

Gradient-Consensus Method for Distributed Optimization in Directed Multi-Agent Networks

Vivek Khatana¹, Govind Saraswat¹, Sourav Patel¹ and Murti V. Salapaka¹

Abstract—In this article, a distributed optimization problem for minimizing a sum, $\sum_{i=1}^n f_i$, of convex objective functions, f_i , on directed graph topologies is addressed. Here each function f_i is a function of n variables, private to agent i which defines the agent's objective. These f_i 's are assumed to be Lipschitz-differentiable convex functions. For solving this optimization problem, we develop a novel distributed algorithm, which we term as the *gradient-consensus* method. The *gradient-consensus* scheme uses a finite-time terminated consensus protocol called ρ -consensus, which allows each local estimate to be ρ -close to each other at every iteration. The parameter ρ is a fixed constant independent of the network size and topology. It is shown that the estimate of the optimal solution at any local agent i converges geometrically to the optimal solution within an $O(\rho)$ neighborhood, where ρ can be chosen to be arbitrarily small.

keywords: Distributed optimization, distributed gradient descent, multi-agent networks, finite-time consensus.

I. INTRODUCTION

We focus on a distributed optimization problem defined on a directed graph containing n nodes (or agents). The n agents solve the following problem cooperatively,

$$\underset{u \in \mathbb{R}^n}{\text{minimize}} \quad f(u) = \sum_{i=1}^n f_i(u), \quad (1)$$

over a common decision vector $u \in \mathbb{R}^n$, with $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ being a private convex cost function of the agent i . Problems of the form (1) arise frequently in various engineering applications such as in sensor networks for robust statistical inference [1], multi-agent coordination and control [2], as well as distributed machine learning [3] that can be reduced to variations of the problem (1). The initial study of the problem of distributed optimization can be traced back to the seminal works [4], [5]. Since then, the problem of optimizing the sum of convex functions in a distributed manner has received significant interest [6], [7], [8], [9], [10]. Existing first-order methods for solving (1) in a distributed manner include the gradient or subgradient methods [6], [11]. The subgradient-push method of [12], [13], the fast subgradient method of [14], and the dual averaging method proposed in [10] are among some other existing methods in

the literature. The methods in [6]-[13] achieve a sub-linear convergence rate for convex function f 's. More recent works [15], [16], [17], [18] have achieved better results with linear convergence rates for strongly convex and smooth functions. The algorithm in [14] adopts Nesterov's acceleration method to provide an outer loop complexity of $O(1/k^2)$, where k is the iteration counter. However, the work in [14] utilizes a diminishing step-size rule and an inner loop utilizing multiple consensus steps per iteration which incurs a larger communication burden on the network. To this end, the paper [19] presented a method which utilizes a single consensus step and achieves $O(1/k^{1.4-\epsilon})$ convergence rate for all $\epsilon \in (0, 1.4)$, in terms of the iteration count k .

In this article, we propose a novel approach termed as the *gradient-consensus* scheme. Under the assumption of function f_i 's being strongly-convex with parameter m_{f_i} and L_{f_i} -smooth the *gradient-consensus* gives a geometric rate of convergence. In particular, at iteration k of the algorithm the local estimate of all the agents converge to the optimal solution at a geometric rate with an error which is $O(\rho)$, where ρ can be chosen to be arbitrarily small. The *gradient-consensus* scheme uses a finite-time terminated ρ -consensus protocol (discussed in detail later) along with gradient descent to move towards the optimal solution. The ρ -consensus protocol allows each agent to compute an approximate average of the local estimates produced after the gradient descent iteration. This approximate average is consensual and is ρ -close to the average. Thus, the agent estimates are designed to remain ρ -close to each other after every iteration of the *gradient-consensus* algorithm. As, after each iteration the local estimates of the agents are close to each other the *gradient-consensus* method effectively carries out an approximate centralized gradient descent iteration to solve problem (1). Note that the parameter ρ here, is a constant independent of the problem and the network topology. To establish the applicability of the proposed *gradient-consensus* algorithm we present the results of solving two distributed optimization problems over a directed graph of ten nodes. The results demonstrate the effectiveness of the algorithm.

The rest of the paper is organized as follows. In Section II, we discuss the problem description, and the *gradient-consensus* algorithm. Section III presents the ρ -consensus protocol in detail and provide the convergence results for the *gradient-consensus* method. The numerical implementation results are presented in IV. Section V provides the concluding remarks.

