CPCA: A Chebyshev Proxy and Consensus based Algorithm for General Distributed Optimization

Zhiyu He, Jianping He, Cailian Chen and Xinping Guan

Abstract—We consider a general distributed optimization problem, aiming to optimize the average of a set of local objectives that are Lipschitz continuous univariate functions, with the existence of same local constraint sets. To solve the problem, we propose a Chebyshev Proxy and Consensus-based Algorithm (CPCA). Compared with existing distributed optimization algorithms, CPCA is able to address the problem with non-convex Lipschitz objectives, and has low computational costs since it is free from gradient or projection calculations. These benefits result from i) the idea of optimizing a Chebyshev polynomial approximation (i.e. a proxy) for the global objective to obtain $\varepsilon$-optimal solutions for any given precision $\varepsilon$, and ii) the use of average consensus where the local proxies’ coefficient vectors are gossiped to enable every agent to obtain such a global proxy. We provide comprehensive analysis of the accuracy and complexities of the proposed algorithm. Simulations are conducted to illustrate its effectiveness.

I. INTRODUCTION

Distributed optimization algorithms enable multiple agents in a network to collaboratively solve the problem of optimizing the average of local objective functions. Their developments have been motivated by wide application scenarios, including distributed machine learning [1], statistics [2], estimation [3] and coordination [4]–[6] in large-scale multi-agent systems, like wireless sensor networks, smart grids, multi-robot systems and so on.

Most distributed optimization algorithms are first-order methods based on gradients, and can be divided into three categories: primal, dual and primal-dual methods. They all employ consensus to make all agents’ local estimates be close after iterations. Primal methods [7]–[10] use subgradient descent to drive the estimates to the optimal points in the primal domain. Dual methods [11], [12] consider the dual of the problem with consensus equality constraints and go on iterations in the dual domain. Primal-dual methods [13]–[15] update primal and dual variables in parallel, so as to reach the saddle points of the Lagrangian. When there also exist local set constraints, projection operations onto the constraint sets can be taken to extend existing unconstrained methods [16], [17].

However, existing iterative gradient-based algorithms generally work under the premise that the objective functions are convex, and involve computationally expensive operations at every iteration. They need convexity assumptions either to guarantee the reach of the global rather than local optimizer, or to make sure the hold of strong duality. Since they require every agent at every iteration to compute gradients or optimize a local sub-problem, which can be costly in general, the total computational costs are rather high when the iterations go long. Moreover, constrained methods have extra costs of calculating projections and still suffer from sub-linear convergence rates. It still remains an open problem how to design algorithms that can handle problems with non-convex objectives and are computationally inexpensive.

Recently, there have been works in the numerical analysis field that use the Chebyshev polynomial approximation to substitute for the target function defined on an interval, so as to make the study of its property much easier [18]–[20]. Since an arbitrarily precise approximation (i.e. a proxy) can be constructed for any continuous objective on the entire interval, the difference between their optimal values can be arbitrarily small. This means that we can turn to solve the easier problem of optimizing the proxy of the global objective instead. In addition, the coefficient vector of the proxy serves as a discrete representation of it. This implies that by going on average consensus, agents can obtain the average of all those vectors of the local proxies. This average is exactly the representation of the proxy of the global objective. Agents can then compute the optimum of the polynomial recovered from this vector by evaluating it on all its critical points, and thus obtain close estimates for that of the global objective.

Based on these intuitions, we develop a Chebyshev Proxy and Consensus-based Algorithm (CPCA) for the constrained distributed optimization problem. This problem has same local constraint sets (the extension to the case with different sets is not difficult) and general Lipschitz continuous univariate objectives. The main contributions are summarized as follows:

1) To the best of our knowledge, this is the first work to achieve global optimum for constrained distributed optimization problems without convexity assumptions on the objectives.

2) We propose a novel algorithm, CPCA, based on Chebyshev polynomial approximation and consensus. It jumps out of the scope of iterative gradient- and projection-based methods, and solves an easier problem of optimizing the polynomial approximation for the global objective instead. Also, it has a simple iterating structure of processing coefficient vectors, rather than gossiping estimates of the optimizers followed by additional operations.

