb{E}\left[F(\mathbf{x}^{(t)})\right]-F(\mathbf{x}^*)\right) + \sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}\frac{\left(\left[\mathbb{E}\left[\sum_{t=1}^{T}g_{ij}(\mathbf{x}_i^{(t)},\mathbf{x}_j^{(t)})\right]\right]^+\right)^2}{2\left(\delta\eta T+\frac{1}{\eta}\right)} \leq \frac{2R^2}{\eta}+\eta T\left[\frac{16}{\omega^2}d^2(1+m)G^2+C^2\right]+\frac{CT2\sqrt{m}\tilde{G}\alpha R}{\delta\eta} + 4\alpha RGT+4\zeta\sqrt{n}GT \quad (39)$$

Dividing both sides of (39) by T and noting that the second term on the L.H.S. of (39) is positive, we can bound the time-average sub-optimality of F as:

$$\sum_{t=1}^{T}\frac{\left(\mathbb{E}\left[F(\mathbf{x}^{(t)})\right]-F(\mathbf{x}^*)\right)}{T}\leq\frac{2R^2}{\eta T}+\eta\left[\frac{16}{\omega^2}d^2(1+m)G^2\right] + C^2\eta+2\sqrt{m}\tilde{G}\alpha R\frac{C}{\delta\eta}+4\alpha RG+4\zeta\sqrt{n}G$$

Using the convexity of F and setting the values $\eta=\frac{a}{\sqrt{T}}$, $\zeta=\frac{1}{T}$ and $\alpha=\frac{1}{rT}$ for some positive constant a, r, concludes the proof for the suboptimality of the function F given in (20) of Theorem 2. We now consider the expected constraint violations. From Fact 3, we have that $\forall\mathbf{x}\in\mathcal{X}^n$, $\mathbb{E}[F(\mathbf{x})]-F(\mathbf{x}^*)>-4GR$. Using this relation in (39) gives:

$$\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}\left(\left[\mathbb{E}\left[\sum_{t=1}^{T}g_{ij}(\mathbf{x}_i^{(t)},\mathbf{x}_j^{(t)})\right]\right]^+\right)^2 \leq \frac{4R^2}{\eta^2}+T\left[\left(\frac{32}{\omega^2}d^2(1+m)G^2+2C^2\right)+\frac{4\sqrt{m}\tilde{G}\alpha RC}{\delta\eta^2} + \frac{8(\alpha R+\zeta\sqrt{n})G}{\eta}+4R^2\delta+\frac{8GR}{\eta}\right] + T^2\left[\delta\eta^2\left(\frac{32}{\omega^2}d^2(1+m)G^2+2C^2\right)+4\sqrt{m}\tilde{G}\alpha RC + 8\delta\eta(\alpha R+\zeta\sqrt{n})G+8GR\delta\eta\right]$$

Note that the above bound also holds for a given $i\in[n]$ and $j\in\mathcal{N}_i$, that is, the R.H.S. of the above equation is also a bound for the term $\left(\left[\mathbb{E}\left[\sum_{t=1}^{T}g_{ij}(\mathbf{x}_i^{(t)},\mathbf{x}_j^{(t)})\right]\right]^+\right)^2$. Taking the square root of both sides and using the fact $\sqrt{\sum_{i=1}^{n}c_i}\leq\sum_{i=1}^{n}\sqrt{c_i}$ for positive $c_1,\ldots,c_n$, we get:

$$\mathbb{E}\left[\sum_{t=1}^{T}g_{ij}(\mathbf{x}_i^{(t)},\mathbf{x}_j^{(t)})\right]\leq\frac{2R}{\eta} + \sqrt{T}\left[\left(\frac{32}{\omega^2}d^2(1+m)G^2+2C^2\right)+4\sqrt{m}\tilde{G}\alpha R\frac{C}{\delta\eta^2} + \frac{8(\alpha R+\zeta\sqrt{n})G}{\eta}+4R^2\delta+\frac{8GR}{\eta}\right]^{1/2} + T\left[\delta\eta^2\left(\frac{32}{\omega^2}d^2(1+m)G^2+2C^2\right)+4\sqrt{m}\tilde{G}\alpha RC + 8\delta\eta(\alpha R+\zeta\sqrt{n})G+8GR\delta\eta\right]^{1/2}$$

