2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

Constrained Distributed Nonconvex Optimization over Time-varying Directed Graphs

Zhiyu He, Jianping He, Cailian Chen and Xinping Guan

Abstract— **This paper addresses a class of constrained distributed nonconvex optimization problems involving univariate objective functions, considering time-varying directed networks. We propose a novel algorithm named Extended-CPCA (E-CPCA), exploiting the notion of combining Chebyshev polynomial approximation and average consensus. The proposed algorithm is i) able to yield ϵ globally optimal solutions for any arbitrarily small given tolerance ϵ, ii) efficient in terms of both oracle complexities and inter-agent communication costs, and iii) distributed terminable when the specified precision requirement is met. The idea of leveraging polynomial proxy and consensus to deal with the mentioned problems over static undirected graphs is first presented in our previous work. The novelties of this work lie in i) the utilization of push-sum average consensus with distributed stopping mechanism to enable agents to acquire a proxy for the global objective over time-varying digraphs without much wastes of extra communications, and ii) the transformation of the optimization of this global proxy into a semidefinite program to help in obtaining solutions in a fast and reliable manner. Both the analysis and simulations are provided to illustrate the efficacy of the proposed algorithm.**

I. Introduction

Distributed optimization enables multiple agents in a network to collaboratively optimize the average of local objective functions by using local communication and computation only. It has gained considerable attention for its wide application scenarios, including distributed learning [1], statistics [2], estimation [3] and coordination [4] in various large-scale networked systems, e.g., wireless sensor networks, smart grids and robot swarms.

A variety of efficient distributed convex optimization algorithms have been proposed. Most of them can be divided into two categories, i.e., primal and dual-based methods. Primal methods generally combine (sub)gradient descent with consensus, so as to drive local estimates to converge consensually to the globally optimal point in the primal domain [5]–[7]. Dual-based methods transform the problem by introducing consensus equality constraints, and then solve the dual problem or carry on primal-dual updates to finally reach a saddle point of the Lagrangian function [8]–[10]. These methods are preferable if the computations of dual (sub)gradients or sub-optimizations relating to the alternating direction method of multipliers are easy to execute. However, they are rather hard to be extended to deal with time-varying or directed graphs, since how to enforce consensus as equality constraints in those cases remains unclear.

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, xpguan}@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).

On the other hand, distributed nonconvex optimization is more challenging, and has recently received increasing attention due to its promising values in certain important applications. Several noticeable algorithms have appeared in the literatures, e.g., [11]–[13]. The overall algorithmic frameworks designed share similarities with those for convex problems. Nevertheless, the use of various techniques, including successive convex approximation [13], stochastic gradient descent [11] and proximal methods [12], managed to enable agents to iteratively converge to the stationary points or locally optimal points of nonconvex problems.

We point out two notable issues within the field of distributed nonconvex optimization. First, for general problems with nonconvex objectives, only the convergence to stationary or locally optimal solutions is guaranteed. Second, for most existing algorithms, the load of oracle queries (i.e., calls for gradients or values of objective functions) grows with respect to iterations. This increasing load stems from the iterative algorithmic structures, where local oracle queries are constantly performed at every iteration. It remains challenging to resolve these issues simultaneously by using existing distributed first-order or zeroth-order algorithms.

Recently, to address set constrained distributed nonconvex optimization with univariate objectives, [14] proposed a promising algorithm termed CPCA. The key ideas include i) optimizing locally a sufficiently precise Chebyshev polynomial approximation (i.e., proxy) [15]–[17] for the global objective on the constraint set, and ii) employing average consensus, where the information of local proxies is communicated, to enable agents to acquire such a global proxy. These designs contribute to its advantages of being able to obtain ϵ globally optimal solutions, and being computationally efficient, in that zeroth-order oracle queries are only involved when constructing local proxies and are independent of iterations. More importantly, this Chebyshev-proxy-aided idea differentiates it from existing gradient-based iterative methods, and offers a new view to approach distributed optimization problems.

Nevertheless, CPCA has two issues remaining to be resolved. First, it is communication-inefficient due to the conservative stopping criterion for its inner consensus iterations. The required number of iterations is set in advance based on the initial value and bound for the decaying rate of the errors. Since such an available bound may not be tight enough, which is especially the case while dealing with time-varying digraphs, the number of iterations undergone can exceed what is actually needed to some extent. Second, certain practical numerical errors are not taken into account. CPCA optimizes the polynomial proxy by first finding its stationary points, which are exactly the eigenvalues of a specific colleague matrix, and then evaluating the proxy at these points and deciding the minimum. However, the influence of numerical errors arising while calculating theses points is not covered in the theoretical analysis of [14]. To address these issues, E-CPCA is proposed in this work. The main contributions are summarized as follows.

