2020 American Control Conference
Denver, CO, USA, July 1-3, 2020

Byzantine-Resilient Distributed Optimization of Multi-Dimensional Functions

Kananart Kuwaranancharoen, Lei Xin, Shreyas Sundaram

Abstract— **The problem of distributed optimization requires a group of agents to reach agreement on a parameter that minimizes the average of their local cost functions using information received from their neighbors. While there are a variety of distributed optimization algorithms that can solve this problem, they are typically vulnerable to malicious (or "Byzantine") agents that do not follow the algorithm. Recent attempts to address this issue focus on single dimensional functions, or provide analysis under certain assumptions on the statistical properties of the functions at the agents. In this paper, we propose a resilient distributed optimization algorithm for multi-dimensional convex functions. Our scheme involves two filtering steps at each iteration of the algorithm: (1) distance-based and (2) component-wise removal of extreme states. We show that this algorithm can mitigate the impact of up to F Byzantine agents in the neighborhood of each regular node, without knowing the identities of the Byzantine agents in advance. In particular, we show that if the network topology satisfies certain conditions, all of the regular states are guaranteed to asymptotically converge to a bounded region that contains the global minimizer.**

I. Introduction

The problem of distributed optimization requires a network of agents to reach agreement on a parameter that minimizes the average of their objective functions using local information received from their neighbors. This framework is motivated by various applications including machine learning, power systems, and robotic networks, and there are a variety of approaches to solve this problem [1]–[3]. However, these existing works typically make the assumption that all agents follow the prescribed protocol; indeed, such protocols fail if even a single agent behaves in a malicious or incorrect manner [4].

A handful of recent papers have considered this problem for the case where agent misbehavior follows prescribed patterns [5], [6]. A more general (and serious) form of misbehavior is captured by the *Byzantine* adversary model from computer science, where misbehaving agents can send arbitrary (and conflicting) values to their neighbors at each iteration of the algorithm. Under such Byzantine behavior, it has been shown that it is impossible to guarantee computation of the true optimal point [4], [7]. Thus, some papers have formulated distributed optimization algorithms that allow the non-adversarial nodes to converge to a certain region surrounding the the true minimizer, regardless of the adversaries' actions [4], [8], [9]. However, the above works focus on single dimensional functions. The extension to general multi-dimensional functions remains largely open, however, since even the region containing the true minimizer of the functions is challenging to characterize in such cases [10]. The recent papers [11], [12] consider a vector version of the resilient decentralized machine learning problem by utilizing block coordinate descent. Those papers show that the states of regular nodes will converge to the statistical minimizer with high probability, but the analysis is restricted to i.i.d training data across the network.[1]

In this paper, we propose an algorithm that extends the "local-filtering" dynamics proposed in [4], [8] (for single-dimensional convex functions) to the multi-dimensional case. However, this extension is non-trivial, as simply applying the filtering operations proposed in those papers to each coordinate of the parameter vector does not appear to yield clear guarantees. Instead, we show that by having each node apply an additional filtering step on the parameter vectors that it receives from its neighbors at each step (based on the distance of those vectors from a commonly chosen reference point), one can recover certain performance guarantees in the face of Byzantine adversaries.

II. Notation and Terminology

A. General Notation

Let $\mathbb{R}$ and $\mathbb{N}$ denote the set of real and natural numbers, respectively. For $N \in \mathbb{N}$, let $[N]$ denote the set $\{1, 2, \ldots, N\}$. Vectors are taken to be column vectors, unless otherwise noted. We use $[x]_p$ to represent the p-th component of a vector x. The cardinality of a set is denoted by $|\cdot|$, and the Euclidean norm on $\mathbb{R}^d$ is denoted by $\|\cdot\|_2$. We denote by $\langle u, v\rangle$ the Euclidean inner product of u and v i.e., $\langle u, v\rangle = u^T v$ and by $\angle(u, v)$ the angle between vectors u and v i.e., $\angle(u, v) = \arccos\left(\frac{\langle u,v\rangle}{\|u\|_2\|v\|_2}\right)$. The Euclidean ball in d-dimensional space with center at x_0 and radius r is denoted by $\mathcal{B}(x_0, r) \triangleq \{x \in \mathbb{R}^d : \|x - x_0\|_2 \leq r\}$. Given a convex function $f : \mathbb{R}^d \to \mathbb{R}$, the set of subgradients of f at any point $x \in \mathbb{R}^d$ is denoted by $\partial f(x)$.

B. Graph Theory

We denote a network by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which consists of the set of nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ and the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. If $(v_i, v_j) \in \mathcal{E}$, then node v_j can receive information from node v_i. The in-neighbor and out-neighbor sets are denoted by $\mathcal{N}_i^- = \{v_j \in \mathcal{V} : (v_j, v_i) \in \mathcal{E}\}$ and $\mathcal{N}_i^+ = \{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\}$, respectively. A

This research was supported by NSF CAREER award 1653648. The authors are with the School of Electrical and Computer Engineering at Purdue University. Email: {kkuwaran,lxin,sundara2}@purdue.edu. Both of the first two authors contributed equally to this paper.

[1] We note that there is also a branch of literature pertaining to optimization in a client-server architecture [13] [14]; this differs from our setting where we consider a fully distributed network of agents.

path from node $v_i \in \mathcal{V}$ to node $v_j \in \mathcal{V}$ is a sequence of nodes $v_{k_1}, v_{k_2}, \ldots, v_{k_l}$ such that $v_{k_1} = v_i$, $v_{k_l} = v_j$ and $(v_{k_r}, v_{k_{r+1}}) \in \mathcal{E}$ for $1 \leq r \leq l-1$. Throughout the paper, the terms nodes and agents will be used interchangeably.