The authors are with the Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China. Emails: {hzy970920, jphe, cailianchen, xguan}@sjtu.edu.cn. This research work is partially sponsored by the National Key R&D Program of China 2017YFE0114600, and NSFC of China (61973218, 61828301, 61622307).
3) We provide comprehensive analysis of the accuracy and the complexities of CPCA. Specifically, for any given error tolerance \( \epsilon \), the solutions it yields are \( \epsilon \)-optimal. The communication and zero-order oracle complexities for every agent are \( O(N \log(N \log m / \epsilon)) \) and \( O(m) \) respectively, where \( N \) is the number of agents and \( m \) is the degree of the approximation. Note that our algorithm is free from gradient computations, and has oracle complexities independent of \( N \) and \( \epsilon \). This shows that our algorithm are computationally inexpensive.

The rest of this paper is organized as follows. Section II provides some preliminaries and formally defines the problem of interest. Section III presents our algorithm, namely CPCA. Section IV provides the analysis of the accuracy and complexities of CPCA. Section V shows the simulation results. Finally, Section VI concludes this paper.

**Notations.** For vector \( a \), we use \( a' \) to denote its transpose, \( \| a \| \) to denote its \( l_2 \)-norm, and \( \| a \|_{\infty} \) to denote its \( l_{\infty} \)-norm. We denote the all-ones vector by \( \mathbf{1} \). The superscript \( t \) denotes the index of the iterations, and the subscripts \( i \), \( j \) denote the indexes of the agents. The scripts in parentheses \( k, l \) denote the indexes of the elements in a vector.

**II. Preliminaries and Problem Formulation**

Consider a network with \( N \) agents. Its communication topology is described as a graph \( G = (V, E) \), where \( V \) is the set of agents, and \( E \subseteq V \times V \) is the set of edges. It is noted that agent \( j \) can receive information from agent \( i \) if and only if \((i,j) \in E\). Throughout the paper, we assume that \( G \) is a static, connected and undirected graph.

**A. Typical Types of Consensus**

Suppose every agent \( i \) maintains a local variable \( x_i^t \in \mathbb{R} \), where \( t \in \mathbb{N} \) is the number of iterations. The maximum consensus protocol is

\[
x_i^t = \max \{ x_i^{t-1}, \max_{j \in N_i} x_j^{t-1} \},
\]

where \( N_i \) denotes the set of agent \( i \)'s neighbors. By (1), all \( x_i^t \) converge to \( \max_{i \in V} x_i^0 \) in \( T \) (\( T \leq D \)) iterations, where \( D \) is the diameter of \( G \).

Another type of consensus is average consensus. There have already been such protocols, e.g., [21], whose convergence time scale quadratically with \( N \). Recently, [22] shows that the order of convergence in terms of \( N \) can be brought down to linear with the following assumption.

**Assumption 1** ([22]). Every agent \( i \) in \( G \) knows an upper bound \( U \) on \( N \), such that \( \exists \ c \in \mathbb{R} : N \leq U \leq cN \).

The linear time average consensus protocol [22] is

\[
\begin{align*}
    y_i^t &= x_i^{t-1} + \frac{1}{2} \sum_{j \in N_i} \max(d_i, d_j) x_j^{t-1} - x_i^{t-1}, \\
    x_i^t &= y_i^t + \left( 1 - \frac{2}{9U + 1} \right) (y_i^t - y_i^{t-1}),
\end{align*}
\]

where \( y_i^0 \) is initialized to be \( x_i^0 \), and \( d_i \) is the degree of node \( i \) in \( G \). Let \( x^t = [x_1^t, \ldots, x_N^t]' \), \( y^t = [y_1^t, \ldots, y_N^t]' \). The convergence of (2) is discussed in the following theorem.

**Theorem 1** ([22]). If Assumption 1 holds, with (2), we have

\[
\|y^t - x^t\| \leq \sqrt{2} \rho^t \|y^0 - x^t\|,
\]

where \( x = 1/N \sum_{i=1}^N x_i^0 \), and \( \rho = \sqrt{1 - 1/(9U)} \).

Let \( e^t = y^t - x^t \). We derive the decaying property of \( \max_{k,l} |e^t(k) - e^t(l)| \), where \( k \) and \( l \) are the indexes of the elements in \( e^t \).