1) E-CPCA is developed to solve a class of constrained distributed nonconvex optimization problems with univariate Lipschitz objectives and different local constraint sets over time-varying directed graphs. We demonstrate that it preserves CPCA's advantages of being able to yield ϵ globally optimal solutions for any arbitrarily small given error tolerance ϵ and being computationally efficient.
2) We address the aforementioned issues of inefficiency in communications and presence of unconsidered numerical errors. E-CPCA is more communication-efficient. This improvement results from the incorporation of effective distributed stopping mechanism, which introduces auxiliary variables to bound errors and achieve in-time terminations, for its inner push-sum consensus iterations. It also guarantees that the total errors are controllable both theoretically and practically. This aim is achieved by transforming the optimization of the polynomial proxy to a semidefinite program, and then using the given tolerance ϵ to help terminate the solving procedures.
3) We provide explicit expressions for the overall communication and zeroth-order oracle complexities of E-CPCA regarding the number of agents N, the maximum degree of local approximations m, and ϵ. It is shown that E-CPCA is efficient in terms of both communication and computational complexities, and achieves distributed termination once the precision requirement is met.

Notations. For any vector a, we use $\|a\|$ to denote its l_2-norm. The superscript t denotes the number of iterations, and the subscripts i, j denote the indexes of agents. The script in parentheses k denotes the index of the elements in a vector.

II. Problem Formulation and Preliminaries

A. Problem Formulation

Consider a network with N agents, each of which has a local objective function $f_i(x) : X_i \to \mathbb{R}$ and a local constraint set $X_i \subseteq \mathbb{R}$. The goal is to solve the following constrained optimization problem

$$\begin{aligned} \min_x \quad & f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x), \\ \text{s.t.} \quad & x \in X = \bigcap_{i=1}^{N} X_i, \end{aligned} \tag{1}$$

in a distributed manner. The network at time t is described as a directed graph $\mathcal{G}^t = (\mathcal{V}, \mathcal{E}^t)$, where $\mathcal{V}$ is the set of agents, and $\mathcal{E}^t \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Note that agent j can receive information from agent i at time t if and only if (*iff*) $(i, j) \in \mathcal{E}^t$. Some basic assumptions are given as follows.

Assumption 1. *Every $f_i(x)$ is Lipschitz continuous on X_i.*

Assumption 2. *All X_i are closed, bounded and convex sets.*

Assumption 3. *$\{\mathcal{G}^t\}$ is B-strongly-connected, i.e., there exists a positive integer B, such that for any $k \in \mathbb{N}$, the graph $\left(\mathcal{V}, \bigcup_{t=kB}^{(k+1)B-1} \mathcal{E}^t\right)$ is strongly connected.*

Problem (1) has (possibly) nonconvex objectives and convex constraint sets, and therefore is a constrained distributed nonconvex optimization problem. Under Assumption 2, for all $i \in \mathcal{V}$, X_i is a closed interval. Thus, let $X_i = [a_i, b_i]$, where $a_i, b_i \in \mathbb{R}$. Hence, $X = [a, b]$, where $a = \max_{i\in\mathcal{V}} a_i,\ b = \min_{i\in\mathcal{V}} b_i$.

B. Preliminaries

We provide some preliminary knowledge on which our algorithm is built.

• *Consensus:* Let $\mathcal{N}_i^{\text{in},t} = \{j | (j,i) \in \mathcal{E}^t\}$ and $\mathcal{N}_i^{\text{out},t} = \{j | (i,j) \in \mathcal{E}^t\}$ be the sets of agent i's in-neighbors[1] and out-neighbors, respectively, and $d_i^{\text{out},t} = |\mathcal{N}_i^{\text{out},t}|$ be its out-degree, where $|\mathcal{N}_i^{\text{out},t}|$ denotes the cardinality of $\mathcal{N}_i^{\text{out},t}$. Suppose that every agent i maintains a local variable $x_i^t \in \mathbb{R}^n$. The maximum consensus algorithm [18] is

$$x_i^{t+1}(k) = \max_{j\in\mathcal{N}_i^{\text{in},t}} x_j^t(k), \tag{2}$$

where $k(= 1, \ldots, n)$ is the index of the elements in x_i^t. It can be proven that with (2), all x_i^t converge to $\max_{i\in\mathcal{V}} x_i^0$ in $T(\le (N-1)B)$ iterations [19], i.e.,

$$x_i^t = \max_{i\in\mathcal{V}} x_i^0, \quad \forall t \ge T,\ i \in \mathcal{V}.$$