Definition 1: A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be rooted at node $v_i \in \mathcal{V}$ if for for all nodes $v_j \in \mathcal{V} \setminus \{v_i\}$, there is a path from v_i to v_j. A graph is said to be rooted if it is rooted at some node $v_i \in \mathcal{V}$. □

We will rely on the following definitions from [15].

Definition 2 (r-reachable set): For any given $r \in \mathbb{N}$, a subset of nodes $\mathcal{S} \subseteq \mathcal{V}$ is said to be r-reachable if there exists a node $v_i \in \mathcal{S}$ such that $|\mathcal{N}_i^- \setminus \mathcal{S}| \geq r$. □

Definition 3 (r-robust graphs): For $r \in \mathbb{N}$, a graph $\mathcal{G}$ is said to be r-robust if for all pairs of disjoint nonempty subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$, at least one of $\mathcal{S}_1$ or $\mathcal{S}_2$ is r-reachable. □

C. Adversarial Behavior

Definition 4: A node $v_j \in \mathcal{V}$ is said to be Byzantine if during each iteration of the prescribed algorithm, it is capable of sending arbitrary (and perhaps conflicting) values to different neighbors. □

The set of Byzantine nodes is denoted by $\mathcal{A} \subset \mathcal{V}$. The set of regular nodes is denoted by $\mathcal{R} = \mathcal{V} \setminus \mathcal{A}$.

The identities of the Byzantine agents are unknown to regular agents in advance. Furthermore, we allow the Byzantine agents know the entire topology of the network and functions equipped by the regular nodes (such worst case behavior is typical in the study of such adversarial models [4], [7], [11]).

Definition 5 (F-local model): For $F \in \mathbb{N}$, we say that the set of adversaries $\mathcal{A}$ is an F-local set if $|\mathcal{N}_i^- \cap \mathcal{A}| \leq F$, for all $v_i \in \mathcal{R}$. □

Thus, the F-local model captures the idea that each regular node has at most F Byzantine in-neighbors.

III. Problem formulation

Consider a group of N agents $\mathcal{V}$ interconnected over a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each agent $v_i \in \mathcal{V}$ has a local convex cost function $f_i : \mathbb{R}^d \to \mathbb{R}$. The objective is to collaboratively solve the following minimization problem:

$$\min_{x \in \mathbb{R}^d} \frac{1}{N} \sum_{v_i \in \mathcal{V}} f_i(x) \tag{1}$$

where x is the common decision variable. We assume that nodes can only communicate with their immediate neighbors to solve the above problem. However, since Byzantine nodes are allowed to send arbitrary values to their neighbors at each iteration of any algorithm, it is not possible to solve Problem (1) under such misbehavior (since one is not guaranteed to infer any information about the true functions of the Byzantine agents) [4], [7]. Thus, the optimization problem is recast into the following form:

$$\min_{x \in \mathbb{R}^d} \frac{1}{|\mathcal{R}|} \sum_{v_i \in \mathcal{R}} f_i(x). \tag{2}$$

We will now propose an algorithm that allows the regular nodes to approximately solve the above problem (as characterized later in the paper). We will make the following assumption throughout.

Assumption 1: For all $v_i \in \mathcal{V}$, the functions $f_i(x)$ are convex, and the sets $\operatorname{argmin} f_i(x)$ are non-empty and bounded.

IV. A Resilient Distributed Optimization Algorithm

The algorithm that we propose is stated as Algorithm 1. At each time-step k, each node $v_i \in \mathcal{V}$ maintains and updates a vector $x_i[k]$, which is its estimate of the solution to Problem (2). After presenting the algorithm, we describe each of the steps and the update rule.

Algorithm 1 Distance-MinMax Filtering Dynamics

Input Network $\mathcal{G}$, functions $\{f_i\}_{i=1}^N$, the parameter F

Each $v_i \in \mathcal{R}$ sets $x_i^* \leftarrow \texttt{optimize}(f_i)$
$\hat{x} \leftarrow \texttt{resilient\_consensus}\ (F,\ \{x_i^*\})$
Each $v_i \in \mathcal{R}$ sets $x_i[0] = x_i^*$
for $k \in \mathbb{N}$ **do**
  for $v_i \in \mathcal{R}$ **do** ▷ Implement in parallel
    $\texttt{broadcast}(\mathcal{N}_i^+, x_i[k])$
    $\mathcal{S}_i[k] \leftarrow \texttt{receive}(\mathcal{N}_i^-)$
    $\mathcal{S}_i^{\text{dist}}[k] \leftarrow \texttt{dist\_filter}(F,\ \hat{x},\ \mathcal{S}_i[k])$
    $\mathcal{S}_i^{\text{mm}}[k] \leftarrow \texttt{minmax\_filter}(F,\ \mathcal{S}_i^{\text{dist}}[k])$
    $z_i[k] \leftarrow \texttt{average}(\mathcal{S}_i^{\text{mm}}[k])$
    $x_i[k+1] \leftarrow \texttt{gradient}(f_i, z_i[k])$
  end for
end for

Note that Byzantine nodes do not necessarily need to follow the above algorithm, and can update their states however they wish. We now explain each step used in Algorithm 1.

