2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

A Linearly Convergent Robust Compressed Push-Pull Method for Decentralized Optimization

Yiwei Liao, Zhuorui Li, and Shi Pu

Abstract— **In the modern paradigm of multi-agent networks, communication has become one of the main bottlenecks for decentralized optimization, where a large number of agents are involved in minimizing the average of the local cost functions. In this paper, we propose a robust compressed push-pull algorithm (RCPP) that combines gradient tracking with communication compression. In particular, RCPP is compatible with a much more general class of compression operators that allow both relative and absolute compression errors. We show that RCPP achieves linear convergence rate for smooth objective functions satisfying the Polyak-Łojasiewicz condition over general directed networks. Numerical examples verify the theoretical findings and demonstrate the efficiency, flexibility, and robustness of the proposed algorithm.**

I. Introduction

In this paper, we study the decentralized optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}), \tag{1}$$

where n is the number of agents, $\mathbf{x}$ is the global decision variable, and each agent i only has access to its local objective function $f_i : \mathbb{R}^p \to \mathbb{R}$. The goal is to find an optimal and consensual solution through local computation and local sharing of information in a directed communication network.

Decentralized algorithms for solving (1) were well studied in recent years. The seminal work [1] proposed the distributed subgradient descent (DGD) method, where each agent updates its local copy by mixing with the received copies from neighbors in the network and moving towards the local gradient descent direction. However, under a constant step-size, DGD only converges to a neighborhood of the optimal solution. To obtain better convergence results, various works with bias-correction techniques were proposed, including EXTRA [2], exact diffusion [3], and gradient tracking based methods [4]–[7]. These methods achieve linear convergence for minimizing strongly convex and smooth objective functions. Under the more general directed network topology, several modifications have been considered; see [6], [8]–[14] and the references therein.

In decentralized computation, exchanging complete information between neighboring agents may suffer from the communication bottleneck due to the limited energy and/or bandwidth. One of the promising means for reducing the communication costs is applying compression operators [15]–[23]. Most of the works have considered the relative compression error assumption, including unbiased compressors [18]–[20] and contractive biased compressors [16], [17], or the unification of them [21]. Recently, a few works have also considered quantized compression operators with absolute compression errors [22]–[24]. To explore a unified framework for both relative and absolute compression errors, the work in [25] studied finite-bit quantization, but the absolute compression error needs to diminish exponentially fast for the desired convergence. In [26], the unbiased relative compression error was considered together with the absolute compression error, but the latter slows down the algorithmic convergence.

In this paper, we propose a robust compressed push-pull method (RCPP) for decentralized optimization with communication compression over general directed networks. In particular, we consider a more general assumption on the communication compressors, which unifies both relative and absolute compression errors. By employing the dynamic scaling compression technique, RCPP provably achieves linear convergence for minimizing smooth objective functions satisfying the Polyak-Łojasiewicz inequality (PL condition) under the general class of compression operators.

The main contribution of this paper is summarized as follows:

- For decentralized optimization with communication compression, we consider a general class of compression operators, which unifies the commonly used relative and absolute error compression assumptions. Such a condition is most general in the decentralized optimization literature to the best of our knowledge.
- We propose a new method called the robust compressed push-pull algorithm that works over general directed networks. Based on the dynamic scaling compression technique, RCPP provably achieves linear convergence for minimizing smooth objective functions satisfying the PL condition under the general unified assumption on the compression operators.
- Numerical results demonstrate that RCPP is efficient compared to the state-of-the-art methods and robust under various compressors.

Yiwei Liao and Shi Pu are with School of Data Science, Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), China. Zhuorui Li is with H. Milton Stewart School of Industrial and System Engineering, Georgia Institute of Technology, Atlanta, USA. This work is partially supported by National Natural Science Foundation of China (NSFC) (Grant No. 62003287), Shenzhen Research Institute of Big Data (Grant No. T00120220003) and Shenzhen Science and Technology Program (Grant No. RCYX20210609103229031). (emails: liaoyiwei@cuhk.edu.cn, lizhuorui27@gmail.com, pushi@cuhk.edu.cn)

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

In Table I, we compare this paper with related works regarding the assumptions on the compression operators, objective functions, graph topologies and convergence guarantees.

The rest of this paper is organized as follows. We introduce the notation in Section I-A. In Section II, we state the standing assumptions and discuss the compression methods. In Section III, we introduce the RCPP method. In Section IV, we establish the linear convergence of RCPP under communication compression. Numerical experiments are provided to verify the theoretical findings in Section V. Finally, conclusions are given in Section VI.

TABLE I

COMPARISON OF RELATED WORKS ON DECENTRALIZED OPTIMIZATION WITH COMMUNICATION COMPRESSION.

References	relative	absolute	convergence	graph	function
[16], [17]	C [1]	×	sublinear	Und	SVX [4]
[18], [19]	U	×	linear	Und	SVX
[27], [28]	C	×	linear	Und	SVX [27], PL [28]
[21], [29]	G	×	linear	Und	SVX [21], PL [29]
[23], [24], [29]	×	Q	linear	Und	SVX [23], [24], PL [29]
[25]	C	dim-d	linear* [2]	Und	SVX
[26]	U	✓	neighborhood [3]	Und	SVX
[20]	U	×	linear	Di	SVX
[22]	×	Q	linear	Di	SVX
our paper	**G**	**✓**	**linear**	**Di**	**PL**

[1] ‘C’, ‘U’, ‘G’ represent contractive biased, unbiased, general relative compression assumptions, respectively. ‘dim-d’ and ‘Q’ represent dimension-dependent absolute compression assumption and quantizer, respectively. ‘Und’ and ‘Di’ denote undirected and directed graphs, respectively.

[2] * The result has extra requirement, e.g., exponentially decaying error.

[3] The algorithm converges to the neighborhood of the optimal solution.

[4] ‘SVX’ and ‘PL’ represent strongly convex functions and the PL condition, respectively.

A. Notation

A vector is viewed as a column by default. The n-dimensional column vector with all entries equal to 1 is denoted by $\mathbf{1}$. Each agent i holds a local copy $\mathbf{x}_i \in \mathbb{R}^p$ of the decision variable and an auxiliary variable $\mathbf{y}_i \in \mathbb{R}^p$ to track the average gradient. Vectors $\mathbf{x}_i^k$ and $\mathbf{y}_i^k$ represent their corresponding values at the k-th iteration. For simplicity, denote the aggregated variables as $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\intercal \in \mathbb{R}^{n\times p}$, $\mathbf{Y} := [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]^\intercal \in \mathbb{R}^{n\times p}$. At step k, $\mathbf{X}^k$ and $\mathbf{Y}^k$ represent their corresponding values. The other aggregated variables $\mathbf{H}_x$, $\mathbf{H}_y$, $\mathbf{Q}_x$, $\mathbf{Q}_y$, $\widehat{\mathbf{X}}$, $\widehat{\mathbf{Y}}$, $\widetilde{\mathbf{X}}$, and $\widetilde{\mathbf{Y}}$ are defined similarly. The aggregated gradients are $\nabla \mathbf{F}(\mathbf{X}) := [\nabla f_1(\mathbf{x}_1), \nabla f_2(\mathbf{x}_2), \ldots, \nabla f_n(\mathbf{x}_n)]^\intercal \in \mathbb{R}^{n\times p}$. With slight notation abuse, the gradients $\nabla f_i(\mathbf{x}_i)$ and $\nabla f(\mathbf{x})$ are occasionally regarded as row vectors, and the average of all the local gradients is $\nabla \overline{\mathbf{F}}(\mathbf{X}) := \frac{1}{n}\mathbf{1}^\intercal \nabla \mathbf{F}(\mathbf{X}) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i)$. The notations $\|\cdot\|$ and $\|\cdot\|_F$ define the Euclidean norm of a vector and the Frobenius norm of a matrix, respectively.