The push-sum average consensus algorithm [20] is

$$x_i^{t+1} = \sum_{j=1}^{N} a_{ij}^t x_j^t, \quad y_i^{t+1} = \sum_{j=1}^{N} a_{ij}^t y_j^t, \tag{3}$$

where $y_i^t \in \mathbb{R}$ is initialized to be 1 for all $i \in \mathcal{V}$. The weight a_{ij}^t is set as $1/d_i^{\text{out},t}$ if $j \in \mathcal{N}_i^{\text{in},t}$, and 0 otherwise. Let $A^t \triangleq \left(a_{ij}^t\right)_{N\times N}$. It follows that A^t is column stochastic. With (3), the ratio $z_i^t \triangleq x_i^t / y_i^t$ converges geometrically to the average of the initial values $\bar{x} = 1/N \sum_{i=1}^{N} x_i^0$, i.e.,

$$\lim_{t\to\infty} z_i^t = \bar{x}, \quad \forall i \in \mathcal{V}.$$

Based on the proof of Theorems 1 and 12 in [21], we derive the following theorem characterizing the decaying of $e^t(k) \triangleq \max_{i,j\in\mathcal{V}} \left|z_i^t(k) - z_j^t(k)\right|$.

[1]We follow the convention that $i \in \mathcal{N}_i^{\text{in},t}$, i.e., agent i can always access its own information.

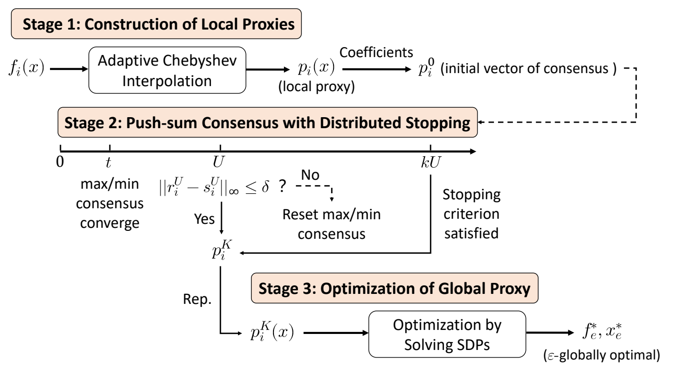

Stage 1: Construction of Local Proxies
$f_i(x)$
Adaptive Chebyshev Interpolation
$p_i(x)$
(local proxy)
Coefficients
p_i^0 (initial vector of consensus)
Stage 2: Push-sum Consensus with Distributed Stopping
0
t
U
kU
max/min consensus converge
$\|r_i^U - s_i^U\|_\infty \le \delta$?
No
Yes
Reset max/min consensus
Stopping criterion satisfied
p_i^K
Rep.
Stage 3: Optimization of Global Proxy
$p_i^K(x)$
Optimization by Solving SDPs
f_e^*, x_e^*
(ε-globally optimal)

Fig. 1. An Overview of E-CPCA

Theorem 1. *If Assumption 3 holds, with (3), we have*

$$e^t(k) \le \left(1 - \frac{2}{N^{(N-1)B}}\right)^{\left\lfloor \frac{t}{(N-1)B} \right\rfloor} e^0(k), \quad \forall k.$$

• *Chebyshev Polynomial Approximation:* For a Lipschitz continuous function $g(x)$ with $x \in [a, b]$, its degree m Chebyshev interpolant $p_m(x)$ is

$$p_m(x) = \sum_{j=0}^{m} c_j T_j\left(\frac{2x - (a+b)}{b-a}\right), \quad x \in [a,b], \tag{4}$$

where $T_j(\cdot)$ denotes the j-th Chebyshev polynomial defined on $[-1, 1]$. As m increases, $p_m(x)$ converges to $g(x)$ uniformly on the given interval [15], i.e.,

$$\forall x \in [a,b], \ |p_m(x) - g(x)| \to 0, \text{ as } m \to \infty.$$

This observation makes computing $p_m(x)$ a practical way to construct arbitrarily close polynomial approximation for $g(x)$, as theoretically guaranteed by the *Weierstrass Approximation Theorem.*

Let $v(x) \triangleq [T_0(x), \ldots, T_d(x)]^T$ be the vector of Chebyshev polynomials of degree less than or equal to d. For any univariate polynomial $p(x)$ of degree $2d$, it is non-negative (or equivalently, a sum of squares (SOS)) on $\mathbb{R}$ *iff*

$$p(x) = v(x)^T Q v(x) \tag{5}$$

holds, where $Q \in \mathbb{S}_+^{d+1}$ is a positive semidefinite matrix [22].