This work is supported by Advanced Research Projects Agency-Energy OPEN through the project titled "Rapidly Viable Sustained Grid" via grant no. DE-AR0001016.

¹ Vivek Khatana {khatana010@umn.edu}, Govind Saraswat {saras006@umn.edu}, Sourav Patel {patel292@umn.edu} and Murti V. Salapaka {murtis@umn.edu} are with department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, USA

II. PROBLEM FORMULATION AND ALGORITHM

A. Graph Theory and Distributed System Model

In this section, we present basic notions of graph theory and linear algebra. Detailed description of graph theory and linear algebra notions are available in [20] and [21].

Definition 1. (Directed Graph) A directed graph $\mathcal{G}$ is a pair $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of vertices and $\mathcal{E}$ is a set of edges, which are ordered subsets of two distinct elements of $\mathcal{V}$.

Definition 2. (Path) In a directed graph, a directed path from node i to j exists if there is a sequence of distinct directed edges of $\mathcal{G}$ of the form $(k_1, i), (k_2, k_1), \dots, (j, k_m)$.

Definition 3. (Strongly Connected Graph) A directed graph is strongly connected if it has a directed path between each pair of distinct nodes i and j .

Definition 4. (Column Stochastic Matrix) A real $n \times n$ matrix $A = [a_{ij}]$ is called a column-stochastic matrix if $0 \leq a_{ij} \leq 1$ and $\sum_{i=1}^n a_{ij} = 1$ for $1 \leq j \leq n$.

Definition 5. (Irreducible Matrix) A $n \times n$ matrix A is said to be irreducible if for any $i, j \in \{1, \dots, n\}$, there exists $m \in \mathbb{N}$ such that $[A^m]_{ij} > 0$.

Definition 6. (Primitive Matrix) A non-negative matrix A is primitive if it is irreducible and has only one eigenvalue of maximum modulus.

Definition 7. (Diameter of a Graph) The diameter of a graph is the longest shortest directed path between any two nodes in the network.

We will consider D as an upper bound on the diameter of the graph throughout the rest of the article.

Definition 8. (Epoch) k^{th} epoch defines the time instant kD .

Definition 9. (In-Neighborhood) The set of in-neighbors of node $i \in \mathcal{V}$ is denoted by $N_i^- = \{j \mid (i, j) \in \mathcal{E}\}$ not including i .

Definition 10. A function f is Lipschitz differentiable with constant L if there exists $L > 0$ such that the following inequality holds for all x, y in the domain of f :

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \forall x, y \in \text{dom } f.$$

Definition 11. (Strongly Convex Function) A differentiable function f is strongly convex with parameter $m_f > 0$ if the following inequality holds for all x, y in the domain of f :

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq m_f \|x - y\|^2, \forall x, y \in \text{dom } f.$$

B. Problem Formulation

Consider a set of n agents that form a connected directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ containing n nodes. The global objective of the agents is to cooperatively find the optimal solution of the problem

$$\underset{u \in \mathbb{R}^n}{\text{minimize}} \quad f(u) = \sum_{i=1}^n f_i(u), \quad (2)$$

where, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex cost function, only available to agent i , and $u \in \mathbb{R}^n$ is a decision vector. At any time instant k a node i can only exchange information with its neighboring nodes connected with a directed edge in $\mathcal{G}$. Let, the optimal value of (2) is denoted by f^* and $\mathcal{U}^* = \{u \in \mathbb{R}^n \mid \sum_{i=1}^n f_i(u) = f^*\}$, represent the set of solutions to (2). Solving problem (2) using the centralized gradient descent iteration:

$$u(k+1) = u(k) - \alpha \nabla f(u(k)) \quad (3)$$

is not possible here as f_i is only known to agent i , where $f = \sum_{i=1}^n f_i$ is not available centrally. Moreover, if we were to implement iteration (3), it would incur a significant amount of communication traffic over the network, especially if the network is sparse. To this end, we propose a distributed gradient-descent method to solve problem (2) over directed graphs. Our scheme is a combination of a finite-time consensus protocol of [22] and the gradient descent method. We will refer to our protocol as *gradient-consensus* method.