The set of nodes (agents) is denoted by $\mathcal{N} = \{1, 2, \ldots, n\}$. A directed graph (digraph) is a pair $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where the edge set $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ consists of ordered pairs of nodes. If there exists a directed edge from node i to node j in $\mathcal{G}$, or $(i, j) \in \mathcal{E}$, then i is called the parent node, and j is the child node. The parent node can directly transmit information to the child node, but not the other way around. Let $\mathcal{G}_\mathbf{B} = (\mathcal{N}, \mathcal{E}_\mathbf{B})$ denote a digraph induced by a nonnegative square matrix $\mathbf{B}$, where $(i, j) \in \mathcal{E}_\mathbf{B}$ if and only if $\mathbf{B}_{ji} > 0$. In addition, $\mathcal{R}_\mathbf{B}$ is the set of roots of all the possible spanning trees in $\mathcal{G}_\mathbf{B}$.

II. PROBLEM FORMULATION

In this section, we first provide the basic assumptions on the communication graphs and the objective functions. Then, we introduce a general assumption on the compression operators to unify both the relative and absolute compression errors.

A. Communication graphs and objective functions

Consider the following conditions on the communication graphs among the agents and the corresponding mixing matrices.

Assumption 1: The matrices $\mathbf{R}$ and $\mathbf{C}$ are both supported by a strongly connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, i.e., $\mathcal{E}_\mathbf{R} = \{(j, i) \in \mathcal{N} \times \mathcal{N} \big| \mathbf{R}_{ij} > 0\} \subset \mathcal{E}$ and $\mathcal{E}_\mathbf{C} = \{(j, i) \in \mathcal{N} \times \mathcal{N} \big| \mathbf{C}_{ij} > 0\} \subset \mathcal{E}$. The matrix $\mathbf{R}$ is row stochastic, and $\mathbf{C}$ is column stochastic, i.e., $\mathbf{R}\mathbf{1} = \mathbf{1}$ and $\mathbf{1}^\intercal \mathbf{C} = \mathbf{1}^\intercal$. In addition, there exists at least one node that is a root of spanning trees for both $\mathcal{G}_R$ and $\mathcal{G}_{C^\intercal}$, i.e., $\mathcal{R}_\mathbf{R} \cap \mathcal{R}_{\mathbf{C}^\intercal} \neq \emptyset$.

Remark 1: Assumption 1 is weaker than requiring both $\mathcal{G}_\mathbf{R}$ and $\mathcal{G}_\mathbf{C}$ are strongly connected [12]. It implies that $\mathbf{R}$ has a unique nonnegative left eigenvector $\mathbf{u}_R$ w.r.t. eigenvalue 1 with $\mathbf{u}_R^\intercal \mathbf{1} = n$, and $\mathbf{C}$ has a unique nonnegative right eigenvector $\mathbf{u}_C$ w.r.t. eigenvalue 1 such that $\mathbf{u}_C^\intercal \mathbf{1} = n$. The nonzero entries of $\mathbf{u}_R$ and $\mathbf{u}_C$ correspond to the nodes in $\mathcal{R}_\mathbf{R}$ and $\mathcal{R}_{\mathbf{C}^\intercal}$, respectively. Since $\mathcal{R}_\mathbf{R} \cap \mathcal{R}_{\mathbf{C}^\intercal} \neq \emptyset$, we have $\mathbf{u}_R^\intercal \mathbf{u}_C > 0$. For a more detailed explanation, please refer to [12].

The objective functions are assumed to satisfy the following condition.

Assumption 2: The objective function f satisfies the Polyak-Łojasiewicz inequality (PL condition), i.e.,

$$\|\nabla f(\mathbf{x})\|^2 \geq 2\mu(f(\mathbf{x}) - f(\mathbf{x}^*)), \tag{2}$$

where $\mathbf{x}^*$ is an optimal solution to problem (1). For each agent i, its gradient is L_i-Lipschitz continuous, i.e.,

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{x}')\| \leq L_i \|\mathbf{x} - \mathbf{x}'\|, \ \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^p. \tag{3}$$

Remark 2: If f is μ-strongly convex as commonly assumed, the PL condition is automatically satisfied. From Assumption 2, the gradient of f is L-Lipschitz continuous, where $L = \max\{L_i\}$. We denote $\kappa = L/\mu$ as the condition number.

B. A unified compression assumption

We now present a general assumption on the compression operators which incorporates both relative and absolute compression errors.

Assumption 3: The compression operator $\mathcal{C}\colon \mathbb{R}^d \to \mathbb{R}^d$ satisfies

$$\mathbb{E}_\mathcal{C} \|\mathcal{C}(\mathbf{x}) - \mathbf{x}\|^2 \leq C\|\mathbf{x}\|^2 + \sigma^2, \qquad \forall \mathbf{x} \in \mathbb{R}^d, \tag{4}$$

for some constants $C, \sigma^2 \geq 0$, and the r-scaling of $\mathcal{C}$ satisfies

$$\mathbb{E}_\mathcal{C} \|\mathcal{C}(\mathbf{x})/r - \mathbf{x}\|^2 \leq (1 - \delta)\|\mathbf{x}\|^2 + \sigma_r^2, \quad \forall \mathbf{x} \in \mathbb{R}^d, \tag{5}$$

for some constants $r > 0$, $\delta \in (0, 1]$ and $\sigma_r^2 \geq 0$.

Among the compression conditions considered for decentralized optimization algorithms with convergence guarantees, Assumption 3 is the weakest to the best of our knowledge. Specifically, if there is no absolute error, i.e., $\sigma^2 = \sigma_r^2 = 0$, then Assumption 3 degenerates to the assumption in [21] that unifies the compression operators with relative errors. If there is no relative error, i.e., $C = 0$ and $\delta = 1$, then the condition becomes the assumption on the quantizers in [22], [23]. Therefore, Assumption 3 provides a unified treatment for both relative and absolute compression errors. In addition, if $C < 1$, Assumption 3 reduces to the condition in [25].

III. A Robust Compressed Push-Pull Method

In this section, we first introduce the dynamic scaling compression technique that deals with the absolute compression error. Then, we propose the RCPP algorithm and discuss its connections with the existing methods.