III. Algorithmic Framework

In this section, we present E-CPCA to solve problem (1) in a distributed manner. Figure 1 illustrates the main architecture of the proposed algorithm.

A. Construction of Local Chebyshev Proxies

In this stage, based on the adaptive Chebyshev interpolation scheme [17], every agent i constructs the approximating polynomial $p_i(x)$ corresponding to $f_i(x)$ on $X = [a, b]$, s.t.

$$|f_i(x) - p_i(x)| \le \epsilon_1, \quad \forall x \in [a,b] \tag{6}$$

holds, where ϵ_1 is a specified positive tolerance. The key insight is to systematically increase the degree of the Chebyshev interpolant until certain stopping criterion is satisfied. To illustrate, agent i initializes m_i as 2 and starts to calculate a Chebyshev interpolant of degree m_i. It first evaluates $f_i(x)$ at the $(m_i + 1)$-point grid $S_{m_i} \triangleq \{x_k\}$ according to

$$\begin{cases} x_k = \dfrac{b-a}{2}\cos\left(\dfrac{j\pi}{m_i}\right) + \dfrac{a+b}{2}, \\ f_k = f_i(x_k), \end{cases} \tag{7}$$

where $k = 0, 1, \ldots, m_i$. Then, it computes the Chebyshev coefficients of the interpolant by

$$c_j = \frac{1}{m_i}\left(f_0 + f_{m_i}\cos(j\pi)\right) + \frac{2}{m_i}\sum_{k=1}^{m_i - 1} f_k \cos\left(\frac{jk\pi}{m_i}\right), \tag{8}$$

where $j = 0, 1, \ldots, m_i$ [23]. The degree m_i is doubled at every iteration until the stopping criterion

$$\max_{x_j \in \left(S_{2m_i} - S_{m_i}\right)} |f_i(x_j) - p_i(x_j)| \le \epsilon_1, \tag{9}$$

is met, where $p_i(x)$ is in the form of (4) with $\{c_k\}$ being the coefficients. Since $S_{m_i} \subset S_{2m_i}$, the evaluations of $f_i(x)$ are continuously reused. Note that agents know the intersection set $X = [a, b]$ by running a finite number of max/min consensus iterations like (2) in advance.

B. Consensus-based Information Dissemination

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

$$\max_{i \in \mathcal{V}} \left\| p_i^K - \bar{p} \right\|_\infty \le \delta$$

holds, where K is the number of iterations, p_i^K is the local variable of agent i and $\bar{p} = 1/N \sum_{i=1}^{N} p_i^0$ is the average. To meet the requirement of precision, δ is set as $\epsilon_2/(m+1)$, where $m \triangleq \max_{i \in \mathcal{V}} m_i$ is the highest degree among the local approximations. The details are as follows.

Every agent keeps two local variables, x_i^t and y_i^t, and update them according to the push-sum average consensus algorithm in (3). The ratio $p_i^t \triangleq x_i^t / y_i^t$ converges to $\bar{p}$ geometrically. To achieve distributed stopping once the consensus has reached within a given error tolerance, we incorporate the max/min-consensus-based stopping mechanism in [24]. This mechanism requires the following assumption so that agents know how long to wait to ensure that the max/min consensus algorithms converge.

Assumption 4. *Every agent i in $\mathcal{G}$ knows an upper bound U on $(N-1)B$, such that U is of the same order as $(N-1)B$.*

To realize distributed stopping, there are two auxiliary variables, r_i^t and s_i^t, updated in parallel with x_i^t and y_i^t according to

$$r_i^{t+1}(k) = \max_{j \in \mathcal{N}_i^{\text{in},t}} r_j^t(k), \quad s_i^{t+1}(k) = \min_{j \in \mathcal{N}_i^{\text{in},t}} s_j^t(k), \tag{10}$$

where $k = 0, \ldots, m$. They are reinitialized as p_i^t every U iterations, so as to facilitate the constant dissemination of recent information of p_i^t. Note that the number of iterations that the max/min consensus need to reach convergence is less than $(N-1)B$ [19], and therefore less than U. Once the stopping criterion

$$\|r_i^t - s_i^t\|_\infty \le \delta \tag{11}$$

is met, agents terminate the iterations simultaneously, and set $p_i^K = x_i^t / y_i^t$.

C. Polynomial Optimization by Solving SDPs

In this stage, agents optimize the polynomial proxy $p_i^K(x)$ recovered from p_i^K independently, thus obtaining ϵ-optimal solutions for problem (1). The optimization of $p_i^K(x)$ on $X = [a, b]$ can be reformulated as a semidefinite program (SDP). It can be readily solved by the interior-point method [25]. We offer such a reformulation based on the coefficients of $p_i^K(x)$ with respect to the Chebyshev polynomial basis, rather than the monomial basis as in [22].