1. $x_i^* \leftarrow \texttt{optimize}\ (f_i)$
   Each node $v_i \in \mathcal{R}$ finds the minimizer x_i^* of its local function f_i (using any appropriate algorithm).
2. $\hat{x} \leftarrow \texttt{resilient\_consensus}\ (F,\ \{x_i^*\})$
   The nodes run a resilient consensus algorithm to calculate a consensus point (which we term an *auxiliary point*) $\hat{x} \in \mathbb{R}^d$, with each node $v_i \in \mathcal{R}$ setting its initial vector to be its individual minimizer x_i^*. For example, d parallel versions of the the resilient scalar consensus algorithm from [15] can be applied (one for each component of the parameter vector). This is guaranteed to return a consensus value $\hat{x}$ that is in the smallest hyperrectangle containing all of the minimizers of the regular nodes' functions, regardless of the actions of any F-local set of adversaries (under the network conditions provided in the next section) [15].
3. $\texttt{broadcast}\ (\mathcal{N}_i^+, x_i[k])$
   Node $v_i \in \mathcal{R}$ broadcasts its current state $x_i[k]$ to its out-neighbors $\mathcal{N}_i^+$.

4. $\mathcal{S}_i[k] \leftarrow$ `receive`$(\mathcal{N}_i^-)$
Node $v_i \in \mathcal{R}$ receives the current states from its in-neighbors $\mathcal{N}_i^-$. So, at time step k, node v_i possesses the set of states $\mathcal{S}_i[k] \triangleq \{x_j[k] : j \in \mathcal{N}_i^-\} \cup \{x_i[k]\}$.
5. $\mathcal{S}_i^{\text{dist}}[k] \leftarrow$ `dist_filter` $(F,\ \hat{x},\ \mathcal{S}_i[k])$
Node $v_i \in \mathcal{R}$ computes

$$D_{ij}[k] \triangleq \|x_j[k] - \hat{x}\|_2 \ \text{ for } \ x_j[k] \in \mathcal{S}_i[k] \setminus \{x_i[k]\}. \tag{3}$$

Then, node v_i removes the states $x_j[k] \in \mathcal{S}_i[k] \setminus \{x_i[k]\}$ that produce the F-highest values of $D_{ij}[k]$. The remaining states $x_j[k]$ are stored in $\mathcal{S}_i^{\text{dist}}[k]$.
6. $\mathcal{S}_i^{\text{mm}}[k] \leftarrow$ `minmax_filter` $(F,\ \mathcal{S}_i^{\text{dist}}[k])$
Node v_i further removes states that have extreme values in any of their components. More specifically, node $v_i \in \mathcal{R}$ removes the state $x_j[k] \in \mathcal{S}_i^{\text{dist}}[k] \setminus \{x_i[k]\}$ if there exists $p \in [d]$ such that $[x_j[k]]_p$ is in the F-highest or F-lowest values from the set of scalars $\big\{[x_l[k]]_p : x_l[k] \in \mathcal{S}_i^{\text{dist}}[k] \setminus \{x_i[k]\}\big\}$. The remaining states are stored in $\mathcal{S}_i^{\text{mm}}[k]$.
7. $z_i[k] \leftarrow$ `average` $(\mathcal{S}_i^{\text{mm}}[k])$
Each node $v_i \in \mathcal{R}$ computes

$$z_i[k] = \sum_{v_j \in \mathcal{N}_i^{\text{mm}}[k]} \frac{1}{|\mathcal{S}_i^{\text{mm}}[k]|} x_j[k] \tag{4}$$

where $\mathcal{N}_i^{\text{mm}}[k] \triangleq \{v_s : x_s[k] \in \mathcal{S}_i^{\text{mm}}[k]\}$.
8. $x_i[k+1] \leftarrow$ `gradient` $(f_i, z_i[k])$
Node $v_i \in \mathcal{R}$ computes the gradient update as follows:

$$x_i[k+1] = z_i[k] - \eta[k] g_i[k] \tag{5}$$

where $g_i[k] \in \partial f_i(z_i[k])$ and $\eta[k]$ is the step-size at time-step k.

Remark 1: The role of the auxiliary point $\hat{x}$ is to give the states $x_i[k]$ a "sense of good direction". In other words, since adversarial nodes can try to pull the regular nodes away from the true minimizer, the auxiliary point provides a common reference point for each regular node with which to evaluate the states in its neighborhood. This motivates the distance-based filtering step of Algorithm 1 (in line 8), which removes the F states that are furthest away from the auxiliary point at each time-step. Note that the auxiliary point will, in general, be different from the true minimizer of interest. We also note that the resilient consensus algorithm from [15] for computing the auxiliary point will only provide asymptotic consensus in general. The regular agents will converge to consensus exponentially fast under that algorithm, however, and thus we expect that running the resilient consensus algorithm to compute the auxiliary point simultaneously with the other update steps (given by lines 6-11) will also lead to the same guarantees. In the interest of space, we do not provide that analysis here. □

V. Assumptions and Main Results

We will make the following assumptions in our analysis.

Assumption 2: There exists $L > 0$ such that $\|g_i(x)\|_2 \leq L$ for all $x \in \mathbb{R}^d$ and $v_i \in \mathcal{V}$, where $g_i(x) \in \partial f_i(x)$.

Assumption 3: The step-sizes used in line 11 of Algorithm 1 satisfy $\lim_{k\to\infty} \eta[k] = 0$, $\eta[k+1] < \eta[k]$ for all k, and $\sum_{k=0}^{\infty} \eta[k] = \infty$.

Assumption 4: The underlying communication graph $\mathcal{G}$ is $\big((2d+1)F+1\big)$-robust, and the Byzantine agents form a F-local set.

We will also use the following lemma from [15].

Lemma 1: Suppose a graph $\mathcal{G}$ satisfies Assumption 4. Let $\mathcal{G}'$ be a graph obtained by removing $(2d+1)F$ or fewer incoming edges from each node in $\mathcal{G}$. Then $\mathcal{G}'$ is rooted. □

A. Convergence to Consensus

We first show that the states $x_i[k]$ of all regular nodes $v_i \in \mathcal{R}$ reach consensus under Algorithm 1.