**Corollary 1.** If Assumption 1 holds, with (2), we have

\[
\max_{k,l} |e^t(k) - e^t(l)| \leq 2 \sqrt{2N} \rho^t \max_{k,l} |e^0(k) - e^0(l)|.
\]

**Proof.** By rewriting (2) as matrix equations and noting that the weight matrix is doubly stochastic, we have \( 1' \gamma^t = 1' x^0 \), \( \forall t \in \mathbb{N} \). Hence, \( 1' e^t = 0 \), \( \forall t \in \mathbb{N} \]. As a result, \( \|e^t\|_{\infty} \leq \max_{k,l} |e^t(k) - e^t(l)| \leq 2 \|e^0\|_{\infty} \). Therefore,

\[
\max_{k,l} |e^t(k) - e^t(l)| \leq 2 \|e^0\|_{\infty} \leq 2 \|e^0\|_{2} \leq 2 \sqrt{2} \rho^t \|e^0\|_{2} \leq 2 \sqrt{2N} \rho^t \max_{k,l} |e^0(k) - e^0(l)|.
\]

**B. Chebyshev Polynomial Approximation**

For a Lipschitz continuous function \( g(x) \) with \( x \in [a, b] \), its degree \( m \) Chebyshev interpolant \( p_m(x) \) is

\[
p_m(x) = \sum_{j=0}^m c_j T_j \left( \frac{2x - (a + b)}{b - a} \right), \quad x \in [a, b],
\]

where \( T_j(\cdot) \) denotes the \( j \)-th Chebyshev polynomial. As \( m \) grows, \( p_m(x) \) converges to \( g(x) \) uniformly on the given interval [18], i.e.,

\[
\forall x \in [a, b], \quad |p_m(x) - g(x)| \to 0, \quad m \to \infty.
\]

Let \( c_j' \) be the Chebyshev coefficients of the derivative of \( p_m(x) \). Then, we have the following recurrence formula,

\[
c_j' = \begin{cases} 
0, & j = m, m + 1, \ldots \\
2j + 1 + S c_{j+1}, & j = m - 1, \ldots, 1 \\
2j + S c_1, & j = 0,
\end{cases}
\]

where \( S = 2/(b - a) \).

By [18], the roots of \( p_m(x) \) are the eigenvalues of a colleague matrix \( M_C \), which is given by

\[
\begin{pmatrix}
    0 & 1 & 0 & \cdots & 0 \\
    1 & 0 & 1 & \cdots & 0 \\
    \vdots & \ddots & \ddots & \ddots & \vdots \\
    0 & \cdots & \cdots & \cdots & 0 \\
    \frac{c_m}{2^{m-1}} & \frac{c_m}{2^{m-1}} & \cdots & \frac{c_m}{2^{m-1}} & 0 \\
\end{pmatrix}
\]

Clearly, \( M_C \) is a sparse matrix whose non-zero elements are trivial functions of the Chebyshev coefficients of \( p_m(x) \).

\[1\text{A detailed proof can be found in the proof of Theorem 2.}\]
C. Problem Formulation

In this paper, we mainly focus on the following constrained distributed optimization problem

\[
\min_x f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x),
\]

s.t. \( x \in X = \bigcap_{i=1}^{N} X_i, \)

where \( f_i(x) : X_i \to \mathbb{R} \) is the local objective function and is known only by agent \( i \), and \( X_i \subseteq \mathbb{R} \) is the local constraint set. Some basic assumptions are given as follows.

Assumption 2. Every \( f_i(x) \) is Lipschitz continuous on \( X_i \).

The Lipschitz continuous assumption on local objective functions is quite common among the literature [12], [13]. What distinguishes our work from others is that we require neither global nor local objectives to be convex.

Assumption 3. All \( X_i \) are the same closed, bounded and convex set.

By Assumption 3, for all \( i \in \mathcal{V}, X_i \) is the closed interval \([a, b]\), where \( a, b \in \mathbb{R} \). We make this simplification to highlight the design and the structure of our algorithm.

III. ALGORITHM DEVELOPMENT

In this section, we present CPCA to solve problem (6) distributively. Figure 1 provides an overview of our algorithm.

A. Initialization: Construction of Approximations

In the initialization stage, every agent first constructs the approximating polynomial, \( p_i(x) \), corresponding to \( f_i(x) \) on the given interval, such that

\[
|f_i(x) - p_i(x)| \leq \epsilon_1, \quad \forall x \in [a, b].
\]  

Then, it obtains a local vector \( p_i^{0} \) storing the information of the Chebyshev coefficients with additional computations. The details are summarized in Algorithm 1.