C. The Gradient-Consensus Method

We first present the main idea of our scheme before rigorously establishing results. Under the scheme presented in the article, each agent i holds two vector estimates: optimization variable $u^i(k) \in \mathbb{R}^n$ and gradient step update variable $z^i(k) \in \mathbb{R}^n$. At first step, each node makes an initial guess for its own optimal solution $u_i(0) \in \mathbb{R}$. Individual node values are propagated to the entire network using the MXP protocol (see Section III-B) such that each node has access to the initial guess of all the other nodes in the network. Thus, the initial condition of the optimization variable $u^i(k)$ for every node $i \in \mathcal{V}$ is set as:

$$u^i(0) = [u_1(0) \dots u_i(0) \dots u_n(0)]^T \quad (4)$$

where, $u_i(0) \in \mathbb{R}$ is initial guess of agent i . Each iteration $k = 1, 2, \dots$ is divided into two stages: In the first stage every individual agent performs the following update:

$$z^i(k) = u^i(k-1) - \alpha \nabla f_i(u^i(k-1)) \quad (5)$$

based on its optimization variable at the previous iteration $u^i(k-1)$ and its own gradient information. Here, $z^i(k) \in \mathbb{R}^n$ for all $i \in \mathcal{V}$ is the gradient step update variable at time instant k . Let,

$$\hat{z}(k) := \frac{1}{n} \sum_{i=1}^n z^i(k) \quad (6)$$

for all $k \geq 1$. $\hat{z}(k) \in \mathbb{R}^n$ is the average of the gradient step update variables $z^i(k)$. In the next stage, every agent uses a finite-time terminated consensus protocol denoted by ρ -consensus protocol (see Section III) to determine an estimate $\hat{z}_\rho^i(k)$ of $\hat{z}(k)$ and update its optimization variable to $\hat{z}_\rho^i(k)$,

$$u^i(k) = \hat{z}_\rho^i(k). \quad (7)$$

The ρ -consensus protocol guarantees that the estimate $\hat{z}_\rho^i(k)$ is within ρ of the actual average $\hat{z}(k)$ that is,

$$\|\hat{z}_\rho^i(k) - \hat{z}(k)\| < \rho, \text{ for all } i \in \mathcal{V}. \quad (8)$$

Thus, all agents arrive within ρ of $\hat{z}(k)$ where each agent also has a criterion (as established later in Section III) to realize when it is within ρ of the average. We further establish that over the iterations, the optimal solution can be reached with an error which is $O(\rho)$ under unrestrictive assumptions. Hence, our *gradient-consensus* method carries out an approximate centralized version of iteration (3). The *gradient-consensus* method is summarized in Algorithm 1 where, ρ -consensus($z^i(k)$, $i \in \mathcal{V}$) denotes the ρ -consensus protocol running at each agent $i \in \mathcal{V}$ to compute the estimate $\hat{z}_\rho^i(k)$ of the average $\hat{z}(k)$. We also provide the convergence results for the *gradient-consensus* presented in Algorithm 1 and characterize the rate of convergence for an arbitrary fixed step-size $\alpha > 0$ in (5). The results in this paper holds under the following assumptions.

Assumption 1:

- 1) The directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ representing the agent interconnections is strongly-connected.
- 2) Let $P = [p_{ij}]$ be the weighted adjacency matrix associated with the digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. P is a primitive, column-stochastic matrix.

Assumption 2:

- 1) The optimal value f^* is finite and the optimal solution set $\mathcal{U}^*$ is non-empty.
- 2) For all $i \in \{1, 2, \dots, n\}$, f_i is a proper closed strongly convex function with parameter $m_{f_i} > 0$. Further, each f_i is Lipschitz differentiable with parameter $L_{f_i} > 0$.

Algorithm 1: Gradient-Consensus at node $i \in \mathcal{V}$

Input:

| choose $\alpha > 0$, $\rho > 0$

Initialize:

| Pick any $u_i(0) \in \mathbb{R}$;

STEP 1:

| Using MXP protocol

| $u^i(0) := [u_1(0), u_2(0), \dots, u_n(0)]^T$, for all $i \in \mathcal{V}$

STEP 2:

| **for** $k = 1, 2, \dots$ **do**

| | $z^i(k) = u^i(k-1) - \alpha \nabla f_i(u^i(k-1))$

| | $u^i(k) \leftarrow \rho$ -consensus($z^i(k)$, $i \in \mathcal{V}$)