Note that $p_i^K(x)$ is a polynomial of degree m in the form of (4), with the elements of $p_i^K = [c_0, \ldots, c_m]^T$ being the coefficients. We first transform the problem of optimizing $p_i^K(x)$ on $[a, b]$ to its equivalent form

$$\max_{x,t}\ t \quad \text{s.t. } p_i^K(x) - t \ge 0,\ x \in [a, b]. \tag{12}$$

For $g(x) \triangleq p_i^K(x) - t$, its non-negativity on $[a, b]$ holds *iff* it can be expressed as

$$g(x) = \begin{cases} (x+1)h_1^2(x) + (1-x)h_2^2(x), & \text{if } m \text{ is odd,} \\ h_1^2(x) + (x+1)(1-x)h_2^2(x), & \text{if } m \text{ is even,} \end{cases}$$

where $h_1(x), h_2(x)$ are SOS of degree $\lfloor m/2 \rfloor$, $\lfloor (m-1)/2 \rfloor$, respectively [22]. Our reformulations are based on the fact that

$$2T_u(x)T_v(x) = T_{u+v}(x) + T_{|u-v|}(x).$$

When m is odd, we reformulate problem (12) as

$$\begin{aligned} \max_{t,Q,Q'} \quad & t \\ \text{s.t.} \quad & c_0 = t + Q_{00} + Q'_{00} + \frac{1}{2}\sum_{u=1}^{m}(Q_{uu} + Q'_{uu}) \\ & \qquad + \frac{1}{4}\sum_{|u-v|=1}(Q_{uv} - Q'_{uv}) \\ & c_j = \frac{1}{2}\sum_{(u,v)\in\mathcal{A}}(Q_{uv} + Q'_{uv}) \\ & \qquad + \frac{1}{4}\sum_{(u,v)\in\mathcal{B}}(Q_{uv} - Q'_{uv}), \quad j = 1, \ldots, m, \\ & Q \in \mathbb{S}_+^{\lfloor m/2 \rfloor + 1},\ Q' \in \mathbb{S}_+^{\lfloor (m-1)/2 \rfloor + 1}, \end{aligned}$$

where rows and columns of Q and Q' are indexed by $0, 1, \ldots$, $\mathcal{A} = \{(u,v) | u+v = i \vee |u-v| = i\}$, and $\mathcal{B} = \big\{(u,v) | u+v = i-1 \vee |u-v| = i-1 \vee |u+v-1| = i \vee \big||u-v|-1\big| = i\big\}$. When m is even, the reformulated problem has a similar form. We refer readers to our recent paper [26] for more details on the forms and sensibilities of these reformulations[2].

[2]Such SDP reformulations are only dependent on the Chebyshev coefficients of the polynomial $p_i^K(x)$ to be optimized, and are independent of the network topology. Hence, the analysis of the reformulations in [26], which considers static undirected networks, also applies to this paper.

Algorithm 1 E-CPCA

```
Input: $f_i(x), X_i = [a_i, b_i], U$ and $\epsilon$.
Output: $f_e^*$ for every agent $i \in \mathcal{V}$.
 1: Initialize: $a_i^0 = a_i, b_i^0 = b_i, m_i = 2$.
 2: for each agent $i \in \mathcal{V}$ do
 3:     for $t = 0, \ldots, U-1$ do
 4:         $a_i^{t+1} = \max_{j \in \mathcal{N}_i^{\text{in},t}} a_i^t,\ b_i^{t+1} = \max_{j \in \mathcal{N}_i^{\text{in},t}} b_i^t$.
 5:     end for
 6:     Set $a = a_i^t,\ b = b_i^t$.
    ......................................................
 7:     Calculate $\{x_j\}$ and $\{f_j\}$ by (7).
 8:     Calculate $\{c_k\}$ by (8).
 9:     If (9) is satisfied (where $\epsilon_1 = \epsilon/3$), go to step 10. Or set
        $m_i \leftarrow 2m_i$ and go to step 7.
    ......................................................
10:     Set $x_i^0 = r_i^0 = s_i^0 = [c_0, c_1, \ldots, c_{m_i}]^T, y_i^0 = l = 1$.
11:     for $t = 0, 1, \ldots$ do
12:         if $t = lU$ then
13:             if $l = 1$ then
14:                 $\delta = \epsilon_2/(m+1) = \epsilon/3(m+1)$.
15:             end if
16:             if $\|r_i^t - s_i^t\|_\infty \le \delta$ then
17:                 $p_i^K = x_i^t/y_i^t$. break
18:             end if
19:             $r_i^t = s_i^t = x_i^t/y_i^t,\ l \leftarrow l+1$.
20:         end if
21:         Update $x_i^{t+1}, y_i^{t+1}, r_i^{t+1}, s_i^{t+1}$ by (3) and (10).
22:         Set $t \leftarrow t+1$.
23:     end for
    ......................................................
24:     Solve the reformulated problems in Sec. III-C with $\epsilon_3 = \epsilon/3$
        and return $f_e^*$.
25: end for
```