Dividing both sides of the above by T, using the convexity of constraint function g_{ij}, and substituting the values $\eta=\frac{a}{\sqrt{T}}$, $\zeta=\frac{1}{T}$ and $\alpha=\frac{1}{rT}$ concludes proof of (21). □

5 Experiments

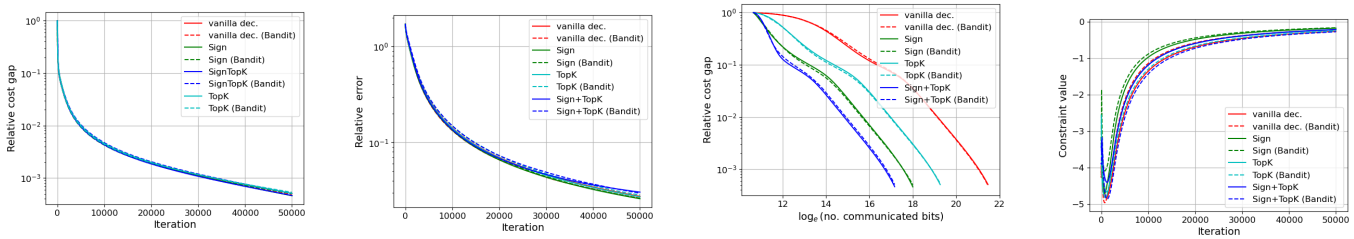
5.1 QCQP Objective

5.1.1 Setup and Hyperparameters

We consider decentralized optimization on a randomly generated Erdos-Renyi graph of $n=30$ nodes with an edge probability of 0.15. For each node $i\in[n]$, we consider a quadratic objective given by $f_i(\mathbf{x}_i,\boldsymbol{\xi})=\mathbf{x}_i^T\mathbf{A}_i\mathbf{x}_i+\mathbf{b}_i^T\mathbf{x}_i$ where $\mathbf{x}_i$ denotes the node model parameter and $\boldsymbol{\xi}_i=(\mathbf{A}_i,\mathbf{b}_i)$ denotes the sample. For each node, $\mathbf{A}_i\in\mathbb{R}^{10\times10}$ is sampled from a Wishart distribution with 10 degrees of freedom identity scaling matrix, and vector $\mathbf{b}_i$ is sampled from a Gaussian distribution with mean and variance drawn uniformly at random from the interval [0,1] in each iteration.

We consider the feasible parameter space $\mathcal{X}$ to be the Euclidean ball of radius $\frac{40}{\sqrt{30}}$ centered at the origin. For each $i\in[n], j\in\mathcal{N}_i$, we model the constraints on the node parameters as $g_{ij}(\mathbf{x}_i,\mathbf{x}_j)=\|\mathbf{x}_i-\mathbf{x}_j\|^2+c_{ij}$ where c_{ij} is independently drawn uniformly at random from $[-5,-3]$. The overall objective is thus given by:

$$\min_{\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}\in\mathcal{X}}F(\mathbf{x})=\sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{\xi}_i}[\mathbf{x}_i^T\mathbf{A}_i\mathbf{x}_i+\mathbf{b}_i^T\mathbf{x}_i] \quad (40)$$
$$\text{s.t. } \|\mathbf{x}_i-\mathbf{x}_j\|_2^2+c_{ij}\leq0,\quad\forall i\in[n],j\in\mathcal{N}_i$$



(a) Comparison of relative cost gap $\frac{F(\bar{\mathbf{x}}^{(t)})-F(\mathbf{x}^*)}{F(\bar{\mathbf{x}}^{(1)})-F(\mathbf{x}^*)}$ at iteration t.

(b) Comparison of relative parameter error $\frac{\|\bar{\mathbf{x}}^{(t)}-\mathbf{x}^*\|}{\|\mathbf{x}^*\|}$ at iteration t.

(c) Comparison of relative cost gap comparison for number of bits communicated for different schemes.

(d) Constraint value $g_{ij}(\bar{\mathbf{x}}_i^{(t)}, \bar{\mathbf{x}}_j^{(t)})$ for a randomly chosen edge (i, j).

Fig. 1. Performance comparison for various schemes on a decentralized QCQP objective in (40)

where $\mathbf{x}$ denotes concatenation of $\{\mathbf{x}_i, \ldots, \mathbf{x}_n\}$. Note that choosing $c_{ij} \leq 0$ for all $i \in [n], j \in \mathcal{N}_i$ implies that the above QCQP has a non-empty feasible set. We set $\eta = 0.001$, and choose $\delta = 100$, and run all considered schemes for a total of 5×10^4 iterations. For gradient estimation in case of bandit feedback, we take $\zeta = 10^{-4}$.