Theorem 1 (Consensus): Under Assumptions 2, 3, and 4, $\lim_{k\to\infty} \|x_i[k] - x_j[k]\| = 0$ for all $v_i, v_j \in \mathcal{R}$. □

Proof: We will argue that all regular nodes $v_i \in \mathcal{R}$ reach consensus on each component of their vectors $x_i[k]$, which will then prove the result. For all $p \in [d]$ and for all $v_i \in \mathcal{R}$, from (5), the p-th component of the vector $x_i[k]$ evolves as

$$[x_i[k+1]]_p = [z_i[k]]_p - \eta[k][g_i[k]]_p. \tag{6}$$

From (4), the quantity $z_i[k]$ is an average of a subset of the parameter vectors from node v_i's neighborhood. In particular, the set $\mathcal{S}_i^{mm}[k]$ is obtained by removing at most $(2d+1)F$ of the vectors received from v_i's neighbors (F vectors removed by the distance based filtering in line 8, and up to $2F$ additional vectors removed by the minmax filtering step on each of the d components in line 9 of the algorithm). Thus, at each time-step k, component $[z_i[k]]_p$ is an average of at least $|\mathcal{N}_i^-| - (2d+1)F$ of v_i's neighbors values on that component. Since the graph is $((2d+1)F+1)$-robust and the Byzantine agents form an F-local set (by Assumption 4), and leveraging Lemma 1 and the fact that the term $\eta[k]g_i[k]$ asymptotically goes to zero (by Assumptions 2 and 3), we can use an identical argument as in Theorem 6.1 from [4] to show that the scalar dynamics (6) converge to consensus. ■

B. What Region Do the States Converge To?

We now analyze the trajectories of the states of the agents under Algorithm 1. We start with the following result about the quantity $z_i[k]$ calculated in line 10 of the algorithm (and given by equation (4)).

Proposition 1: For all $k \in \mathbb{N}$ and $v_i \in \mathcal{R}$, if there exists $R_i[k] \in \mathbb{R}_{\geq 0}$ such that $\|x_j[k] - \hat{x}\|_2 \leq R_i[k]$ for all $v_j \in (\mathcal{N}_i^- \cap \mathcal{R}) \cup \{v_i\}$ then $\|z_i[k] - \hat{x}\|_2 \leq R_i[k]$. □

Proof: Consider the step $\mathcal{S}_i^{\text{dist}}[k] \leftarrow$ `dist_filter` $(F,\ \hat{x},\ \mathcal{S}_i[k])$ in Algorithm 1. We will first prove the following claim. For each $v_i \in \mathcal{R}$, there exists $v_r \in (\mathcal{N}_i^- \cap \mathcal{R}) \cup \{v_i\}$ such that $\|x_j[k] - \hat{x}\|_2 \leq \|x_r[k] - \hat{x}\|_2$ for all $v_j \in \{v_s : x_s[k] \in \mathcal{S}_i^{\text{dist}}[k]\}$.

There are two possible cases. First, if the set $\mathcal{S}_i^{\text{dist}}[k]$ contains only regular nodes, we can simply choose $v_r \in (\mathcal{N}_i^- \cap \mathcal{R}) \cup \{v_i\}$ to be the node whose state $x_r[k]$ is furthest away from $\hat{x}$. Next, consider the case where $\mathcal{S}_i^{\text{dist}}[k]$ contains the states of one or more Byzantine nodes. Since

node $v_i \in \mathcal{R}$ removes the F states from $\mathcal{N}_i^-$ that are furthest away from $\hat{x}$ (in line 8 of the algorithm), and there are at most F Byzantine nodes in $\mathcal{N}_i^-$, there is at least one regular state removed by the node v_i. Let v_r be one of the regular nodes whose state is removed. We then have $D_{ir}[k] \geq D_{ij}[k]$, for all $v_j \in \{v_s : x_s[k] \in \mathcal{S}_i^{\text{dist}}[k]\}$ which proves the claim.

Consider the step $\mathcal{S}_i^{\text{mm}}[k] \leftarrow$ `minmax_filter` $(F,\ \mathcal{S}_i^{\text{dist}}[k])$ in Algorithm 1. We have that $\mathcal{S}_i^{\text{mm}}[k] \subset \mathcal{S}_i^{\text{dist}}[k]$. Then, consider the step $z_i[k] \leftarrow$ `average` $(\mathcal{S}_i^{\text{mm}}[k])$. We have

$$z_i[k] - \hat{x} = \sum_{v_j \in \mathcal{N}_i^{\text{mm}}[k]} \frac{1}{|\mathcal{S}_i^{\text{mm}}[k]|} \big(x_j[k] - \hat{x}\big).$$

Since $\|x_j[k] - \hat{x}\|_2 \leq \|x_r[k] - \hat{x}\|_2$ for all $v_j \in \mathcal{N}_i^{\text{mm}}[k]$ (where v_r is the node identified in the claim at the start of the proof), we obtain

$$\|z_i[k] - \hat{x}\|_2 \leq \sum_{v_j \in \mathcal{N}_i^{\text{mm}}[k]} \frac{1}{|\mathcal{S}_i^{\text{mm}}[k]|} \|x_j[k] - \hat{x}\|_2 \leq R_i[k].$$

■

Next, we will establish certain quantities that will be useful for our analysis of the convergence region. Since the set $\operatorname{argmin} f_i(x)$ is non-empty for all $v_i \in \mathcal{V}$ (by Assumption 1), we define $x_i^* \in \mathbb{R}^d$ to be a minimizer of $f_i(x)$ for all $v_i \in \mathcal{V}$. For $\epsilon > 0$, define

$$\mathcal{C}_i(\epsilon) \triangleq \{x \in \mathbb{R}^d : f_i(x) \leq f_i(x_i^*) + \epsilon\}. \tag{7}$$

Since the set $\operatorname{argmin} f_i(x)$ is bounded for all $v_i \in \mathcal{V}$, there exists $\delta_i(\epsilon) \in (0, \infty)$ such that $\mathcal{C}_i(\epsilon) \subseteq \mathcal{B}(x_i^*, \delta_i(\epsilon))$ for all $v_i \in \mathcal{V}$.