A. The dynamic scaling compression technique

While Assumption 3 provides a unified condition on the compression operators, new challenges are brought to the algorithm design and analysis. Without proper treatment for the compression errors, the algorithmic performance could deteriorate, particularly due to the absolute error that may lead to compression error accumulation. To tackle the challenge, we consider the dynamic scaling compression technique [23]. Consider the operator $\mathcal{Q}(\mathbf{x}) = s_k \mathcal{C}(\mathbf{x}/s_k)$, where s_k is a dynamic parameter related to the iteration k. Then from Assumption 3, we have $\mathbb{E}_{\mathcal{Q}} \|\mathcal{Q}(\mathbf{x}) - \mathbf{x}\|^2 = \mathbb{E}_{\mathcal{C}} \|s_k \mathcal{C}(\mathbf{x}/s_k) - \mathbf{x}\|^2 = s_k^2 \mathbb{E}_{\mathcal{C}} \|\mathcal{C}(\mathbf{x}/s_k) - \mathbf{x}/s_k\|^2 \leq s_k^2(C\|\mathbf{x}/s_k\|^2 + \sigma^2) = C\|\mathbf{x}\|^2 + s_k^2\sigma^2$. Similarly, we know $\mathbb{E}_{\mathcal{Q}} \|\mathcal{Q}(\mathbf{x})/r - \mathbf{x}\|^2 \leq (1-\delta)\|\mathbf{x}\|^2 + s_k^2\sigma_r^2$. Note that only $\mathcal{C}(\mathbf{x}/s_k)$ needs to be transmitted during the communication process, and the recovery of signal is done by computing $\mathcal{Q}(\mathbf{x}) = s_k \mathcal{C}(\mathbf{x}/s_k)$ on the receiver's side. By using the dynamic scaling compression technique, the absolute errors can be controlled by decaying the parameter s_k.

B. A robust compressed push-pull method

We describe the proposed RCPP method in Algorithm 1. Lines 2 and 9 represent the updates for the local decision variables and the gradient trackers, respectively. In Lines 3 and 10, the dynamic scaling compression technique is applied to execute difference compression between the local updates and the auxiliary variables. Difference compression reduces the relative compression errors [16], [21], while the dynamic scaling compression controls the absolute compression errors. More specifically, the operator $\mathcal{Q}$ is a dynamic scaling compressor given by $\mathcal{Q}(\mathbf{x}) = s_k \mathcal{C}(\mathbf{x}/s_k)$. The compressed vector $\mathcal{C}((\widetilde{\mathbf{x}}_i^k - \mathbf{h}_{i,x}^k)/s_k)$ is transmitted to the neighbors of agent i and recovered by computing $s_k \mathcal{C}((\widetilde{\mathbf{x}}_i^k - \mathbf{h}_{i,x}^k)/s_k)$ after communication, where $\widetilde{\mathbf{x}}_i^k$ and $\mathbf{h}_{i,x}^k$ denote agent i's local update and auxiliary variable, respectively. It is worth noting that if the dynamic scaling compression technique is not used, then the absolute compression error would accumulate and significantly impact the algorithm's convergence.

In Lines 4 and 11, the decision variables and the gradient trackers are locally recovered, respectively. Lines 5 and 12 represent the communication steps, where each agent mixes the received compressed vectors multiplied by s_k. The variables $\widehat{\mathbf{X}}_R^k$ and $\widehat{\mathbf{Y}}_C^k$ are introduced to store the aggregated information received from the communication updates. By introducing such auxiliary variables, there is no need to store all the neighbors' reference points [16], [18]. Lines 6-7 and 13-14 update the auxiliary variables, where parameters α_x, α_y control the relative compression errors; see e.g., [21] for reference. The consensus updates are performed in Lines 8 and 15, where γ_x, γ_y are the global consensus parameters to guarantee the algorithmic convergence.

Algorithm 1 A Robust Compressed Push-Pull Method

Input: step-sizes $\Lambda = \operatorname{diag}([\lambda_1, \lambda_2, \ldots, \lambda_n])$, parameters $\alpha_x, \alpha_y, \gamma_x, \gamma_y, \{s_k\}_{k \geq 0}$, initial values $\mathbf{X}^0$, $\mathbf{Y}^0 = \nabla \mathbf{F}(\mathbf{X}^0)$, $\mathbf{H}_x^0 = 0$, $\mathbf{H}_y^0 = 0$, $\mathbf{H}_R^0 = 0$, $\mathbf{H}_C^0 = 0$, number of iterations K

```
1:  for $k = 0, 1, 2, \ldots, K-1$ do
2:      $\widetilde{\mathbf{X}}^k = \mathbf{X}^k - \Lambda \mathbf{Y}^k$
3:      $\mathbf{C}_x^k = \mathcal{C}((\widetilde{\mathbf{X}}^k - \mathbf{H}_x^k)/s_k)$
4:      $\widehat{\mathbf{X}}^k = \mathbf{H}_x^k + \mathbf{Q}_x^k$ [1]
5:      $\widehat{\mathbf{X}}_R^k = \mathbf{H}_R^k + \mathbf{R}\mathbf{Q}_x^k$        ▷ Communication
6:      $\mathbf{H}_x^{k+1} = (1-\alpha_x)\mathbf{H}_x^k + \alpha_x \widehat{\mathbf{X}}^k$
7:      $\mathbf{H}_R^{k+1} = (1-\alpha_x)\mathbf{H}_R^k + \alpha_x \widehat{\mathbf{X}}_R^k$
8:      $\mathbf{X}^{k+1} = \widetilde{\mathbf{X}}^k - \gamma_x(\widehat{\mathbf{X}}^k - \widehat{\mathbf{X}}_R^k)$
9:      $\widetilde{\mathbf{Y}}^k = \mathbf{Y}^k + \nabla \mathbf{F}(\mathbf{X}^{k+1}) - \nabla \mathbf{F}(\mathbf{X}^k)$
10:     $\mathbf{C}_y^k = \mathcal{C}((\widetilde{\mathbf{Y}}^k - \mathbf{H}_y^k)/s_k)$
11:     $\widehat{\mathbf{Y}}^k = \mathbf{H}_y^k + \mathbf{Q}_y^k$
12:     $\widehat{\mathbf{Y}}_C^k = \mathbf{H}_C^k + \mathbf{C}\mathbf{Q}_y^k$        ▷ Communication
13:     $\mathbf{H}_y^{k+1} = (1-\alpha_y)\mathbf{H}_y^k + \alpha_y \widehat{\mathbf{Y}}^k$
14:     $\mathbf{H}_C^{k+1} = (1-\alpha_y)\mathbf{H}_C^k + \alpha_y \widehat{\mathbf{Y}}_C^k$
15:     $\mathbf{Y}^{k+1} = \widetilde{\mathbf{Y}}^k - \gamma_y(\widehat{\mathbf{Y}}^k - \widehat{\mathbf{Y}}_C^k)$
16: end for
```