5.1.2 Results

The simulation results for optimizing objective (40) are presented in Figure 1, where we compare vanilla decentralized (no compression) algorithm with our proposed compressed optimization procedure using *Sign* [45], *TopK* [23] and composed *Sign* + *TopK* [31] compression operators. Schemes with *'Bandit'* in parenthesis indicate those implemented via Algorithm 2 for the case of gradient estimation in bandit feedback, and via Algorithm 1 with sample feedback otherwise. Figure 1a shows the relative cost gap for the objective given by $\frac{F(\bar{\mathbf{x}}^{(t)})-F(\mathbf{x}^*)}{F(\bar{\mathbf{x}}^{(1)})-F(\mathbf{x}^*)}$, and Figure 1b shows the difference of the parameter from the optimal value normalized to the latter, given by $\frac{\|\bar{\mathbf{x}}^{(t)}-\mathbf{x}^*\|}{\|\mathbf{x}^*\|}$ for iteration t. We conclude that schemes with compression, including the ones implemented via bandit feedback, effectively perform the same as uncompressed vanilla training to minimize the objective. The benefit of our proposed scheme can be seen in Figure 1c, where we plot the relative cost gap with the number of bits communicated among the nodes, assuming precision of 32bit floats. To achieve a target relative cost gap of around 10^{-3}, compressed schemes use significantly fewer bits than vanilla decentralized training, saving a factor of about 7× with *TopK* compression, factor of 30× when using *Sign* compression operation, and a factor of around 50× for the composed *Sign* + *TopK* compression operator. Figure 1d shows the constraint $g_{ij}(\bar{\mathbf{x}}_i^{(t)}, \bar{\mathbf{x}}_j^{(t)})$ for a randomly chosen $i \in [n]$ and $j \in [n]$. The constraint value settles to a negative value, which implies that each scheme arrives at an objective value lying in the feasible space of the problem (40).

In conclusion, our proposed schemes in Algorithms 1 and 2 for communication efficient decentralized optimization provide performance similar to that in the full precision vanilla decentralized method, while saving substantially in the total number of bits communicated among the nodes during the optimization process.

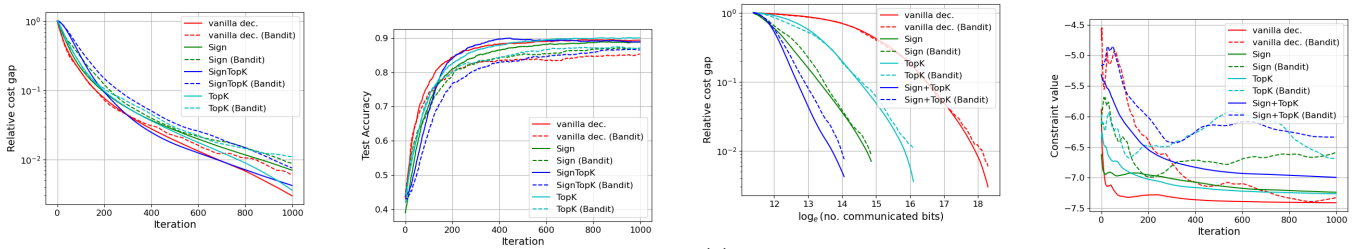
5.2 Logistic Regression Objective

5.2.1 Setup and Hyperparameters

We again work with an Erdos-Renyi graph of $n = 30$ nodes with an edge probability of 0.3. We consider a logistic regression setting where feature vectors $\mathbf{p}_i \in \mathbb{R}^d$ (d=10) for each node are generated from a standard normal distribution. The corresponding output $y_i \in \{1, -1\}$ is sampled under the probability: $p(y_i = 1) = \frac{1}{1+e^{-\mathbf{x}_i^\top \mathbf{p}_i}}$ where $\mathbf{x}_i$ denotes the underlying node model parameter. We generate the underlying model parameters such that they are close (in norm sense) for adjacent nodes. We denote by $\boldsymbol{\xi}_i$ the pair $(\mathbf{p}_i, y_i)$ for each node, which are data samples generated for each iteration of the algorithm. The objective of the nodes is to maximize the log-likelihood of the generated data, which can equivalently be expressed by the following optimization problem:

$$\min_{\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}\in\mathcal{X}} F(\mathbf{x}) = \sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{\xi}_i}[\log(1 + e^{-y_i \mathbf{x}_i^\top \mathbf{p}_i})] \quad (41)$$
$$\text{s.t. } \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + c_{ij} \leq 0, \quad \forall i \in [n], j \in \mathcal{N}_i$$