| **end**

III. ρ -consensus PROTOCOL

This section presents the ρ -consensus protocol. This scheme is a vectorized version of the finite time-terminated ratio consensus protocol introduced in [22]. The algorithm in [22] presents a protocol for node interaction which allows the agents to get an approximate estimate of the average of the initial scalar values in directed graph topologies in finite-time. This approximate estimate is characterized by the pre-specified tolerance value ρ . Consider a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ containing n nodes. Each node $i \in \mathcal{V}$ maintains a state vector at time k , denoted by $x^i(k) \in \mathbb{R}^n$. Let, the initial condition for the state vector of any node $i \in \mathcal{V}$ be $[x_1^i(0) \ x_2^i(0) \ \dots \ x_n^i(0)]^T$, where $x^i(0) \in \mathbb{R}^n$. The goal is to find an approximate estimate of the average of the initial state

vectors that is $\bar{x} = [\bar{x}_1 \ \bar{x}_2 \ \dots \ \bar{x}_n]^T$ where $\bar{x}_j = \sum_{i=1}^n x_j^i(0)$ for $j = 1, 2, \dots, n$. To accomplish this, each agent runs n finite-time consensus protocols of [22] corresponding to each entry of the initial vector $x^i(0)$. This strategy is termed as ρ -consensus scheme. Node i update the l^{th} , $l \in \{1, 2, \dots, n\}$ entry of its state vector at the $(k+1)^{th}$ discrete iteration according to the following update law:

$$x_l^i(k+1) = p_{ii}x_l^i(k) + \sum_{j \in N_i^-} p_{ij}x_l^j(k) \quad (9)$$

$$y_i(k+1) = p_{ii}y_i(k) + \sum_{j \in N_i^-} p_{ij}y_j(k) \quad (10)$$

$$r_l^i(k+1) = \frac{x_l^i(k+1)}{y_i(k+1)} \quad (11)$$

where, N_i^- is the set of in-neighbors of node i and $y_i(k)$ and $r_j^i(k)$ are auxiliary state values required for the scheme.

Theorem 1. Let Assumption 1 hold. Let $\{x_j^i(k)\}$, $\{y_i(k)\}$ and $\{r_j^i(k)\}$ be the sequences generated by (9), (10) and (11), respectively. Let the initial conditions for the network j^{th} state value be $[x_j^1(0) \ x_j^2(0) \ \dots \ x_j^n(0)]^T$, where $x_j^i(0) \in \mathbb{R}$, $i \in \mathcal{V}$. Then $\frac{x_j^i(k)}{y_i(k)}$ asymptotically converges to $\bar{x}_j := \lim_{k \rightarrow \infty} \frac{x_j^i(k)}{y_i(k)} = \frac{1}{n} \sum_{i=1}^n x_j^i(0)$ for all $i \in \{1, \dots, n\}$.

Proof. See [23] for Proof. $\square$

Let the maximum and minimum of the j^{th} state value of the network over all the agents at any time instant k be given as

$$M_j(k) := \max_{i \in \mathcal{V}} r_j^i(k) = \max_{i \in \mathcal{V}} \frac{x_j^i(k)}{y_i(k)}, y_i(k) \neq 0, j \in \mathcal{V}, \quad (12)$$

$$m_j(k) := \min_{i \in \mathcal{V}} r_j^i(k) = \min_{i \in \mathcal{V}} \frac{x_j^i(k)}{y_i(k)}, y_i(k) \neq 0, j \in \mathcal{V}, \quad (13)$$

Define $\bar{M}(k) = [M_1(k) \ M_2(k) \ \dots \ M_n(k)]^T$, $\bar{m}(k) = [m_1(k) \ m_2(k) \ \dots \ m_n(k)]^T$, respectively, where $M_j(k) \in \mathbb{R}$ and $m_j(k) \in \mathbb{R}$ for all $j \in \mathcal{V}$ are as defined above.

Theorem 2. Let Assumption 1 hold. Let $\{M_j(k)\}$, $\{m_j(k)\}$ be the sequences generated by (12), (13), respectively. Consider, the sub-sequences $\{M_j(lD)\}$ and $\{m_j(lD)\}$, $l = 0, 1, \dots$, where, D is the upper bound on the diameter of $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Then, $\{M_j(lD)\}$ and $\{m_j(lD)\}$ are strictly monotonic sequences converging to $\bar{x}_j$ for all $j \in \mathcal{V}$.