Output: $\mathbf{X}^K, \mathbf{Y}^K$

To see the connection between RCPP and the Push-Pull/AB algorithm [11], [12], note that we have $\mathbf{H}_R^0 = \mathbf{R}\mathbf{H}_x^0$ and $\mathbf{H}_C^0 = \mathbf{C}\mathbf{H}_y^0$ from the initialization. It follows by induction that $\widehat{\mathbf{X}}_R^k = \mathbf{R}\widehat{\mathbf{X}}^k$, $\widehat{\mathbf{Y}}_C^k = \mathbf{C}\widehat{\mathbf{Y}}^k$ and $\mathbf{H}_R^k = \mathbf{R}\mathbf{H}_x^k$, $\mathbf{H}_C^k = \mathbf{C}\mathbf{H}_y^k$. Recalling Lines 8 and 15 in Algorithm 1, we have

$$\begin{aligned}\mathbf{X}^{k+1} =& \widetilde{\mathbf{X}}^k - \gamma_x(\widehat{\mathbf{X}}^k - \widehat{\mathbf{X}}_R^k) = \widetilde{\mathbf{X}}^k - \gamma_x(\widehat{\mathbf{X}}^k - \mathbf{R}\widehat{\mathbf{X}}^k) \\ =& \widetilde{\mathbf{X}}^k - \gamma_x(\mathbf{I} - \mathbf{R})\widehat{\mathbf{X}}^k, \end{aligned} \tag{6}$$

$$\begin{aligned}\mathbf{Y}^{k+1} =& \widetilde{\mathbf{Y}}^k - \gamma_y(\widehat{\mathbf{Y}}^k - \widehat{\mathbf{Y}}_R^k) = \widetilde{\mathbf{Y}}^k - \gamma_y(\widehat{\mathbf{Y}}^k - \mathbf{C}\widehat{\mathbf{Y}}^k) \\ =& \widetilde{\mathbf{Y}}^k - \gamma_y(\mathbf{I} - \mathbf{C})\widehat{\mathbf{Y}}^k. \end{aligned} \tag{7}$$

If $\widetilde{\mathbf{X}}^k$ and $\widetilde{\mathbf{Y}}^k$ are not compressed, i.e., $\widehat{\mathbf{X}}^k = \widetilde{\mathbf{X}}^k$ and $\widehat{\mathbf{Y}}^k = \widetilde{\mathbf{Y}}^k$, then, $\mathbf{X}^{k+1} = \widetilde{\mathbf{X}}^k - \gamma_x(\mathbf{I} - \mathbf{R})\widetilde{\mathbf{X}}^k = [(1-\gamma_x)\mathbf{I} +$

[1] $\mathbf{Q}_x^k$ is the result of dynamic scaling compression with $\mathbf{Q}_x^k = \mathcal{Q}(\widetilde{\mathbf{X}}^k - \mathbf{H}_x^k) = s_k \mathcal{C}((\widetilde{\mathbf{X}}^k - \mathbf{H}_x^k)/s_k) = s_k \mathbf{C}_x^k$. The operation for $\mathbf{Q}_y^k$ is the same.

$\gamma_x\mathbf{R}](\mathbf{X}^k - \Lambda\mathbf{Y}^k)$, and $\mathbf{Y}^{k+1} = \widetilde{\mathbf{Y}}^k - \gamma_y(\mathbf{I}-\mathbf{C})\widetilde{\mathbf{Y}}^k = [(1-\gamma_y)\mathbf{I} + \gamma_y\mathbf{C}](\mathbf{Y}^k + \nabla\mathbf{F}(\mathbf{X}^{k+1}) - \nabla\mathbf{F}(\mathbf{X}^k))$. Letting the consensus step-sizes be $\gamma_x = 1$ and $\gamma_y = 1$, the above updates recover those in the Push-Pull/AB algorithm [11], [12].

In addition, RCPP retains the property of gradient tracking based methods. From Line 15 in Algorithm 1, $\mathbf{1}^\intercal\mathbf{Y}^{k+1} = \mathbf{1}^\intercal(\widetilde{\mathbf{Y}}^k - \gamma_y(\mathbf{I}-\mathbf{C})\widehat{\mathbf{Y}}^k) = \mathbf{1}^\intercal(\mathbf{Y}^k + \nabla\mathbf{F}(\mathbf{X}^{k+1}) - \nabla\mathbf{F}(\mathbf{X}^k)) = \mathbf{1}^\intercal\nabla\mathbf{F}(\mathbf{X}^{k+1})$, where the second equality is from $\mathbf{1}^\intercal(\mathbf{I}-\mathbf{C}) = \mathbf{0}$, and the last equality is deduced by induction given that $\mathbf{Y}^0 = \nabla\mathbf{F}(\mathbf{X}^0)$. Define $\overline{\mathbf{X}}^k = \frac{1}{n}\mathbf{u}_R^\intercal\mathbf{X}^k$ and $\overline{\mathbf{Y}}^k = \frac{1}{n}\mathbf{1}^\intercal\mathbf{Y}^k$. Once $(\mathbf{x}_i^k)^\intercal \to \overline{\mathbf{X}}^k$ and $(\mathbf{y}_i^k)^\intercal \to \overline{\mathbf{Y}}^k$, then each agent can track the average gradient, i.e., $(\mathbf{y}_i^k)^\intercal \to \overline{\mathbf{Y}}^k = \frac{1}{n}\mathbf{1}^\intercal\nabla\mathbf{F}(\mathbf{X}^k) \to \frac{1}{n}\mathbf{1}^\intercal\nabla\mathbf{F}(\mathbf{1}\overline{\mathbf{X}}^k)$.

IV. Convergence analysis

In this section, we study the convergence property of RCPP under smooth objective functions satisfying the PL condition. For simplicity of notation, denote $\Pi_R = \mathbf{I} - \frac{\mathbf{1}\mathbf{u}_R^\intercal}{n}$, $\Pi_C = \mathbf{I} - \frac{\mathbf{u}_C\mathbf{1}^\intercal}{n}$ and $\mathbf{X}^* = (\mathbf{x}^*)^\intercal \in \mathbb{R}^{1\times p}$. The main idea is to bound the optimization error $\Omega_o^k := \mathbb{E}\big[f(\overline{\mathbf{X}}^k) - f(\mathbf{X}^*)\big]$, consensus error $\Omega_c^k := \mathbb{E}\big[\|\Pi_R\mathbf{X}^k\|_R^2\big]$, gradient tracking error $\Omega_g^k := \mathbb{E}\big[\|\Pi_C\mathbf{Y}^k\|_C^2\big]$, and compression errors $\Omega_{cx}^k := \mathbb{E}\big[\|\widetilde{\mathbf{X}}^k - \mathbf{H}_x^k\|_F^2\big]$ and $\Omega_{cy}^k := \mathbb{E}\big[\|\mathbf{Y}^k - \mathbf{H}_y^k\|_F^2\big]$ through a linear system of inequalities, where $\|\cdot\|_R$ and $\|\cdot\|_C$ are specific norms introduced in Lemma 1. Denote $\bar{\lambda} = \frac{1}{n}\mathbf{u}_R^\intercal\Lambda\mathbf{u}_C$, $\widehat{\lambda} = \max_i\{\lambda_i\}$.

We first introduce two supporting lemmas.