As in the earlier QCQP formulation, we consider $R = 40$ and the constraints on the node parameters to be $g_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2 + c_{ij}$ where c_{ij} are independently drawn uniformly at random from $[-10, -7]$. We set $\eta = 0.001$, and choose $\delta = 100$, and run all considered schemes for a total of 10^3 iterations. For gradient estimation in case of bandit feedback, we take $\zeta = 10^{-4}$. To evaluate the models for their generalization capabilities, we also evaluate their classification performance on a test set (comprising of 500 samples per node).



(a) Comparison of relative cost gap $\frac{F(\bar{\mathbf{x}}^{(t)})-F(\mathbf{x}^*)}{F(\bar{\mathbf{x}}^{(1)})-F(\mathbf{x}^*)}$ at iteration t.

(b) Comparison of test accuracy performance at iteration t.

(c) Comparison of relative cost gap comparison for number of bits communicated for different schemes.

(d) Constraint value $g_{ij}(\bar{\mathbf{x}}_i^{(t)}, \bar{\mathbf{x}}_j^{(t)})$ for a randomly chosen edge (i, j).

Fig. 2. Performance comparison for various schemes on logistic regression training in (41)

5.2.2 Results

We compare the performance of vanilla decentralized training for optimizing (41) against our proposed method using compression in Figure 2. Figure 2a shows the relative sub-optimality of the different model parameters against the true model [4]. Figure 2b shows the test accuracy performance of the datasets, where we observe all compression schemes achieving similar accuracy as uncompressed vanilla training. To see the gain in using compression, we plot the relative sub-optimality against the total numbers of bits communicated in Figure 2c, where we observe that to achieve a similar level of sub-optimality of around 10^{-2}, compared to vanilla decentralized training, *SignTopK* compression saves a factor of about 50×, *Sign* compression saves a factor of 20× and *TopK* compression saves a factor of around 7×. This, in conclusion, demonstrates the advantage of using our proposed communication efficient scheme for a logistic regression based classification scenario. The constraint values for all the schemes for a randomly chosen edge are shown in Figure 2d, where we observe that all schemes settle to a negative value, and thus end up in the feasible space of the problem (41).

6 Conclusion

We proposed and analyzed a communication-efficient saddle-point algorithm for multi-task decentralized learning under sample feedback and bandit feedback data access scenarios. Our theoretical results demonstrated order-wise same performance as un-compressed training for convex objectives while saving significantly on the number of bits transmitted, which is also corroborated by our numerical experiments.

As many learning paradigms consider non-convex objectives, e.g. Deep Learning, it would be of interest to extend the analysis of our proposed algorithm to such settings as part of future work. It is also of interest to incorporate additional mechanisms for communication reduction along with compression in our proposed algorithm for greater communication efficiency such as local gradient iterations or triggered-communication [31, 52], and theoretically analyze the resulting procedure.

[4] We remark that for a large enough values of c_{ij} such that $g(\mathbf{x}_i, \mathbf{x}_j) \leq 0$ for all $\mathbf{x}_i, \mathbf{x}_j$ (where $\mathbf{x}_i$ denotes the optimal model parameter for node i that generates the data), the optimal solution $\mathbf{x}^*$ is the stacking of all $\mathbf{x}_i, i \in [n]$.

References

[1] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48-61, Jan. 2009.

[2] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM J. Optim.*, vol. 26, no. 3, pp. 1835-1854, 2016.

[3] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592-606, March 2012.

[4] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750-1761, April 2014.

[5] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944-966, 2015.

[6] M.O. Sayin, N.D. Vanli, S.S. Kozat, and T. Başar, "Stochastic subgradient algorithms for strongly convex optimization over distributed networks," *IEEE Transactions on Network Science and Engineering*, 4(4):248-260, October-December 2017.