Proof. See [22] for Proof. $\square$

A. Distributed Finite-Time Terminated ρ -consensus Algorithm

Here, we propose an algorithm using Theorem 2 which allows each node to simultaneously converge to an approximate estimate of $\bar{x}$ within a pre-specified threshold ρ . For $k = 0, 1, \dots$, define $\bar{\alpha}^i(kD) := \bar{M}(kD)$, $\bar{\alpha}^i(kD) = \bar{m}(kD)$ and $\beta^i(kD) := \bar{\alpha}^i(kD) - \bar{\alpha}^i(kD)$. Each node computes the infinity norm of the vector $\beta^i(kD)$ at each epoch and compares it with ρ . That is, if $\|\beta^i(kD)\|_\infty := \max_{j=1,2,\dots,n} |M_j(kD) - m_j(kD)| < \rho$ then each node terminates the consensus protocol. Details of this scheme are given in Algorithm 2.

Algorithm 2: Finite-time terminated ρ -consensus at each node $i \in \mathcal{V}$

Input:
 choose $\rho > 0$;
 $x^i(0) := [x_1(0) \ x_2(0) \ \dots \ x_n(0)]^T$

Initialize:
 $y_i(k) = 1$; $z^i := x^i(0)/y_i(0)$; $w^i := x^i(0)/y_i(0)$;
 $k = 0$; $u = 1$;

Repeat:
for $j = 1, 2, 3, \dots, n$ **do**
 $x_j^i(k+1) = \sum_{l \in N_i^- \cup \{i\}} p_{il} x_j^l(k)$;
 $z_j^i = \max_{l \in N_i^- \cup \{i\}} z_j^l$, $w_j^i = \min_{l \in N_i^- \cup \{i\}} w_j^l$;
end
 $y_i(k+1) = \sum_{j \in N_i^- \cup \{i\}} p_{ij} y_j(k)$;
 // z_j^i is the j^{th} entry of z^i
if $k = uD$ **then**
 $\bar{\alpha}^i := z^i$; $\bar{\alpha}^i := w^i$; $\beta^i := \bar{\alpha}^i - \underline{\alpha}^i$;
if $\|\beta^i\|_\infty < \rho$ **then**
break ; // converged
else
 set $z^i = \frac{x^i(uD)}{y_i(uD)} = w^i$, $u = u + 1$;
end
 $k = k + 1$;

Theorem 3. Algorithm 2 converges in finite-time simultaneously at each node.

Proof. From [22], it follows that for any given $\rho > 0$, there exists an integer $t_j(\rho)$ such that for all $k \geq t_j(\rho)$, $|M_j(kD) - m_j(kD)| < \rho$ for $j \in \mathcal{V}$. Let $t(\rho) := \max_{j \in \mathcal{V}} t_j(\rho)$, then for all $k \geq t(\rho)$, $|M_j(kD) - m_j(kD)| < \rho$ for all $j \in \mathcal{V}$. That is, $\|\beta^i(kD)\|_\infty < \rho$. As each node has access to $\bar{M}(kD)$ and $\bar{m}(kD)$, convergence is detected simultaneously by each node at the same iteration. $\square$

Remark 3.1. Notice that using the above protocol, each node has access to the approximate consensus value simultaneously in finite time. Further, each node does not need to know the actual diameter but some upper bound. In most applications, an upper bound on the diameter is readily available.

B. Maximum Consensus Protocol

Consider a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ containing n nodes. Each node $i \in \mathcal{V}$ maintains a state at time k , denoted by $x^i(k) \in \mathbb{R}^n$. Let the initial condition for the state of any node $i \in \mathcal{V}$ be $x^i(0) = [-\infty \dots x_i \dots -\infty]^T$, where $x_i \in \mathbb{R}$. Let,

$$M_j := \max_{i \in \mathcal{V}} x_j^i(0), \quad i, j \in \mathcal{V} \quad (14)$$

Let $M_j \in \mathbb{R}$ be as defined above then $\bar{M} := [M_1 \dots M_n]^T = [x_1 \dots x_n]^T$. The Maximum Consensus Protocol denoted as MXP protocol computes $\bar{M}$ in a distributed manner. It generates a sequence of estimates based on the following update rules. For any node $i \in \mathcal{V}$,

$$x^i(k+1) = \max_{s \in N_i^- \cup \{i\}} (x^s(k)) \quad (15)$$

where, the maximum is taken row-wise.

Proposition 1. Let Assumptions of Theorem 1 hold, then the MXP protocol of (15) converges to $\bar{M}$ in finite time $k \leq D$.