Note that these reformulated problems are SDPs, and therefore can be efficiently solved by the interior-point method. The optimal points of $p_i^K(x)$ can be computed from the complementary slackness condition [22]. The solving process is terminated when $0 \le f_e^* - p^* \le \epsilon_3$, where f_e^* is the returned estimate of the optimal value p^* of $p_i^K(x)$ on X and $\epsilon_3 > 0$ is some specified tolerance.

The above transformation offers a plausible alternative to the stationary-point-based method of optimizing $p_i^K(x)$ in [14]. The advantage is that it managed to make all potential numerical errors controllable. In fact, the latter method requires calculations of the eigenvalues of a specific colleague matrix to obtain the stationary points. The errors associated with such calculations have not been covered in the analysis of [14]. In contrast, the SDP-based solving procedure presented here ensures that an explicit bound (i.e., ϵ_3) exists for the total errors.

D. Description of E-CPCA

E-CPCA is composed of the three stages discussed previously, and is formulated as Algorithm 1.

In Algorithm 1, lines 1-6 perform the initialization; lines 7-9 complete the construction of the local proxy; lines 10-23 correspond to consensus iterations with distributed stopping; and line 24 is the polynomial optimization step. Note that the proposed algorithm takes the given error tolerance ϵ as one of the inputs. This tolerance ϵ is used to set some key parameters $\epsilon_1, \epsilon_2, \epsilon_3$, whose sum equals to ϵ, utilized in the corresponding stages.

IV. Performance Analysis of E-CPCA

A. Accuracy

We first provide a lemma stating that if two functions f and g are sufficiently close on the entire interval, their minimum values $f(x_f^*)$ and $g(x_g^*)$ are sufficiently close also. The proof of this lemma can be found in [14].

Lemma 2. *Suppose f, g satisfy $|f(x) - g(x)| \leq \epsilon$, $\forall x \in [a, b]$. Then, $\left|f(x_f^*) - g(x_g^*)\right| \leq \epsilon$.*

The following lemma ensures the effectiveness of the stopping criterion (11) for consensus iterations.

Lemma 3. *When (11) is satisfied, we have*

$$\max_{i \in \mathcal{V}} \left\|p_i^K - \bar{p}\right\|_\infty \leq \delta, \tag{13}$$

where $\delta = \epsilon_2/(m+1)$.

Proof. The proof is omitted due to the space limit. □

The accuracy of E-CPCA is established as follows. We use ϵ and f^* to denote the given error tolerance and optimal value of problem (1), respectively.

Theorem 4. *Suppose that Assumptions 1-4 hold. If ϵ is specified, E-CPCA ensures that every agent obtains ϵ-optimal solutions f_e^* for problem (1), i.e., $|f_e^* - f^*| \leq \epsilon$.*

Proof. The proof is omitted due to the space limit. □

B. Complexity

We provide the analysis of the computational and communication complexities of E-CPCA. The following theorem establishes the order of the total number of zeroth-order and first-order oracle queries (calls for local objective functions' values and gradients) and inter-agent communications needed while running E-CPCA from a single agent's perspective. The details are summarized in Table I.

Theorem 5. *With E-CPCA, every agent obtains ϵ-optimal solutions for problem (1), with $\mathcal{O}(m)$ zeroth-order and 0 first-order oracle queries in $\mathcal{O}\Big(N^{NB} \log \dfrac{m}{\epsilon}\Big)$ communication rounds.*

Proof. In the initialization stage, agent i needs to evaluate $f_i(x)$ at $2m+1$ points to construct an approximation of degree m. Hence, the zeroth-order oracle complexity is $\mathcal{O}(m)$. Since the computation of the gradients of $f_i(x)$ is not required, the first-order oracle complexity is 0. Based on Theorem 1 and the setting that U is of the order as $(N-1)B$, to ensure that the difference between any element of p_i^K and $\bar{p}$ is no more than $\epsilon_3 = \epsilon/3(m+1)$, the number of the consensus iterations undergone is $\mathcal{O}\big(N^{NB} \log \frac{m}{\epsilon}\big)$. The iteration complexity amounts to the inter-agent communication complexity. □