In Algorithm 1, \( p_i(x) \) is constructed by using Adaptive Chebyshev Interpolation (ACI) [20]. The key insight is to systematically increase the degree of the Chebyshev interpolant until certain stopping criterion (proposed in [20]) is met. Agents then calculate the Chebyshev coefficients \( \{c_j\} \) of the derivative of \( p_i(x) \) with (4). The \( m_i \)-dimensional local vector \( p_i^{0} \) stores these coefficients as well as the first coefficient \( c_0 \) of \( p_i(x) \), and will be exchanged and updated afterward.

ACI is robust and can yield \( p_i(x) \) satisfying (7) for most Lipschitz continuous \( f_i(x) \) that are analytic or have several orders of derivatives [20]. Therefore, we can conclude that \( p_i(x) \) obtained from Algorithm 1 satisfies (7).

B. Iteration: Consensus-based Update of Local Vectors

In the iteration stage, agents update their local vectors based on consensus, so as to make them converge to the average of all the initial values. The goal is to ensure that when the iteration stage ends,

\[
\max_{i \in \mathcal{V}} \|p_i^K - \bar{p}\|_\infty \leq \delta
\]

holds, where \( K \) is the number of iterations, \( p_i^K \) is the local vector of agent \( i \) and \( \bar{p} = 1/N \sum_{i=1}^{N} p_i^{0} \) is the average. To meet the requirement of precision, \( \delta \) is set by

\[
\delta = \frac{\epsilon_2}{1 + \frac{1}{S} \left( \ln m + \frac{3}{2} \right)}. \tag{8}
\]

Note that \( \delta \) is proportional to the given precision \( \epsilon_2 \), with \( m = \max_{i \in \mathcal{V}} m_i \) and \( S = 2/(b-a) \). The details are summarized in Algorithm 2.

Remark 1. After the initialization, different agents have \( p_i^{0} \) of different dimensions \( m_i + 1 \). While going on the update in (9) and (10), agents ensure agreements in dimension by aligning and padding zeros to lower dimensional local vectors involved. After finite number of iterations (less than the diameter of \( \mathcal{G} \), certainly less than \( U \)), all the local vectors will be of the same dimension \( \max_{i \in \mathcal{V}} m_i + 1 = m + 1 \).

In Algorithm 2, local vectors \( p_i^{k} \) are updated based on the linear time average consensus in (2), and will converge to \( \bar{p} \). There are also two auxiliary vectors, \( r_i^{k} \) and \( s_i^{k} \), updated

Algorithm 1 Initialization of CPCA

\begin{itemize}
  \item \textbf{Input:} \( f_i(x), \epsilon_1 \) and \( X = [a, b] \).
  \item \textbf{Output:} \( p_i^{0} \).
\end{itemize}

1. \textbf{Initialize} \( m_i = 2 \).
2. Calculate \( S_{m_i} = \{x_j\}, \{f_j\} \) according to

\[
\left\{ \begin{array}{l}
  x_j = \frac{b-a}{2} \cos \left( \frac{j\pi}{m_i} \right) + a + \frac{b}{2}, \\
  f_j = f_i(x_j),
\end{array} \right.
\]

where \( j = 0, 1, \ldots, m_i \).
3. Calculate \( \{c_k\} \) according to

\[
c_k = \frac{2}{m_i} \sum_{j=0}^{m_i} f_j \cos \left( \frac{kj\pi}{m_i} \right),
\]

where the double prime indicates that the first and the last terms are halved, and \( k = 0, 1, \ldots, m_i \).
4. If \( \max_{x_j \in \{x_{2m_i} - S_{m_i}\}} |f_i(x_j) - p_i(x_j)| \leq \epsilon_1 \), then go to step 5. Or set \( m_i \leftarrow 2m_i \) and go to step 2.
5. Calculate \( \{c_k^{(k = 0, \ldots, m_i - 1)}\} \) with (4).
6. Set \( p_i^{0} = [c_0, c_0, c_1, \ldots, c_{m_i - 1}] \).
Algorithm 2 Iteration of CPCA

Input: $p^K_t$, $q_t$, $U$ and $X = [a, b]$.
Output: $p^K_t$.
1: Initialize: $q^0_t = r^0_t = s^0_t = p^0_t$, $K = U$, $S = 2/(b - a)$, $\rho = 1/\sqrt{9U}$.
2: while $t \leq K$ do
3:   1. Update $p_t^i$ and $q_t^i$ according to
5:   
6:   
7: if $t = K$ and $t = U$ then
8:   
9: end if
10: return $p^K_t$.

according to the maxi/minium consensus in (1) within the same iteration. After $U$ iterations, agents obtain the initial maximum deviation $d_0 := \max_{i,j \in V} \|p^0_i - p^0_j\|_{\infty} = \|r^0 - s^0\|_{\infty}$, and then calculate an upper bound $K$ on the number of iterations needed to guarantee specific average consensus precision. After going on $K$ iterations, agents terminate simultaneously. The following theorem demonstrates the effectiveness of Algorithm 2.