Proof. From [24], j^{th} row of $x^i(k)$ converges to M_j in finite steps $k \leq D$. Thus, $x^i(k)$ converges to $\bar{M}$ in finite time steps $k \leq D$. $\square$

Thus, by setting $x_i = u_i(0)$ in gradient-consensus method, $x^i(k)$ will converge to $u^i(0)$ (from (4)) in $k \leq D$ time instants.

C. Main Result

Now, we present the main result of the article. Under Assumption 1 and Assumption 2, the sequence of estimates generated by the gradient-consensus method of Algorithm 1 for all nodes converges to the optimal solution u^* within $O(\rho)$ where ρ is the tolerance for the finite time consensus step. Moreover the convergence is provable to be geometric. In particular,

Theorem 4. Let Assumptions 1 and 2 hold. Let $\{u^i(k)\}$ be the sequence of estimates generated by the gradient-consensus method of Algorithm 1 for all $i \in \mathcal{V}$. Then, there exists constants a_1, a_2 such that if $\alpha \leq a_1$ then,

$$\|u^i(k) - u^*\| = a_2^k \|u^i(0) - u^*\| + O(\rho).$$

Proof. (Sketch) We provide a sketch of the proof. The detailed proof is omitted due to space constraints. The key thing is to bound the norm of the difference between $\frac{1}{n} \sum_{i=1}^n u^i(k)$ and the individual estimates $u^i(k)$ at any iteration k . Observing that each function f_i is Lipschitz differentiable and strongly convex it can be shown that

$$\left\| \frac{1}{n} \sum_{i=1}^n u^i(k) - u^i(k) \right\|$$

is bounded. Using this one can get the desired result. $\square$

IV. SIMULATION RESULTS

In this section, we present the results obtained by applying the gradient-consensus protocol of Algorithm 1 to solve two distributed optimization problems. All the simulations shown here are performed on a randomly generated directed graph consisting of 10 nodes.

A. Distributed Least Squares

We apply Algorithm 1 to the following least squares problem

$$\underset{u \in \mathbb{R}^{10}}{\text{minimize}} \frac{1}{2} \|b - Au\|^2 = \sum_{i=1}^{10} \frac{1}{2} \|b_i - A_i u\|^2.$$

Here, each agent has a measurement $b_i \in \mathbb{R}^{10}$. $A_i \in \mathbb{R}^{10 \times 10}$ is the measurement scaling matrix of agent i , the entries of which are chosen from the i.i.d samples of $\mathcal{N}(0, 1)$. The unknown state or the true state $u \in \mathbb{R}^{10}$ is also generated from the i.i.d samples of $\mathcal{N}(0, 1)$. Fig. 1 shows the convergence of the gradient-consensus algorithm for five different values of the step-size α . It can be seen that the

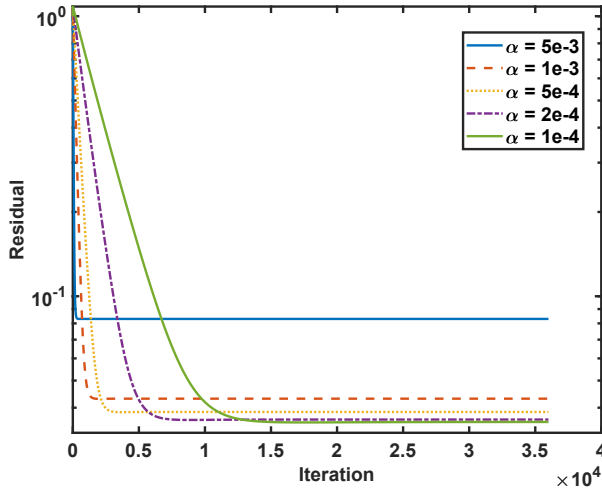


Fig. 1: Plot of residuals $\frac{\|u^i(k) - u^*\|}{\|u^i(0) - u^*\|}$ for all $i \in \mathcal{V}$ for different values of α .

residuals $\frac{\|u^i(k) - u^*\|}{\|u^i(0) - u^*\|}$ for all $i \in \mathcal{V}$ reduces with a geometric rate until reaching a neighborhood of $O(\rho)$. This agrees with the convergence result agrees with Theorem 4. A smaller value of step-size α reduces the final error between the estimate and the optimal value, but at the same time slows down the convergence. Next, we compare the performance of the distributed gradient descent (DGD) algorithm of [6] with the *gradient-consensus* protocol. We observe that with a fixed step-size α both DGD and *gradient-consensus* follow a similar rate of convergence in the first 1000 iterations as shown in Fig 2. After, that DGD stalls and converges with a large error. The *gradient-consensus* method has an error between the agents' local estimates and the optimal solution an order of magnitude less even with a higher value of step-size $\alpha = 5e^{-3}$ compared to the DGD protocol with a lot lesser step-size $\alpha = 1e^{-4}$. Fig. 3 shows the convergence with varying ρ . As expected, residual decreases as ρ is decreased.