Remark 1. *The order of m depends on ϵ, and also on the smoothness of $f_i(x)$. If $f_i(x)$ has $(v-1)$-th continuous derivative and v-th derivative of bounded variation on X_i, then $m \sim \mathcal{O}\big(\epsilon^{-1/v}\big)$; if $f_i(x)$ is analytic, then $m \sim \mathcal{O}\big(\ln \frac{1}{\epsilon}\big)$ [15]. In practice, extremely high precision (e.g., machine epsilon) can be attained with moderate m (of the order of $10^1 \sim 10^2$) [15], [17].*

TABLE I
Complexities of E-CPCA

Stages	0th-order Oracles	1st-order Oracles	Communications
init	$\mathcal{O}(m)$	0	/
iteration	/	/	$\mathcal{O}\Big(N^{NB} \log \dfrac{m}{\epsilon}\Big)$
solve	/	/	/
whole	$\mathcal{O}(m)$	0	$\mathcal{O}\Big(N^{NB} \log \dfrac{m}{\epsilon}\Big)$

Remark 2. *The communication complexity of E-CPCA amounts to that of push-sum consensus algorithms for time-varying digraphs. We derive this complexity based on the analysis in [21]. The result we obtained is a rather conservative estimate, whose refinement calls for further studies.*

Note that in E-CPCA, the evaluations of the local objectives are required only in the initialization stage. Therefore, the zeroth-order oracle complexity of E-CPCA is independent of the number of iterations. Even if the convergence of push-sum consensus algorithm is slow due to the large scale or poor connectivity of the network, E-CPCA still has fixed oracle complexity for fixed specified precision.

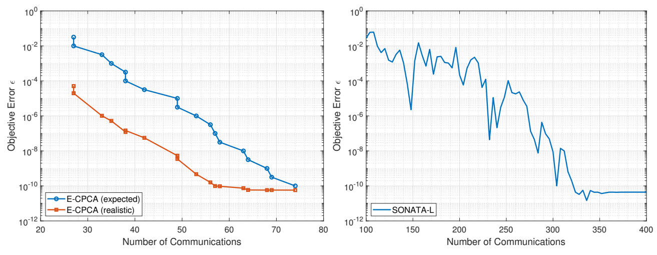
V. Numerical Evaluations

In this section, we present simulation results to illustrate the performance of E-CPCA and compare it with other algorithms. Consider a network with $N = 40$ agents. At each time t, besides itself, every agent i has two out-neighbors. One belongs to a fixed cycle, and the other is chosen uniformly at random. Suppose that all the local constraint sets are the same interval $X = [-5, 5]$. The local objective function of agent i is

$$f_i(x) = \frac{a_i}{1 + e^{-x}} + b_i \log(1 + x^2), \tag{14}$$

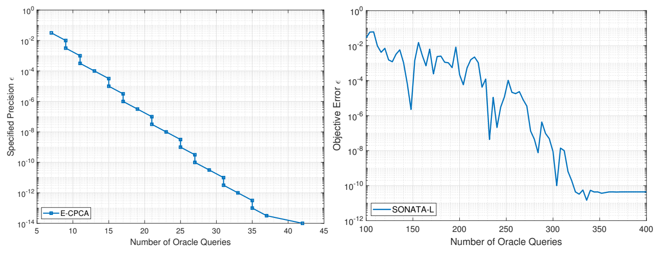
where $a_i \sim \mathcal{N}(10, 2), b_i \sim \mathcal{N}(5, 1)$ are Gaussian random variables. Note that $f_i(x)$ is nonconvex and Lipschitz continuous on X. We use the Chebfun toolbox [16] to help construct Chebyshev polynomial approximations. For comparison, we implement SONATA-L [13], which is the most suitable algorithm in the existing literatures to address problem (1) with the above setups. Its step-size rule is set as $\alpha^t = \alpha^{t-1}(1 - \mu\alpha^{t-1})$ with $\alpha^0 = 0.5$ and $\mu = 0.01$ based on the guideline therein.

Figure 2 shows the relationships between the objective error ϵ and number of communications for both algorithms. For E-CPCA, ϵ denotes $|f_e^* - f^*|$. For SONATA-L, ϵ denotes $|f(\bar{x}^t) - f^*|$, where $\bar{x}^t$ is the average of all agents' local estimates at time t. In Fig. 2(a), the point on the line with circle markers indicates the number of communications needed to meet the specified precision requirement. The point on the line with square markers indicates the actual error after



(a) E-CPCA (b) SONATA-L

Fig. 2. Comparison of E-CPCA and SONATA-L regarding inter-agent communications.