Theorem 2. For Algorithm 2, we have
\[
\max_{i \in V} \|p^K_i - p^t\|_{\infty} \leq \delta,
\] (12)
where $\delta$ satisfies (8).

Proof. First, we show that $\sum_{i \in V} p^t_i = \sum_{i \in V} p^0_i$, $\forall t \in \mathbb{N}_+$. It follows from Remark 1 that all $p_t^i$ will be of the same dimension in finite time. To simplify the analysis, we assume that the aligning and zero-padding mentioned in Remark 1 are completed in advance and that $p^0_i \in \mathbb{R}^{n+1}$, $\forall i \in V$. Let $P^t = [p^t_1, \ldots, p^t_N]$ and $Q^t = [q^t_1, \ldots, q^t_N]^t$. By introducing the lazy Metropolis weight matrix $W$, we rewrite (9) as
\[
\begin{align*}
P^t &= WQ^{t-1}, \\
Q^t &= P^t + \left(1 - \frac{2}{9U + 1}\right)(P^t - P^{t-1}),
\end{align*}
\]
where $P^0$ and $Q^0$ are initialized to be equal. Since $W$ is doubly stochastic, $1^tP^t = 1^tWQ^0 = 1^tQ^0 = 1^tP^0$. Assume that for $t = k$, $1^tP^k = 1^tP^{k-1}$. Since
\[
1^tQ^k = 1^tP^k + \left(1 - \frac{2}{9U + 1}\right)1^t(P^k - P^{k-1}) = 1^tP^k,
\]
we have $1^tP^{k+1} = 1^tWQ^k = 1^tQ^k = 1^tP^k$. Therefore,

Algorithm 3 Finding Minima of CPCA

Input: $p^K_t$ and $X = [a, b]$.
Output: $f^*_t$ and $X^*_t$.
1: Construct $M_C$ based on $\{p^K_t(j) \mid j = 2, \ldots, m + 1\}$ by (5).
2: Calculate $\{c_j \mid j = 0, \ldots, m\}$ according to
\[
\begin{align*}
c_j &= \begin{cases}
\frac{p^K_t(1)}{2S}, & j = 0, \\
n\frac{2p^K_t(j + 1) - p^K_t(j + 3)}{2S}, & j = 2, \ldots, m,
\end{cases}
\]
where $S = 2/(b - a)$, and $p^K_t(j)$ is defined as 0 for $j$ out of range.
3: Calculate the real eigenvalues $E = \{\lambda_j\}$ of $M_C$ that lie in $[a, b]$.
4: Set
\[
f^*_t = \min_{x \in X} p^K_t(x), \quad X^*_t = \arg\min_{x \in X} p^K_t(x),
\]
where $X_K = E \cup \{a, b\}$, and
\[
p^K_t(x) = \sum_{j=0}^{m} c_j T_j \left(\frac{2x - (a + b)}{b - a}\right), \quad x \in [a, b].
\]

Finally, we prove (12). By (11), Algorithm 2 will at least go on $\left[\ln(\delta/2\sqrt{2Ud_0})/\ln(\rho)\right]$ iterations. Therefore,
\[
\|p^K_t - p^t\|_{\infty} \leq d_T \leq 2\sqrt{2U}d_0 \leq \delta.
\]
C. Finding Minima: Taking a Straightforward Approach

In this stage, agents optimize the polynomial proxy $p^K_t(x)$ recovered from $p^K_t$ independently, and thus obtain $\epsilon$-optimal solutions for (6). This is done by first finding the stationary points of $p^K_t(x)$, then taking the minimum of all the critical values of it. The details are summarized in Algorithm 3.