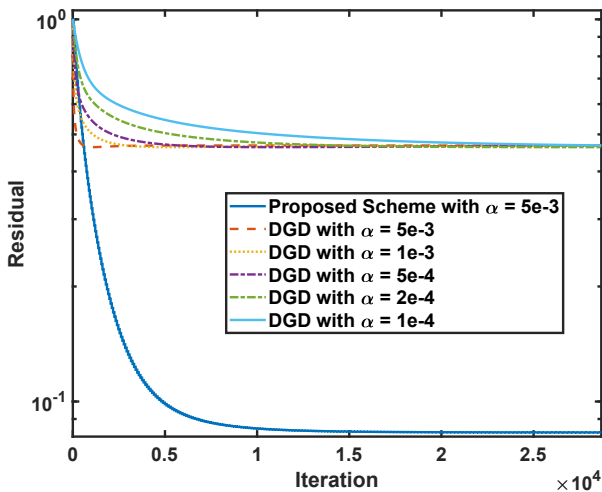


Fig. 2: Comparison between DGD of [6] using different step-sizes with *gradient-consensus* using a fixed $\alpha = 5e^{-3}$.

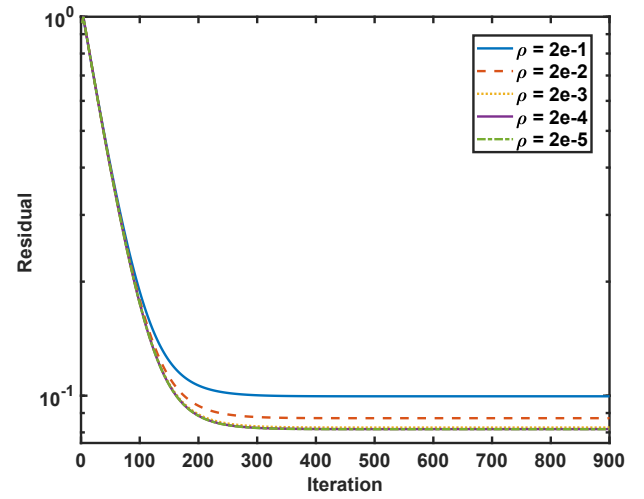


Fig. 3: Plot of residuals $\frac{\|u^i(k) - u^*\|}{\|u^i(0) - u^*\|}$ for all $i \in \mathcal{V}$ for different values of ρ .

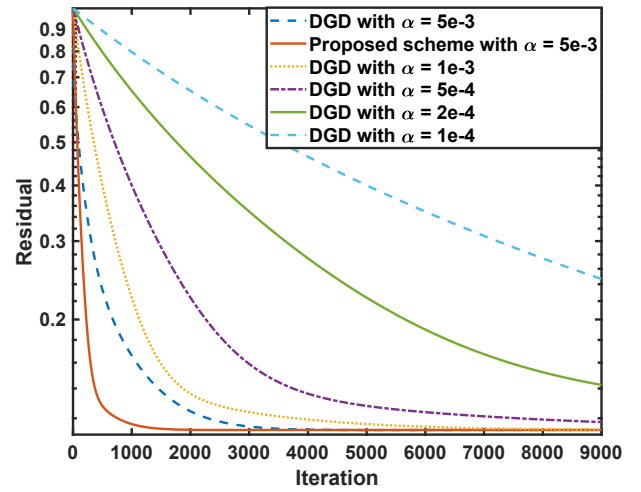


Fig. 4: Comparison between DGD of [6] with *gradient-consensus* for solving (16).