Lemma 1: There exist invertible matrices $\widetilde{\mathbf{R}}, \widetilde{\mathbf{C}}$ such that the induced vector norms $\|\cdot\|_R$ and $\|\cdot\|_C$ satisfy $\|\mathbf{v}\|_R = \|\widetilde{\mathbf{R}}\mathbf{v}\|$ and $\|\mathbf{v}\|_C = \|\widetilde{\mathbf{C}}\mathbf{v}\|$, respectively. Then, for any $\gamma_x, \gamma_y \in (0,1]$, $\|\Pi_R\mathbf{R}_\gamma\|_R \le 1 - \theta_R\gamma_x$ and $\|\Pi_C\mathbf{C}_\gamma\|_C \le 1 - \theta_C\gamma_y$, where $\mathbf{R}_\gamma = \mathbf{I} - \gamma_x(\mathbf{I}-\mathbf{R})$, $\mathbf{C}_\gamma = \mathbf{I} - \gamma_y(\mathbf{I}-\mathbf{C})$, θ_R and θ_C are constants in $(0,1]$. In particular, $\|\Pi_R\|_R = \|\Pi_C\|_C = 1$, $\|\mathbf{R}\|_R \le 2, \|\mathbf{C}\|_C \le 2, \|\mathbf{R}_\gamma\|_R \le 2, \|\mathbf{C}_\gamma\|_C \le 2$ and $\|\mathbf{I}-\mathbf{R}\|_R \le 3, \|\mathbf{I}-\mathbf{C}\|_C \le 3$. In addition, there hold $\|\mathbf{v}\| \le \|\mathbf{v}\|_R, \|\mathbf{v}\| \le \|\mathbf{v}\|_C$ and there exist constants $\delta_{R,2}, \delta_{C,2}$ such that $\|\mathbf{v}\|_R \le \delta_{R,2}\|\mathbf{v}\|, \|\mathbf{v}\|_C \le \delta_{C,2}\|\mathbf{v}\|$.

Proof: See the supplementary material in [20]. ■

Lemma 2: For $\|\mathbf{Y}^k\|_F^2$, we have

$$\|\mathbf{Y}^k\|_F^2 \le 3\|\Pi_C\mathbf{Y}^k\|_C^2 + \frac{3\|\mathbf{u}_C\|^2}{n}L^2\|\Pi_R\mathbf{X}^k\|_R^2 + 3\|\mathbf{u}_C\|^2\left\|\nabla f(\overline{\mathbf{X}}^k)\right\|^2. \tag{8}$$

Proof: Reformulating $\mathbf{Y}^k$, we get $\|\mathbf{Y}^k\|_F^2 = \|\mathbf{Y}^k - \mathbf{u}_C\overline{\mathbf{Y}}^k + \mathbf{u}_C(\overline{\mathbf{Y}}^k - \nabla f(\overline{\mathbf{X}}^k)) + \mathbf{u}_C\nabla f(\overline{\mathbf{X}}^k)\|_F^2 \le 3\|\mathbf{Y}^k - \mathbf{u}_C\overline{\mathbf{Y}}^k\|_F^2 + 3\|\mathbf{u}_C(\overline{\mathbf{Y}}^k - \nabla f(\overline{\mathbf{X}}^k))\|_F^2 + 3\|\mathbf{u}_C\nabla f(\overline{\mathbf{X}}^k)\|_F^2$, where the last inequality is based on Lemma 1. Based on Lemma 1 and $\|\nabla f(\overline{\mathbf{X}}^k) - \overline{\mathbf{Y}}^k\| \le \frac{L}{\sqrt{n}}\|\mathbf{X}^k - \mathbf{1}\overline{\mathbf{X}}^k\|_F \le \frac{L}{\sqrt{n}}\|\Pi_R\mathbf{X}^k\|_R$, the relation (8) is derived. ■

Next, we present the key linear system of inequalities.

Lemma 3: Suppose Assumptions 1, 2 and 3 hold and $\widehat{\lambda} \le \min\left\{\frac{1}{6L}, \frac{1}{6\sqrt{C}L}\right\}$. Then we have

$$\mathbf{w}^{k+1} \le \mathbf{A}\mathbf{w}^k + \mathbf{b}\left\|\mathbf{Y}^k\right\|_F^2 + \boldsymbol{\zeta}^k, \tag{9}$$

where $\mathbf{w}^k = \left[\Omega_c^k, \Omega_g^k, \Omega_{cx}^k, \Omega_{cy}^k\right]^\intercal$, $\boldsymbol{\zeta}^k = s_k^2 \cdot \left[\zeta_c, \zeta_g, \zeta_{cx}, \zeta_{cy}\right]^\intercal$, $\mathbf{A}_{1,:} = \left[1 - \theta_R\gamma_x, 0, \frac{18\delta_{R,2}^2C\gamma_x^2}{\theta_R\gamma_x}, 0\right]$, $\mathbf{A}_{2,:} = \left[\frac{27e_1L^2\gamma_x^2}{\theta_C\gamma_y}, 1 - \theta_C\gamma_y, \frac{27e_1CL^2\gamma_x^2}{\theta_C\gamma_y}, \frac{36\delta_{C,2}^2C\gamma_y^2}{\theta_C\gamma_y}\right]$, $\mathbf{A}_{3,:} = \left[\frac{324\gamma_x^2}{\alpha_xr\delta}, \frac{162\gamma_y^2\widehat{\lambda}^2}{\alpha_xr\delta}, 1 - \frac{\alpha_xr\delta}{2} + \frac{324C\gamma_x^2}{\alpha_xr\delta}, \frac{324C\gamma_y^2\widehat{\lambda}^2}{\alpha_xr\delta}\right]$, $\mathbf{A}_{4,:} = \left[\frac{27e_2L^2\gamma_x^2}{\alpha_yr\delta}, \frac{108\gamma_y^2}{\alpha_yr\delta}, \frac{27e_2CL^2\gamma_x^2}{\alpha_yr\delta}, 1 - \frac{\alpha_yr\delta}{4} + \frac{108C\gamma_y^2}{\alpha_yr\delta}\right]$, and $\mathbf{b} = \left[\frac{\delta_{R,2}^2(1-\theta_R\gamma_x)^2\widehat{\lambda}^2}{\theta_R\gamma_x}, \frac{12e_1L^2\widehat{\lambda}^2}{\theta_C\gamma_y}, \frac{48\widehat{\lambda}^2}{\alpha_xr\delta}, \frac{12e_2L^2\widehat{\lambda}^2}{\alpha_yr\delta}\right]^\intercal$.

The constants $\zeta_c, \zeta_g, \zeta_{cx}, \zeta_{cy}$ can be found in the extended version [30].

Proof: See Appendix D in the extended version [30]. ■

The following descent lemma results from the smoothness of the gradients and will be used for proving the main theorem together with Lemma 3.