In Algorithm 3, agent $i$ first prepares the colleague matrix $M_C$ of order $(m - 1)$ based on all but the first elements of $p^K_t$. Then, it calculates $\{c_j\}$ based on a reverse recurrence
We first establish the closeness between $p_i^K(x)$ and $p_i(x)$. Let $e^K = p_i^K - p_i$. By Theorem 2, $||e^K||_\infty \leq \delta$. Suppose the Chebyshev coefficients of $p_i^K(x)$ and $p_i(x)$ are $\{c_j\}$ and $\{\tau_j\}$, respectively. Then,

$$
|c_j - \tau_j| \leq \begin{cases} 
|e^K(1)| \leq \delta, & j = 0, \\
\frac{|2e^K(2) - e^K(4)|}{2S} \leq \frac{3\delta}{2S}, & j = 1, \\
\frac{|e^K(j + 1) - e^K(j + 3)|}{2jS} \leq \frac{\delta}{jS}, & j = 2, \ldots, m.
\end{cases}
$$

Consequently,

$$
|p_i^K(x) - p_i(x)| \leq \sum_{j=0}^{m} |c_j - \tau_j| \cdot 1 < \delta \left(1 + \frac{1}{S} \left(\frac{1}{2} + \sum_{j=1}^{m} \frac{1}{m} \right)\right)
$$

$$
< \delta \left(1 + \frac{1}{S} \left(\frac{3}{2} + \ln m\right)\right) \leq \epsilon_2,
$$

where we use the fact that $|T_j(x)| \leq 1, \forall x \in [-1, 1]$.

Then, we establish the closeness between $\hat{p}(x)$ and $f(x)$. Since $\hat{p}$ is the average of all $p_i^0(x)$, $\hat{p}(x)$ is also the average of all $p_i(x)$. Based on the results in Section III-A, we have

$$
|\hat{p}(x) - f(x)| = \frac{1}{N} \sum_{i=1}^{N} \left|p_i(x) - f_i(x)\right| 
$$

$$
\leq \frac{1}{N} \sum_{i=1}^{N} |p_i(x) - f_i(x)| \leq \frac{1}{N} N \epsilon_1 = \epsilon_1.
$$

Note that $\epsilon_1 = \epsilon_2 = \epsilon/2$. Hence,

$$
|p_i^K(x) - f(x)| \leq |p_i^K(x) - \hat{p}(x)| + |\hat{p}(x) - f(x)| 
$$

$$
\leq \epsilon_1 + \epsilon_2 = \epsilon.
$$

Based on Lemma 3, we have $|f_e^* - f^*| \leq \epsilon$. \qed

B. Complexity of CPCA

We provide the analysis of the computational and communication complexities of CPCA. The following theorem establishes the total number of elementary operations (additions and multiplications), zero-order oracle queries (calls for objective functions’ values) and inter-agent communication needed while running CPCA from a single agent’s perspective. The details are summarized in Table I.

<table>
<thead>
<tr>
<th>Stages</th>
<th>Operations</th>
<th>Oracles</th>
<th>Communications</th>
</tr>
</thead>
<tbody>
<tr>
<td>init</td>
<td>$O\left(m^2 \log m\right)$</td>
<td>$O(m)$</td>
<td>$O\left(N \log \left(\frac{N \log m}{\epsilon}\right)\right)$</td>
</tr>
<tr>
<td>iteration</td>
<td>$O\left(N \log \left(\frac{N \log m}{\epsilon}\right)\right)$</td>
<td>$O(m)$</td>
<td>$O\left(N \log \left(\frac{N \log m}{\epsilon}\right)\right)$</td>
</tr>
<tr>
<td>solve</td>
<td>$O\left(m^2\right)$</td>
<td>$0$</td>
<td>$0$</td>
</tr>
<tr>
<td>whole</td>
<td>$O\left(N \log \left(\frac{N \log m}{\epsilon}\right)\right)$</td>
<td>$O(m)$</td>
<td>$O\left(N \log \left(\frac{N \log m}{\epsilon}\right)\right)$</td>
</tr>
</tbody>
</table>

Note that $m$ is decided by the properties of the objectives, and is independent of $N$ and $\epsilon$. The first-order and zero-order oracle complexities of CPCA are $0$ and $O(m)$, respectively, and is independent of the iterations. This implies that our algorithm has low computational costs.
not estimates of the optimizers. Future works include designing more efficient terminating rules for the iterations within the algorithm to reduce communication complexities, and developing similar algorithms for problems with multivariate objectives based on the idea of approximation.

REFERENCES