B. Distributed Optimization Problem

Here, we apply Algorithm 1 on a different distributed optimization problem. The private convex objective function of each node is: $f_i(u) = u^T Q^i u$ where, $Q^i \in \mathbb{R}^{10 \times 10}$ matrix of agent i , the entries of which are as follows: for all $i, j, k \in \mathcal{V}$,

$$Q_{jk} = \begin{cases} q_i, & \text{if } j = k = i \\ 0, & \text{otherwise} \end{cases}$$

Then, the function f can be written as $f(u) = \sum_{i=1}^n f_i(u) = u^T Q u$, where $Q = \text{diag}(q_1, q_2, \dots, q_{10}) \in \mathbb{R}^{10 \times 10}$. Note in this case the function f is a strongly convex function in the vector $u \in \mathbb{R}^{10}$ but the individual f_i 's are not. The entries $q_i \in \mathbb{R}$ are generated from the i.i.d samples of $\mathcal{N}(0, 1)$. Therefore, the distributed optimization problem becomes:

$$\underset{u \in \mathbb{R}^{10}}{\text{minimize}} \quad u^T Q u. \quad (16)$$

Fig. 4 provides a comparison between the performance of the distributed gradient descent (DGD) algorithm of [6] with the *gradient-consensus* protocol applied to problem (16). It can be seen that the *gradient-consensus* method achieves better rates of convergence compared to the DGD method of [6].

Remark 4.1. *Note that, the number of iterations on the x-axis of each plot include the additional iterations required for finite time termination of ρ -consensus at each step. Further, in comparison with the DGD algorithm, time steps needed for ρ -consensus algorithm are much smaller.*

V. CONCLUSION

In this article, we present a protocol for solving an optimization problem defined over a network in a distributed manner. Using the finite-time terminated ρ -consensus protocol, we developed a *gradient-consensus* protocol. We prove that the proposed *gradient-consensus* algorithm converges to the optimal solution at a geometric rate if the function f is assumed to be strongly convex and Lipschitz-differentiable. The effectiveness of the *gradient-consensus* algorithm is demonstrated by solving a distributed least squares problem and a quadratic optimization problem. The *gradient-consensus* showed better performance compared to the distributed gradient descent in [6]. In an ongoing journal paper [25], we are extending the analysis of the *gradient-consensus* algorithm to the case when the all the function f_i 's are not necessarily strongly convex and would include the proofs which are omitted here due to space constraints. An extensive performance comparison of the *gradient-consensus* with the existing schemes in the literature is also a future work of this article.

REFERENCES

- [1] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 20–27.
- [2] A. Olshevsky, "Efficient information aggregation strategies for distributed control and signal processing," *arXiv preprint arXiv:1009.6036*, 2010.
- [3] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "A collaborative training algorithm for distributed learning," *IEEE Transactions on Information Theory*, vol. 55, no. 4, pp. 1856–1871, 2009.
- [4] J. N. Tsitsiklis, "Problems in decentralized decision making and computation." Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, Tech. Rep., 1984.
- [5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.
- [6] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, p. 48, 2009.
- [7] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1291–1306, 2010.
- [8] A. Nedic, "Asynchronous broadcast-based convex optimization over a network," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1337–1351, 2010.
- [9] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.
- [10] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2011.
- [11] I. Matei and J. S. Baras, "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 754–771, 2011.
- [12] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.
- [13] —, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3936–3947, 2016.
- [14] D. Jakovetić, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, 2014.
- [15] 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.
- [16] A. Nedic, 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.
- [17] C. Xi and U. A. Khan, "On the linear convergence of distributed optimization over directed graphs," *arXiv preprint arXiv:1510.02149*, 2015.
- [18] S. Pu, W. Shi, J. Xu, and A. Nedić, "A push-pull gradient method for distributed optimization in networks," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 3385–3390.
- [19] G. Qu and N. Li, "Accelerated distributed nesterov gradient descent," *IEEE Transactions on Automatic Control*, 2019.
- [20] R. Diestel, *Graph Theory*. Berlin, Germany: Springer-Verlag, 2006.
- [21] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.
- [22] M. Prakash, S. Talukdar, S. Attree, S. Patel, and M. V. Salapaka, "Distributed stopping criterion for ratio consensus," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2018, pp. 131–135.
- [23] A. D. Dominguez-Garcia and C. N. Hadjicostis, "Coordination and control of distributed energy resources for provision of ancillary services," in *2010 First IEEE International Conference on Smart Grid Communications*. IEEE, 2010, pp. 537–542.
- [24] V. Yadav and M. V. Salapaka, "Distributed protocol for determining when averaging consensus is reached," in *45th Annual Allerton Conf*, 2007, pp. 715–720.
- [25] V. Khatana, G. Saraswat, S. Patel, and M. V. Salapaka, "Gradient-consensus method for distributed optimization in directed networks," *Working Journal Paper*, 2020.