Lemma 4: Suppose Assumption 2 holds, $\bar{\lambda} \le \frac{1}{L}$ and $\bar{\lambda} \ge M\widehat{\lambda}$ for some $M > 0$. We have

$$f(\overline{\mathbf{X}}^{k+1}) \le f(\overline{\mathbf{X}}^k) - \frac{M\widehat{\lambda}}{2}\left\|\nabla f(\overline{\mathbf{X}}^k)\right\|^2 + E_1M\widehat{\lambda}L^2\left\|\Pi_R\mathbf{X}^k\right\|_R^2 + E_2M\widehat{\lambda}\left\|\Pi_C\mathbf{Y}^k\right\|_C^2, \tag{10}$$

where $E_1 = \frac{\|\mathbf{u}_R\|\|\mathbf{u}_C\|}{n^2M}$ and $E_2 = \frac{\|\mathbf{u}_R\|^2}{n^2M^2}$.

Proof: From Assumption 2, the gradient of f is L-Lipschitz continuous. In addition, we know $\overline{\mathbf{X}}^{k+1} = \overline{\mathbf{X}}^k - \frac{1}{n}\mathbf{u}_R^\intercal\Lambda\mathbf{Y}^k$. Then, we have

$$\begin{aligned}
f(\overline{\mathbf{X}}^{k+1}) &\le f(\overline{\mathbf{X}}^k) + \langle\nabla f(\overline{\mathbf{X}}^k), \overline{\mathbf{X}}^{k+1} - \overline{\mathbf{X}}^k\rangle + \frac{L}{2}\left\|\overline{\mathbf{X}}^{k+1} - \overline{\mathbf{X}}^k\right\|^2 \\
&= f(\overline{\mathbf{X}}^k) - \bar{\lambda}\langle\nabla f(\overline{\mathbf{X}}^k), \frac{1}{n\bar{\lambda}}\mathbf{u}_R^\intercal\Lambda\mathbf{Y}^k\rangle + \frac{L\bar{\lambda}^2}{2}\left\|\frac{1}{n\bar{\lambda}}\mathbf{u}_R^\intercal\Lambda\mathbf{Y}^k\right\|^2 \\
&= f(\overline{\mathbf{X}}^k) - \frac{\bar{\lambda}}{2}\Bigg(\left\|\nabla f(\overline{\mathbf{X}}^k)\right\|^2 + \left\|\frac{1}{n\bar{\lambda}}\mathbf{u}_R^\intercal\Lambda\mathbf{Y}^k\right\|^2 \\
&\quad - \left\|\nabla f(\overline{\mathbf{X}}^k) - \frac{1}{n\bar{\lambda}}\mathbf{u}_R^\intercal\Lambda\mathbf{Y}^k\right\|^2\Bigg) + \frac{L\bar{\lambda}^2}{2}\left\|\frac{1}{n\bar{\lambda}}\mathbf{u}_R^\intercal\Lambda\mathbf{Y}^k\right\|^2,
\end{aligned}$$

where the second equality is based on $\langle\mathbf{a},\mathbf{b}\rangle = \frac{1}{2}(\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - \|\mathbf{a}-\mathbf{b}\|^2)$. Note that $\bar{\lambda} \le \frac{1}{L}$ from the assumption in Lemma 4, we know $\frac{L\bar{\lambda}^2}{2}\left\|\frac{1}{n\bar{\lambda}}\mathbf{u}_R^\intercal\Lambda\mathbf{Y}^k\right\|^2 \le \frac{\bar{\lambda}}{2}\left\|\frac{1}{n\bar{\lambda}}\mathbf{u}_R^\intercal\Lambda\mathbf{Y}^k\right\|^2$. Meanwhile, $\nabla f(\overline{\mathbf{X}}^k) - \frac{1}{n\bar{\lambda}}\mathbf{u}_R^\intercal\Lambda\mathbf{Y}^k = \nabla f(\overline{\mathbf{X}}^k) - \overline{\mathbf{Y}}^k - \frac{1}{n\bar{\lambda}}\mathbf{u}_R^\intercal\Lambda\Pi_C\mathbf{Y}^k$. Thus, we have

$$\begin{aligned}
f(\overline{\mathbf{X}}^{k+1}) &\le f(\overline{\mathbf{X}}^k) - \frac{\bar{\lambda}}{2}\left\|\nabla f(\overline{\mathbf{X}}^k)\right\|^2 \\
&\quad + \frac{\bar{\lambda}}{2}\left\|\nabla f(\overline{\mathbf{X}}^k) - \overline{\mathbf{Y}}^k - \frac{1}{n\bar{\lambda}}\mathbf{u}_R^\intercal\Lambda\Pi_C\mathbf{Y}^k\right\|^2 \\
&\le f(\overline{\mathbf{X}}^k) - \frac{\bar{\lambda}}{2}\left\|\nabla f(\overline{\mathbf{X}}^k)\right\|^2 + \frac{L^2\bar{\lambda}}{n}\left\|\Pi_R\mathbf{X}^k\right\|_R^2 \\
&\quad + \frac{\|\mathbf{u}_R\|^2\widehat{\lambda}^2}{n^2\bar{\lambda}}\left\|\Pi_C\mathbf{Y}^k\right\|_C^2
\end{aligned}$$

where we use $\|\mathbf{a}+\mathbf{b}\|^2 \le 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$ $(\forall\mathbf{a},\mathbf{b})$ in the second inequality. Note that $\bar{\lambda} \ge M\widehat{\lambda}$ and $\bar{\lambda} = \frac{1}{n}\mathbf{u}_R^\intercal\Lambda\mathbf{u}_C \le \frac{1}{n}\mathbf{u}_R^\intercal\mathbf{u}_C\widehat{\lambda} \le \frac{1}{n}\|\mathbf{u}_R\|\|\mathbf{u}_C\|\widehat{\lambda}$. Then, we complete the proof. ■

Based on the above lemmas, we demonstrate the linear convergence rate of RCPP for minimizing smooth objective functions satisfying the PL condition in the following theorem.

Theorem 1: Suppose Assumptions 1, 2 and 3 hold, the scaling parameters $\alpha_x, \alpha_y \in (0, \frac{1}{r}]$, $\bar{\lambda} \geq M\hat{\lambda}$ for some $M > 0$, and the consensus step-sizes γ_x, γ_y and the maximum step-size $\widehat{\lambda}$ satisfy

$$\widehat{\lambda} \leq \left\{\frac{1}{6}, \frac{1}{6\sqrt{C}}, \frac{1}{M}, \frac{\theta_C}{\sqrt{54e_1}}, \frac{e_3\gamma_x}{\sqrt{96E\left\|\mathbf{u}_C\right\|}}, \frac{e_4\theta_C\gamma_y}{\sqrt{48e_1}}\right\}\frac{1}{L},$$

$$\gamma_x \leq \left\{1, \frac{e_5\alpha_x r\delta}{\sqrt{C}}, \frac{\sqrt{M\left\|\mathbf{u}_C\right\|}}{\sqrt{108\left\|\mathbf{u}_R\right\| e_1}}\theta_C\gamma_y\right\},$$

$$\gamma_y \leq \left\{1, \frac{\theta_C e_2}{432e_1}, \frac{\theta_C(\alpha_y r\delta)^2}{432e_2C}, \frac{\alpha_y r\delta}{\sqrt{1728C}}\right\},$$