[7] G. Lan, S. Lee, and Y. Zhou, "Communication-efficient algorithms for decentralized and stochastic optimization," *Math. Program.*, vol. 180, pp. 237-284, 2020.

[8] D. Mateos-Nunez and J. Cortés, "Distributed online convex optimization over jointly connected digraphs," *IEEE Trans. Netw. Sci. Eng.*, vol. 1, no. 1, pp. 23-37, January–June 2014.

[9] M. Akbari, B. Gharesifard, and T. Linder, "Distributed online convex optimization on time-varying directed graphs," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 417-428, September 2017.

[10] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129-4144, August 2014.

[11] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion adaptation over asynchronous networks," *IEEE Trans. Signal Process.*, vol. 64, no. 11, pp. 2835-2850, June 2016.

[12] A. Koppel, B. M. Sadler, and A. Ribeiro, "Proximity without consensus in online multiagent optimization," *IEEE Trans. Signal Process.*, vol. 65, no. 12, pp. 3062-3077, June 2017.

[13] A. S. Bedi, A. Koppel, and K. Rajawat, "Asynchronous saddle point algorithm for stochastic optimization in heterogeneous networks," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1742-1757, April 2019.

[14] A. Kashyap, T. Başar, and R. Srikant, "Quantized consensus," *Automatica*, 43(7):1192-1203, July 2007.

[15] S.R. Etesami and T. Başar, "Convergence time for unbiased quantized consensus over static and dynamic networks," *IEEE Transactions on Automatic Control*, 61(2):443-455, February 2016.

[16] T. Başar, S.R. Etesami, and A. Olshevsky, "Convergence time of quantized Metropolis consensus over time-varying networks," *IEEE Transactions on Automatic Control*, 61(12):4048-4054, December 2016.

[17] M. El Chamie, J. Liu, and T. Başar, "Design and analysis of distributed averaging with quantized communication," *IEEE Transactions on Automatic Control*, 61(12):3870-3884, December 2016.

[18] S. Zhu and B. Chen, "Quantized consensus by the ADMM: Probabilistic versus deterministic quantizers," *IEEE Trans. Signal Processing*, vol. 64, no. 7, pp. 1700-1713, April 2016

[19] J. Zhang, K. You, and T. Başar, "Distributed discrete-time optimization in multi-agent networks using only sign of relative state," *IEEE Trans. Autom. Control*, vol. 64, no. 6, pp. 2352-2367, June 2019.

[20] X. Cao and T. Başar, "Decentralized online convex optimization based on signs of relative states," *Automatica*, 129:109676, July 2021.

[21] X. Cao and T. Başar, "Decentralized online convex optimization with event-triggered communications," *IEEE Transactions on Signal Processing*, 69:284-299, 2021.

[22] D. Alistarh, T. Hoefler, M. Johansson, S. Khirirat, N. Konstantinov, and C. Renggli, "The convergence of sparsified gradient methods," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[23] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," *Advances in Neural Information Processing Systems*, vol. 31, 2018, pp. 4452-4463.

[24] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[25] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798-808, April 2005.

[26] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *Proc. 47th IEEE Conf. Decision & Control*, 2008, pp. 4177-4184.

[27] A. Reisizadeh, A. Mokhtari, H. Hassani, and R. Pedarsani, "An exact quantized decentralized gradient descent algorithm," *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 4934-4947, October 2019.

[28] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1709-1720.

[29] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3478-3487.

[30] J. Wu, W. Huang, J. Huang, and T. Zhang, "Error compensated quantized SGD and its applications to large-scale distributed optimization," in International Conference on Machine Learning, pp. 5325-5333, 2018.

[31] D. Basu, D. Data, C. Karakus, and S. N. Diggavi, "Qsparse-local-SGD: Distributed SGD with quantization, sparsification, and local computations," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[32] X. Cao and T. Başar, "Decentralized multi-agent stochastic optimization with pairwise constraints and quantized communications," *IEEE Trans. on Signal Processing*, vol. 68, pp. 3296-3311, 2020.

[33] K. J. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Linear and Non-linear Programming*, New York, NY, USA: Cambridge Univ. Press, 1958.

[34] A. Nedić and A. Ozdaglar, "Subgradient methods for saddle-point problems," *J. Optim. Theory Appl.*, vol. 142, no. 1, pp. 205-228, 2009.