Proposition 2: Consider a convex function f that has bounded subgradients, and suppose the set $\operatorname{argmin} f(x)$ is non-empty and bounded. Then for all $\epsilon > 0$ there exists $\theta(\epsilon) \geq 0$ such that $\angle(-g(x),\ x^* - x) \leq \theta(\epsilon) < \frac{\pi}{2}$ for all $x \notin \mathcal{C}(\epsilon)$, $g(x) \in \partial f(x)$, and $x^* \in \operatorname{argmin} f(x)$, where $\mathcal{C}(\epsilon)$ is defined in the same way as (7). □

Proof: From the definition of convex functions, for any $x,\ y \in \mathbb{R}^d$, we have $f(y) \geq f(x) + \langle g(x),\ y - x\rangle$, where $g(x) \in \partial f(x)$. Substitute a minimizer x^* of the function f into the variable y to get

$$-\langle g(x),\ x^* - x\rangle \geq f(x) - f(x^*). \tag{8}$$

Let $\hat{\theta}(x) \triangleq \angle(g(x), x - x^*)$. The inequality (8) becomes

$$\|g(x)\|_2\ \|x^* - x\|_2 \cos\hat{\theta}(x) \geq f(x) - f(x^*).$$

Fix $\epsilon > 0$, and suppose that $x \notin \mathcal{C}(\epsilon)$. Applying $\|g(x)\|_2 \leq L$, we have

$$\cos\hat{\theta}(x) \geq \frac{f(x) - f(x^*)}{\|g(x)\|_2\ \|x^* - x\|_2} \geq \frac{f(x) - f(x^*)}{L\ \|x^* - x\|_2}. \tag{9}$$

Let $\tilde{x}$ be the point on the line connecting x^* and x such that $f(\tilde{x}) = f(x^*) + \epsilon$. We can rewrite the point x as

$$x = x^* + t(\tilde{x} - x^*) \quad \text{where} \quad t = \frac{\|x - x^*\|_2}{\|\tilde{x} - x^*\|_2} \geq 1.$$

Suppose $\mathcal{C}(\epsilon) \subseteq \mathcal{B}(x^*, \delta(\epsilon))$. Consider the term on the RHS of (9). We have

$$\begin{aligned}\frac{f(x) - f(x^*)}{\|x - x^*\|_2} &= \frac{f(x^* + t(\tilde{x} - x^*)) - f(x^*)}{t\|\tilde{x} - x^*\|_2} \\ &\geq \frac{f(x^* + t(\tilde{x} - x^*)) - f(x^*)}{t \max_{x \in \mathcal{C}(\epsilon)} \|x - x^*\|_2} \\ &\geq \frac{f(x^* + t(\tilde{x} - x^*)) - f(x^*)}{t\delta(\epsilon)}. \end{aligned} \tag{10}$$

Since the quantity $\frac{f(x^* + t(\tilde{x} - x^*)) - f(x^*)}{t}$ is non-decreasing in $t \in (0, \infty)$ [16, Lemma 2.80], the inequality (10) becomes

$$\frac{f(x) - f(x^*)}{\|x - x^*\|_2} \geq \frac{f(\tilde{x}) - f(x^*)}{\delta(\epsilon)} = \frac{\epsilon}{\delta(\epsilon)}. \tag{11}$$

Therefore, combining (9) and (11), we obtain

$$\cos\hat{\theta}(x) \geq \frac{\epsilon}{L\delta(\epsilon)}. \tag{12}$$

However, from the definition of convex functions, we have

$$f(x^*) \geq f(\tilde{x}) + \langle g(\tilde{x}),\ x^* - \tilde{x}\rangle.$$

Since $\|g(\tilde{x})\|_2 \leq L$ and $\|x^* - \tilde{x}\|_2 \leq \delta(\epsilon)$, we get

$$\epsilon = f(\tilde{x}) - f(x^*) \leq -\langle g(\tilde{x}),\ x^* - \tilde{x}\rangle \leq L\delta(\epsilon).$$

Since $\epsilon > 0$ and $\frac{\epsilon}{L\delta(\epsilon)} \leq 1$ from the above inequality, the inequality (12) becomes

$$\hat{\theta}(x) \leq \arccos\Big(\frac{\epsilon}{L\delta(\epsilon)}\Big) \triangleq \theta(\epsilon) < \frac{\pi}{2}.$$

■

From Proposition 2, if f_i satisfies Assumptions 1 and 2 for all $v_i \in \mathcal{R}$, then for all $\epsilon > 0$, we have $\angle(-g_i(x),\ x_i^* - x) \leq \theta_i(\epsilon) < \frac{\pi}{2}$ for all $x \notin \mathcal{C}_i(\epsilon)$ and $v_i \in \mathcal{R}$.

Define $\tilde{R}_i \triangleq \|x_i^* - \hat{x}\|_2$ and

$$R^* \triangleq \inf_{\epsilon > 0} \Big\{ \max_{v_i \in \mathcal{R}} \big\{ \max\{\tilde{R}_i \sec\theta_i(\epsilon),\ \tilde{R}_i + \delta_i(\epsilon)\}\big\}\Big\}. \tag{13}$$

We now come to the main result of this paper, showing that the states of all the regular nodes will asymptotically converge to a ball of radius R^* around the auxiliary point $\hat{x}$ under Algorithm 1.