Then, the optimization error Ω_o^k and the consensus error Ω_c^k both converge to 0 at the linear rate $\mathcal{O}(c^k)$, where $c \in (\tilde{\rho}, 1)$ with $\tilde{\rho} = \max\{1 - \frac{1}{2}M\widehat{\lambda}\mu, 1 - \frac{\theta_R\gamma_x}{16}, 1 - \frac{\theta_C\gamma_y}{8}, 1 - \frac{\alpha_x r\delta}{4}, 1 - \frac{\alpha_y r\delta}{16}\}$, $s_k^2 = c_0c^k$, $E = \frac{\|\mathbf{u}_R\|\|\mathbf{u}_C\|}{n^2M}$, $e_1 = 38\delta_{C,2}^2$, $e_2 = 108C{+}112$, $e_3 = \min\left\{\frac{1}{2\delta_{R,2}}\frac{\theta_R}{1-\theta_R\gamma_x}, 1\right\}$, $e_4 = \min\left\{\frac{\theta_R}{36(1-\theta_R\gamma_x)}, \frac{1}{3\sqrt{3}}\right\}$, $e_5 = \min\left\{\frac{\theta_R}{432\sqrt{2}\delta_{R,2}}, \frac{1}{72}\right\}$.

Proof: Let $V^k = L^2\Omega_c^k + A\Omega_g^k + B\Omega_{cx}^k + D\Omega_{cy}^k$, where $A = \frac{\theta_C\gamma_y\theta_R}{108e_1\gamma_x}$, $B = \frac{L^2\alpha_x r\delta\theta_R}{1296\gamma_x}$, $D = \frac{\theta_C\gamma_y\alpha_y r\delta\theta_R}{108e_2\gamma_x} \leq \frac{\alpha_y r\delta\theta_R}{108e_2\gamma_x}$. Combining Lemmas 2 and 3 with the conditions on $\gamma_x, \gamma_y, \widehat{\lambda}$, we have

$$\begin{aligned} V^{k+1} \leq &\left(1 - \frac{7\theta_R\gamma_x}{32}\right)L^2\Omega_c^k + \left(1 - \frac{\theta_C\gamma_y}{4}\right)A\Omega_g^k \\ &+ \left(1 - \frac{\alpha_x r\delta}{4}\right)B\Omega_{cx}^k + \left(1 - \frac{\alpha_y r\delta}{16}\right)D\Omega_{cy}^k \\ &+ \frac{\beta}{4E}\left\|\nabla f(\overline{\mathbf{X}}^k)\right\|^2 + s_k^2\zeta_0, \end{aligned}$$

where $\beta = \frac{\theta_R\gamma_x}{8}$ and $E = \frac{\|\mathbf{u}_R\|\|\mathbf{u}_C\|}{n^2M}$. Combining Lemma 4, we obtain

$$\begin{aligned} \Omega_o^{k+1} + \frac{EM\widehat{\lambda}}{\beta}V^{k+1} \leq &\ \Omega_o^k - \frac{M\widehat{\lambda}}{4}\left\|\nabla f(\overline{\mathbf{X}}^k)\right\|^2 \\ &+ \frac{EM\widehat{\lambda}}{\beta}\bigg[\Big(1 - \frac{3\theta_R\gamma_x}{32}\Big)L^2\Omega_c^k + \Big(1 - \frac{\theta_C\gamma_y}{8}\Big)A\Omega_g^k \\ &+ \Big(1 - \frac{\alpha_x r\delta}{4}\Big)B\Omega_{cx}^k + \Big(1 - \frac{\alpha_y r\delta}{16}\Big)D\Omega_{cy}^k\bigg] + s_k^2\tilde{\zeta}_0, \end{aligned}$$

where $\tilde{\zeta_0} = \zeta_0\frac{EM\widehat{\lambda}}{\beta}$ and we use the fact $\gamma_x \leq \frac{\sqrt{M\|\mathbf{u}_C\|}}{\sqrt{108\|\mathbf{u}_R\|e_1}}\theta_C\gamma_y$. Recalling the PL condition, we obtain $-\frac{M\widehat{\lambda}}{4}\left\|\nabla f(\overline{\mathbf{X}}^k)\right\|^2 \leq -\frac{M\widehat{\lambda}\mu}{2}(f(\overline{\mathbf{X}}^k) - f(\mathbf{X}^*))$. Thus, we have

$$\begin{aligned} &\Omega_o^{k+1} + \frac{EM\widehat{\lambda}}{\beta}V^{k+1} \leq \tilde{\rho}(\Omega_o^k + \frac{EM\widehat{\lambda}}{\beta}V^k) + s_k^2\tilde{\zeta}_0 \\ \leq &\tilde{\rho}^{k+1}(\Omega_o^0 + \frac{EM\widehat{\lambda}}{\beta}V^0) + \sum_{l=0}^{k}\tilde{\rho}^{k-l}c^l\Theta. \end{aligned}$$

where $\Theta = c_0\tilde{\zeta_0}$. Furthermore, we obtain

$$\begin{aligned} &\Omega_o^{k+1} + \frac{EM\widehat{\lambda}}{\beta}V^{k+1} \\ \leq &\tilde{\rho}^{k+1}(\Omega_o^0 + \frac{EM\widehat{\lambda}}{\beta}V^0) + c^k\Theta\sum_{l=0}^{k}\left(\frac{\tilde{\rho}}{c}\right)^{k-l} \\ \leq &\tilde{\rho}^{k+1}(\Omega_o^0 + \frac{EM\widehat{\lambda}}{\beta}V^0) + c^{k+1}\frac{\Theta}{c - \tilde{\rho}}. \end{aligned}$$

■

Remark 3: It is worth nothing that the linear convergence of RCPP does not depend on the decaying assumption of the absolute compression error as in [25].

V. Numerical Examples

In this section, we provide some numerical results to confirm the theoretical findings. Consider the following ridge regression problem,

$$\min_{x\in\mathbb{R}^p} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)\left(= (u_i^\intercal x - v_i)^2 + \rho\|x\|^2\right), \quad (11)$$

where $\rho > 0$ is a penalty parameter. The pair (u_i, v_i) is a sample data that belongs to the i-th agent, where $u_i \in \mathbb{R}^p$ represents the features, and $v_i \in \mathbb{R}$ represents the observations or outputs. These parameter settings are the same as in [21].

We compare RCPP wih CPP [20] and QDGT [22] for decentralized optimization over a directed graph. The row-stochastic and column-stochastic weights are randomly generated. Regarding the compressor, we consider an adaptation from the b-bits ∞-norm quantization compression method in [18], stated below:

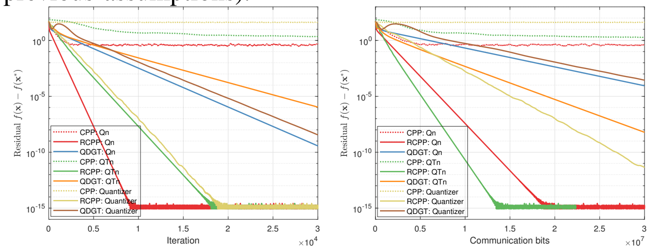
$$\mathcal{Q}(\mathbf{x}) = \frac{h(\|\mathbf{x}\|_\infty)}{2^{b-1}}\mathrm{sign}(\mathbf{x}) \odot \left\lfloor \frac{2^{b-1}|\mathbf{x}|}{\|\mathbf{x}\|_\infty} + \mathbf{u}\right\rfloor, \quad (12)$$

where $\mathrm{sign}(\mathbf{x})$ is the sign function, $\odot$ is the Hadamard product, $|\mathbf{x}|$ is the element-wise absolute value of $\mathbf{x}$, and $\mathbf{u}$ is a random perturbation vector uniformly distributed in $[0, 1]^p$.