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

[36] M. Eisen, C. Zhang, L. F. Chamon, D. D. Lee, and A. Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2775-2790, May 2019.

[37] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: Online convex optimization with long term constraints," *J. Mach. Learn. Res.*, vol. 13, pp. 2503-2528, 2012.

[38] X. Cao and K. J. R. Liu, "Online convex optimization with time-varying constraints and bandit feedback," *IEEE Trans. Autom. Control*, vol. 64, no. 7, pp. 2665-2680, July 2019.

[39] A.D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. 16th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2005, pp. 385-394.

[40] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multi-point bandit feedback," in *Proc. 23rd Annu. Conf. Learn. Theory*, 2010, pp. 28-40.

[41] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2788-2806, May 2015.

[42] O. Shamir, "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback," *J. Mach. Learn. Res.*, vol. 18, no. 52, pp. 1-11, 2017.

[43] S. Liu, B. Kailkhura, P.-Y. Chen, P. Ting, S. Chang, and L. Amini, "Zeroth-order stochastic variance reduction for nonconvex optimization," in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, 2018, pp. 3727-3737.

[44] D. Hajinezhad, M. Hong, and A. Garcia, "ZONE: Zeroth order nonconvex multi-agent optimization over networks," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 3995-4010, October 2019.

[45] S. P. Karimireddy , Q. Rebjock, S. Stich, and M. Jaggi, "Error feedback fixes signsgd and other gradient compression schemes," in *International Conference on Machine Learning*, pp. 3252-3261. PMLR, 2019.

[46] N. Singh, D. Data, J. George and S. Diggavi, "SQuARM-SGD: Communication-Efficient Momentum SGD for Decentralized Optimization.", *IEEE Journal on Selected Areas in Information Theory*, 2021, 2(3), pp.954-969.

[47] H. Mania, X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran and M. Jordan, "Perturbed iterate analysis for asynchronous stochastic optimization", *SIAM Journal on Optimization*, vol. 27, np. 4, pp. 2202-2229, 2017.

[48] T. Chen, Q. Ling and G. Giannakis, "An online convex optimization approach to proactive network resource allocation", in *IEEE Transactions on Signal Processing*, vol. 65, no. 24, pp. 6350-6364, 2017.

[49] I. Necoara and V. Nedelcu, "Rate analysis of inexact dual first-order methods application to dual decomposition", in *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1232-1243, 2013.

[50] H. Yu and MJ. Neely, "A simple parallel algorithm with an $O(1/t)$ convergence rate for general convex programs" in *SIAM Journal on Optimization*, vol. 27, no. 2, pp. 759-783, 2017.

[51] N. Singh, D. Data, J. George and S. Diggavi, "SPARQ-SGD: Event-triggered and compressed communication in decentralized optimization", in *IEEE Conference on Decision and Control* , pp. 3449-3456, 2020.

[52] N. Singh, D. Data, J. George and S. Diggavi, "SPARQ-SGD: Event-triggered and compressed communication in decentralized optimization", *IEEE Transactions on Automatic Control*, vol. 68, no. 2, pp. 721-736, 2023.

[53] X. Cao and T. Başar. "Decentralized online convex optimization with compressed communications". *Automatica*, 156:111186, 2023.

[54] S. Bhatnagar, H.L. Prasad, and L.A. Prashanth, "Stochastic recursive algorithms for optimization: Simultaneous perturbation methods". *Vol. 434. Springer, 2012*

[55] L.A. Prashanth, S. Bhatnagar, M. Fu, and S. Marcus. "Adaptive system optimization using random directions stochastic approximation". *IEEE Transactions on Automatic Control 62, no. 5 (2016): 2223-2238.*

[56] J.C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation." *IEEE Transactions on Automatic Control 37.3 (1992): 332-341.*

[57] R.Y. Rubinstein and D. P. Kroese. "Simulation and the Monte Carlo Method". *John Wiley & Sons, 2016.*

[58] L. Nguyen, P.H. Nguyen, M. Dijk, P. Richtárik, K. Scheinberg, and M. Takác. "SGD and Hogwild! Convergence without the bounded gradients assumption." *International Conference on Machine Learning, pp. 3750-3758. PMLR, 2018.*

[59] N. Singh, X. Cao, S. Diggavi, and T. Başar. "Decentralized Multi-Task Stochastic Optimization With Compressed Communications." https://arxiv.org/abs/2112.12373