Theorem 2 (Convergence): If the states are updated using Algorithm 1, and Assumptions 1, 2, 3 and 4 hold, then for all $v_i \in \mathcal{R}$, $\limsup_k \|x_i[k] - \hat{x}\|_2 \leq R^*$, regardless of the actions of any F-local set of Byzantine adversaries. □

The full proof of the theorem is somewhat long, and is thus provided in [17]. We present here a sketch of the proof. We work towards the proof of Theorem 2 in several steps. For a fixed $\epsilon > 0$ and for all $\xi \in \mathbb{R}_{>0}$, define the *convergence radius*

$$s^*(\xi, \epsilon) \triangleq \max_{v_i \in \mathcal{R}} \big\{ \max\{\tilde{R}_i \sec\theta_i(\epsilon), \tilde{R}_i + \delta_i(\epsilon)\}\big\} + \xi.$$

Throughout the paper, we will omit the dependence of ϵ in $s^*(\xi, \epsilon)$ for notational simplicity.

The key idea is that for a fixed $\xi > 0$, we partition the space $\mathbb{R}^d$ into 3 regions:

1. $\mathcal{B}(\hat{x},\ \max_{v_i \in \mathcal{R}}\{\tilde{R}_i + \delta_i\})$,

2. $\mathcal{B}(\hat{x},\ s^*(\xi)) \setminus \mathcal{B}(\hat{x},\ \max_{v_i\in\mathcal{R}}\{\tilde{R}_i+\delta_i\})$, and
3. $\mathbb{R}^d \setminus \mathcal{B}(\hat{x},\ s^*(\xi))$.

We refer to $\mathcal{B}(\hat{x},\ s^*(\xi))$ as the *convergence region* and note that $R^* = \inf_{\epsilon>0} s^*(0)$. In Steps 1, 2 and 3, we analyze the gradient update (5) for each regular agent $v_i \in \mathcal{R}$, and in Step 4, we consider the sequence of distance updates of the regular agent furthest from the auxiliary point. Finally, we will take ξ to go to zero and take the infimum of the distance bound over $\epsilon > 0$.

Step 1: First, we show that if k is sufficiently large and the state $z_i[k]$ is in Region 1 i.e.,

$$z_i[k] \in \mathcal{B}(\hat{x}, \max_{v_j\in\mathcal{R}}\{\tilde{R}_j+\delta_j\}) \subset \mathcal{B}(\hat{x}, s^*(\xi)),$$

then by applying the gradient update (5), the state $x_i[k+1]$ is still in the convergence region i.e., $x_i[k+1] \in \mathcal{B}(\hat{x}, s^*(\xi))$. To do this, we leverage the fact that the magnitude of the gradient step satisfies $\eta[k]\|g_i[k]\|_2 \leq \eta[k]L$ by Assumption 2 and $\eta[k]L$ decreases as k increases by Assumption 3.

Step 2: We find the relationship between the terms $\|x_i[k+1]-\hat{x}\|_2$ and $\|z_i[k]-\hat{x}\|_2$ which will be used in the subsequent step. Specifically, for $v_i \in \mathcal{R}$, if the state $z_i[k]$ is in region 2 or 3, i.e., $\|z_i[k]-\hat{x}\|_2 > \max_{v_j\in\mathcal{R}}\{\tilde{R}_j+\delta_j\}$, then we show

$$\|x_i[k+1]-\hat{x}\|_2^2 \leq \|z_i[k]-\hat{x}\|_2^2 - \Delta_i(\|z_i[k]-\hat{x}\|_2,\ \eta[k]\ \|g_i[k]\|_2) \quad (14)$$

where we define $\Delta_i : [\tilde{R}_i,\infty)\times\mathbb{R}_+ \to \mathbb{R}$ to be the function

$$\Delta_i(p,l) \triangleq 2l\Big(\sqrt{p^2-\tilde{R}_i^2}\cos\theta_i - \tilde{R}_i\sin\theta_i\Big) - l^2.$$

Step 3: We show that if k is sufficiently large and the state $z_i[k]$ is in Region 2 i.e., $\|z_i[k]-\hat{x}\|_2 \in \big(\max_{v_j\in\mathcal{R}}\{\tilde{R}_j+\delta_j\}, s^*(\xi)\big]$ then by applying the gradient update (5), we have that the state $x_i[k+1]$ is still in the convergence region i.e., $x_i[k+1] \in \mathcal{B}(\hat{x}, s^*(\xi))$. This is done by showing that small $\eta[k]L$ makes the RHS of (14) bounded above by $\big(s^*(\xi)\big)^2$.

From Step 1 and 3, and Proposition 1, we can conclude that for sufficiently large k, for each regular node $v_i \in \mathcal{R}$, if the state $x_i[k]$ is in the convergence region, then the state $x_i[k+1]$ is still in the convergence region, i.e., the regular states cannot leave the convergence region.

Step 4: We show that the states of all regular nodes eventually enter the convergence region. Specifically, from Proposition 1, we have that

$$\max_{v_i\in\mathcal{R}}\|z_i[k]-\hat{x}\|_2 \leq \max_{v_i\in\mathcal{R}}\|x_i[k]-\hat{x}\|_2. \quad (15)$$

On the other hand, for the agents whose state $z_i[k]$ is in Region 3, we show that the term $\Delta_i(\|z_i[k]-\hat{x}\|_2,\ \eta[k]\ \|g_i[k]\|_2)$ in (14) is bounded below as

$$\Delta_i(\|z_i[k]-\hat{x}\|_2, \eta[k]\ \|g_i[k]\|_2) > c_i\eta[k] \quad (16)$$

for sufficiently large k, where c_i is a positive constant. Incorporating (15) and (16) into (14), we can conclude that the quantity $\max_{v_i\in\mathcal{R}}\|x_i[k]-\hat{x}\|_2^2$ strictly decreases if it is greater than $s^*(\xi)$. Eventually, every state will enter the convergence region, which completes the proof.