Compared with the original compressor in [18] which computes $\|\mathbf{x}\|_\infty$, the new operator uses the mapping $h(\|\mathbf{x}\|)$ which is a random variable such that $h(\|\mathbf{x}\|) = \lfloor\|\mathbf{x}\|\rfloor + 1$ with probability $\|\mathbf{x}\| - \lfloor\|\mathbf{x}\|\rfloor$ and $h(\|\mathbf{x}\|) = \lfloor\|\mathbf{x}\|\rfloor$ otherwise. By considering $h(\|\mathbf{x}\|)$, only dynamic finite bits, i.e., about $\log_2(\lfloor\|\mathbf{x}\|\rfloor + 1) + 1$ bits, need to be transmitted for nonzero norms. The quantization with the new mapping $h(\|\mathbf{x}\|)$ is abbreviated as Qn, and QTn denotes the composition of quantization and Top-k compressor with the same operation. Note that these compression operators produce absolute compression errors and satisfy Assumption 3, and QTn does not satisfy the previous conditions on the compression operators. In the simulation we let $b = 2$ and $\mathrm{k} = 10$. In addition to Qn and QTn, we also consider the quantizer compression in [22], [23] which satisfies the absolute compression error assumption. The quantized level is 1, i.e., the quantized values belong to $\{-1, 0, 1\}$.

In Fig. 1(a), we compare the residuals of CPP, RCPP and QDGT against the number of iterations. It can be seen that the performance of CPP deteriorates due to the absolute compression error. Meanwhile, RCPP outperforms QDGT under different compression methods.

From Fig. 1(b) where we further compare the performance of the algorithms against the communication bits, we find that RCPP performs better than the other methods under different compression methods. Besides, RCPP with QTn achieves the best communication efficiency. This implies that by considering Assumption 3 which provides us with more choices on the compression operators, RCPP may achieve better performance under a specific choice of compression method with less communication (which may not satisfy the previous assumptions).



(a)

(b)

Fig. 1. Residuals $\mathbb{E}\big[f(\overline{\mathbf{x}}^k) - f(\mathbf{x}^*)\big]$ against the number of iterations and communication bits respectively for CPP, RCPP and QDGT under different compression methods.

VI. Conclusions

This article considers decentralized optimization with communication compression over directed networks. Specifically, we consider a general class of compression operators that allow both relative and absolute compression errors. For smooth objective functions satisfying the PL condition, we propose a robust compressed push-pull algorithm, termed RCPP, which converges linearly. Numerical results demonstrate that RCPP is efficient and robust to various compressors.

References

[1] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

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

[3] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning—part I: Algorithm development," *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 708–723, 2019.

[4] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in *Proceedings of the 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 2055–2060.

[5] P. Di Lorenzo and G. Scutari, "NEXT: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[6] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[7] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, Sept. 2018.

[8] Y. Sun, G. Scutari, and A. Daneshmand, "Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation," *SIAM Journal on Optimization*, vol. 32, no. 2, pp. 354–385, 2022.

[9] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 5453–5458.

[10] C. Xi and U. A. Khan, "DEXTRA: A fast algorithm for optimization over directed graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4980–4993, 2017.

[11] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 315–320, 2018.

[12] S. Pu, W. Shi, J. Xu, and A. Nedić, "Push–Pull gradient methods for distributed optimization in networks," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 1–16, Jan. 2021.

[13] R. Xin, S. Pu, A. Nedić, and U. A. Khan, "A general framework for decentralized optimization with first-order methods," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 1869–1889, 2020.

[14] S. Pu, "A robust gradient tracking method for distributed optimization over directed networks," in *Proceedings of the 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 2335–2341.

[15] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *Advances in Neural Information Processing Systems*, 2018, pp. 7652–7662.

[16] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, pp. 3479–3487.

[17] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," in *International Conference on Learning Representations*, 2020.

[18] X. Liu, Y. Li, R. Wang, J. Tang, and M. Yan, "Linear convergent decentralized optimization with compression," in *International Conference on Learning Representations*, 2020.

[19] D. Kovalev, A. Koloskova, M. Jaggi, P. Richtarik, and S. Stich, "A linearly convergent algorithm for decentralized optimization: Sending less bits for free!" in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 4087–4095.

[20] Z. Song, L. Shi, S. Pu, and M. Yan, "Compressed gradient tracking for decentralized optimization over general directed networks," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1775–1787, 2022.

[21] Y. Liao, Z. Li, K. Huang, and S. Pu, "A compressed gradient tracking method for decentralized optimization with linear convergence," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5622–5629, 2022.

[22] Y. Xiong, L. Wu, K. You, and L. Xie, "Quantized distributed gradient tracking algorithm with linear convergence in directed networks," *IEEE Transactions on Automatic Control*, 2022.

[23] Y. Kajiyama, N. Hayashi, and S. Takai, "Linear convergence of consensus-based quantized optimization for smooth and strongly convex cost functions," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 1254–1261, 2021.

[24] S. Magnússon, H. Shokri-Ghadikolaei, and N. Li, "On maintaining linear convergence of distributed learning and optimization under limited communication," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6101–6116, 2020.

[25] N. Michelusi, G. Scutari, and C.-S. Lee, "Finite-bit quantization for distributed algorithms with linear convergence," *IEEE Transactions on Information Theory*, vol. 68, no. 11, pp. 7254–7280, 2022.

[26] R. Nassif, S. Vlaski, M. Carpentiero, V. Matta, M. Antonini, and A. H. Sayed, "Quantization for decentralized learning under subspace constraints," *arXiv preprint arXiv:2209.07821*, 2022.

[27] J. Zhang, K. You, and L. Xie, "Innovation compression for communication-efficient distributed optimization with linear convergence," *IEEE Transactions on Automatic Control*, pp. 1–8, 2023.

[28] C.-Y. Yau and H.-T. Wai, "DoCoM-SGT: Doubly compressed momentum-assisted stochastic gradient tracking algorithm for communication efficient decentralized learning," *arXiv preprint arXiv:2202.00255*, 2022.

[29] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "Communication compression for distributed nonconvex optimization," *IEEE Transactions on Automatic Control*, pp. 1–16, 2022.

[30] Y. Liao, Z. Li, and S. Pu, "A linearly convergent robust compressed push-pull method for decentralized optimization," *arXiv preprint arXiv:2303.07091*, 2023.