(a) E-CPCA (b) SONATA-L

Fig. 3. Comparison of E-CPCA and SONATA-L regarding oracle queries.

performing this number of communications. We observe that to reach certain given precision, E-CPCA requires less inter-agent communications, and thus is more communication-efficient. The reason is that E-CPCA employs simple average consensus algorithms and can rapidly converge.

Figure 3 shows the relationships between the objective error ϵ and number of oracle queries involved for both algorithms. E-CPCA requires zeroth-order oracle queries. In Fig. 3(a), the horizontal axis represents the average number of queries needed for one agent (i.e., the average degree of local proxies plus 1). The results presented here correspond to the discussions in Remark 1 that m depends on ϵ, and extremely small ϵ is generally accompanied with moderate m. SONATA-L requires one first-order oracle query within each iteration. Hence, the curve in Fig. 3(b) is identical to that in Fig. 2(b). It can be seen that E-CPCA calls for fewer oracle queries than SONATA-L in this example.

VI. Conclusion

In this paper, we propose E-CPCA to solve distributed nonconvex optimization problems with Lipschitz continuous univariate objectives and different local constraint sets, over time-varying directed graphs. By i) following the Chebyshev-proxy-aided idea of CPCA, ii) incorporating effective distributed stopping mechanism for average consensus, and iii) transforming the optimization of polynomial proxies to SDPs, we enable the proposed algorithm to both preserve advantages and overcome shortcomings of CPCA, and to efficiently handle the given challenging problem. We also provide comprehensive theoretical analysis and numerical results to illustrate its effectiveness. Future directions include leveraging the idea of introducing polynomial approximations to deal with general problems with multivariate non-covex objective functions.

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[2] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, no. 1, pp. 33–46, 2007.

[3] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links-Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, 2007.

[4] C. Zhao, J. He, P. Cheng, and J. Chen, "Consensus-based energy management in smart grid with transmission losses and directed communication," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2049–2061, 2016.

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

[6] 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.

[7] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[8] 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, 2011.

[9] 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, 2014.

[10] K. Scaman, F. Bach, S. Bubeck, L. Massoulié, and Y. T. Lee, "Optimal algorithms for non-smooth distributed optimization in networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2740–2749.

[11] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 391–405, 2012.

[12] M. Hong, D. Hajinezhad, and M.-M. Zhao, "Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks," in *Proc. ICML*, 2017, pp. 1529–1538.

[13] G. Scutari and Y. Sun, "Distributed nonconvex constrained optimization over time-varying digraphs," *Math. Program.*, vol. 176, no. 1-2, pp. 497–544, 2019.

[14] Z. He, J. He, C. Chen, and X. Guan, "CPCA: A chebyshev proxy and consensus based algorithm for general distributed optimization," in *Proc. ACC*, 2020.

[15] L. N. Trefethen, *Approximation theory and approximation practice*. SIAM, 2013, vol. 128.

[16] T. A. Driscoll, N. Hale, and L. N. Trefethen, "Chebfun guide," 2014.

[17] J. P. Boyd, *Solving Transcendental Equations: The Chebyshev Polynomial Proxy and Other Numerical Rootfinders, Perturbation Series, and Oracles*. SIAM, 2014, vol. 139.

[18] R. O. Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proc. ACC*, 2003, pp. 951–956.

[19] J. He, P. Cheng, L. Shi, J. Chen, and Y. Sun, "Time synchronization in WSNs: A maximum-value-based consensus approach," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 660–675, 2014.

[20] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. 44th Annu. IEEE Symp. Foundations of Computer Science (FOCS 03)*, 2003, pp. 482–491.

[21] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[22] G. Blekherman, P. A. Parrilo, and R. R. Thomas, *Semidefinite optimization and convex algebraic geometry*. SIAM, 2013, vol. 13.

[23] A. Gil, J. Segura, and N. M. Temme, *Numerical methods for special functions*. SIAM, 2007, vol. 99.

[24] M. Prakash, S. Talukdar, S. Attree, S. Patel, and M. V. Salapaka, "Distributed stopping criterion for ratio consensus," in *Proc. 56th Annu. Allerton Conf. on Commun., Control and Computing*, 2018, pp. 131–135.

[25] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[26] Z. He, J. He, C. Chen, and X. Guan, "Distributed nonconvex optimization: oracle-free iterations and globally optimal solution," *arXiv e-prints*, p. arXiv:2008.00252, 2020.