To gain insight into the convergence region, we provide the following result.

Proposition 3: Let x^* be a solution of Problem (2). If Assumptions 1 and 2 hold, then $x^* \in \mathcal{B}(\hat{x}, R^*)$ where R^* is defined in (13). □

Proof: We will show that the summation of any subgradients of the regular nodes' functions at any point outside the region $\mathcal{B}(\hat{x}, R^*)$ cannot be zero.

Let x_0 be a point outside $\mathcal{B}(\hat{x}, R^*)$. Since $\|x_0-\hat{x}\|_2 > \max_{v_i\in\mathcal{R}}\{\tilde{R}_i+\delta_i(\epsilon)\}$ for some $\epsilon > 0$, we have that $x_0 \notin \mathcal{C}_i(\epsilon)$ for all $v_i \in \mathcal{R}$. By the definition of $\mathcal{C}_i(\epsilon)$ in (7), we have $f_i(x_0) > f_i(x_i^*)+\epsilon$ for all $v_i \in \mathcal{R}$. Since the functions f_i are convex, we obtain $g_i(x_0) \neq \mathbf{0}$ for all $v_i \in \mathcal{R}$ where $g_i(x_0) \in \partial f_i(x_0)$.

Consider the angle between the vectors $x_0-x_i^*$ and $x_0-\hat{x}$. Suppose $\tilde{R}_i > 0$; otherwise, we have $\angle(x_0-x_i^*,\ x_0-\hat{x}) = 0$. Using Lemma 2 in [17], we can bound the angle as follows:

$$\begin{aligned}\angle(x_0-x_i^*,\ x_0-\hat{x}) &\leq \max_{y\in\mathcal{B}(\hat{x},\tilde{R}_i)} \angle(x_0-y,\ x_0-\hat{x}) \\ &= \arcsin\Big(\frac{\tilde{R}_i}{\|x_0-\hat{x}\|_2}\Big).\end{aligned}$$

Since $\|x_0-\hat{x}\|_2 > \max_{v_i\in\mathcal{R}}\{\tilde{R}_i\sec\theta_i\}$ and $\arcsin(x)$ is an increasing function in $x\in[-1,1]$, we have

$$\begin{aligned}\angle(x_0-x_i^*,\ x_0-\hat{x}) &< \arcsin\Big(\frac{\tilde{R}_i}{\tilde{R}_i\sec\theta_i}\Big) \\ &= \arcsin(\cos\theta_i) \\ &= \frac{\pi}{2}-\theta_i.\end{aligned}$$

Using Proposition 2 and the inequality above, we can bound the angle between the vectors $g_i(x_0)$ and $x_0-\hat{x}$ as follows:

$$\begin{aligned}&\angle(g_i(x_0),\ x_0-\hat{x}) \\ &\leq \angle(g_i(x_0),\ x_0-x_i^*)+\angle(x_0-x_i^*,\ x_0-\hat{x}) \\ &< \theta_i + \Big(\frac{\pi}{2}-\theta_i\Big) = \frac{\pi}{2}.\end{aligned}$$

Note that the first inequality is obtained from [18, Corollary 12]. Let $u = \frac{x_0-\hat{x}}{\|x_0-\hat{x}\|_2}$. Compute the inner product

$$\begin{aligned}\Big\langle \sum_{v_i\in\mathcal{R}} g_i(x_0),\ u\Big\rangle &= \sum_{v_i\in\mathcal{R}}\langle g_i(x_0),\ u\rangle \\ &= \sum_{v_i\in\mathcal{R}}\|g_i(x_0)\|_2\cos\angle(g_i(x_0),\ x_0-\hat{x}) \\ &> 0\end{aligned}$$

since $\|g_i(x_0)\|_2 > 0$ and $\cos\angle(g_i(x_0),\ x_0-\hat{x}) > 0$ for any $v_i \in \mathcal{R}$. This implies that $\sum_{v_i\in\mathcal{R}} g_i(x_0) \neq \mathbf{0}$. Since we can arbitrarily choose $g_i(x_0)$ from the set $\partial f_i(x_0)$, we have $\mathbf{0} \notin \partial f(x_0)$ where $f(x) = \frac{1}{|\mathcal{R}|}\sum_{v_i\in\mathcal{R}} f_i(x)$. ■

Thus, Theorem 2 and Proposition 3 show that Algorithm 1 causes all regular nodes to converge to a region that also contains the true solution, regardless of the actions of any F-local set of Byzantine adversaries. The size of this region scales with the quantity R^*. Loosely speaking, this quantity becomes smaller as the minimizers of the local functions of the regular agents get closer together. More specifically,

consider a fixed $\epsilon > 0$. If the functions $f_i(x)$ are translated so that the minimizers x_i^* get closer together (i.e., $\tilde{R}_i$ is smaller and $\theta(\epsilon)$ is fixed), and the auxiliary point $\hat{x}$ is in the hyperrectangle containing the minimizers (which is the case when we run a resilient consensus algorithm such as the one in [15]) then R^* also decreases, and the state $x_i[k]$ is guaranteed to become closer to the true minimizer as k goes to infinity.

VI. Numerical Experiment

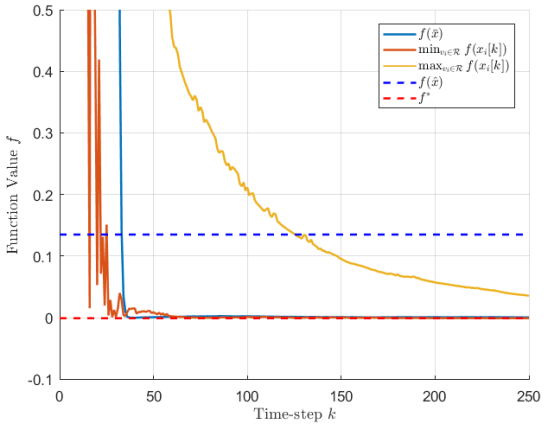

Fig. 1. Function value $f(x)$ evaluated at the average of regular agents' states (blue solid line), at the regular agent's state that gives the lowest and highest value (orange and yellow solid line respectively), at the auxiliary point (blue dashed line), and at the minimizer (red dashed line)

In our numerical experiment, we set the total number of agents to be $n = 100$ and the number of dimensions to be $d = 3$. We construct a 15-robust network in order to tolerate up to 2 Byzantine agents in the neighborhood of any regular nodes i.e., $F = 2$. Each regular node possesses a quadratic function $f_i(x) = \frac{1}{2}x^T Q_i x + b_i^T x$ where Q_i and b_i are randomly chosen with Q_i guaranteed to be positive definite.[2] Each Byzantine agent selects the transmitted vector based on the target regular node. Specifically, the transmitted vector is picked uniformly at random inside the hyper-rectangle that guarantees that its state is not discarded by the target node.

Let $f(x) \triangleq \frac{1}{|\mathcal{R}|}\sum_{v_i \in \mathcal{R}} f_i(x)$ be the objective function (Problem 2) evaluated at x, $f^* \triangleq \min_{x \in \mathbb{R}^d} f(x)$ be the optimal value of the objective function and $\bar{x}[k] \triangleq \frac{1}{|\mathcal{R}|}\sum_{v_i \in \mathcal{R}} x_i[k]$ be the average of the regular nodes' states at time-step k. In Figure 1, the objective value evaluated at the average of all regular states $f(\bar{x}[k])$ is near the optimal value f^* after 40 iterations. Furthermore, the maximum of the objective values among all regular agents $\max_{v_i \in \mathcal{R}} f(x_i[k])$ and the minimum of the objective values among all regular agents $\min_{v_i \in \mathcal{R}} f(x_i[k])$ converge to the same value, which is an implication of the fact that the regular agents reach consensus (from Theorem 1).

Remark 2: The bound in Theorem 2, for large k, does not preclude $\|\bar{x}[k] - x^*\|_2 > \|\hat{x} - x^*\|_2$, where x^* is a minimizer of f, and $f(\bar{x}[k]) > f(\hat{x})$. However, our experiments (with various settings), including the one above, show that the objective function value evaluated at the average of the states is much closer to the optimal value than that of the auxiliary point in practice, i.e., $f(\bar{x}[k]) - f^* \ll f(\hat{x}) - f^*$. □

[2]To satisfy Assumption 2 (bounded gradients), we saturate the gradients during the updates when the norms are sufficiently large.

VII. Conclusion and Future Work

In this paper, we developed a resilient distributed optimization algorithm for multi-dimensional functions. Our results guarantee that the regular states asymptotically reach consensus and enter a bounded region that contains the global minimizer, irrespective of the actions of Byzantine agents, if the network topology satisfies certain conditions. We characterized the size of this region. A promising avenue for future research would be to further refine the size of the convergence region, and to relax the conditions on the network topology.

References

[1] 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.

[2] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2013.

[3] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2011.

[4] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1063–1076, 2018.

[5] N. Ravi, A. Scaglione, and A. Nedić, "A case of distributed optimization in adversarial environment," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 5252–5256.

[6] S. X. Wu, H.-T. Wai, A. Scaglione, A. Nedić, and A. Leshem, "Data injection attack on decentralized optimization," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2018, pp. 3644–3648.

[7] L. Su and N. Vaidya, "Byzantine multi-agent optimization: Part i," *arXiv preprint arXiv:1506.04681*, 2015.

[8] L. Su and N. H. Vaidya, "Fault-tolerant multi-agent optimization: optimal iterative distributed algorithms," in *ACM Symposium on Principles of Distributed Computing*, 2016, pp. 425–434.

[9] C. Zhao, J. He, and Q.-G. Wang, "Resilient distributed optimization algorithm against adversary attacks," in *IEEE International Conference on Control & Automation (ICCA)*, 2017, pp. 473–478.

[10] K. Kuwarabancharoen and S. Sundaram, "On the location of the minimizer of the sum of two strongly convex functions," in *IEEE Conference on Decision and Control (CDC)*, 2018, pp. 1769–1774.

[11] Z. Yang and W. U. Bajwa, "Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning," *IEEE Transactions on Signal and Information Processing over Networks*, 2019.

[12] ——, "Bridge: Byzantine-resilient decentralized gradient descent," *arXiv preprint arXiv:1908.08098*, 2019.

[13] N. Gupta and N. H. Vaidya, "Byzantine fault tolerant distributed linear regression," *arXiv preprint arXiv:1903.08752*, 2019.

[14] P. Blanchard, R. Guerraoui, J. Stainer *et al.*, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, 2017, pp. 119–129.

[15] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 766–781, 2013.

[16] B. S. Mordukhovich and N. M. Nam, "An easy path to convex analysis and applications," *Synthesis Lectures on Mathematics and Statistics*, vol. 6, no. 2, pp. 1–218, 2013.

[17] K. Kuwarabancharoen, L. Xin, and S. Sundaram, "Byzantine-resilient distributed optimization of multi-dimensional functions," *arXiv preprint*, 2020.

[18] D. Castano, V. E. Paksoy, and F. Zhang, "Angles, triangle inequalities, correlation matrices and metric-preserving and subadditive functions," *Linear Algebra and its Applications*, vol. 491, pp. 15–29, 2016.