Byzantine-Resilient Multi-Agent Optimization

Lili Su and Nitin H. Vaidya *Fellow, IEEE*

***Abstract*—We consider the problem of multi-agent optimization wherein an unknown subset of agents suffer Byzantine faults and thus behave adversarially. We assume that each agent i has a local cost function f_i, and the overarching goal of the good agents is to collaboratively minimize a global objective that properly aggregates these local cost functions. To the best of our knowledge, we are among the first to study Byzantine-resilient optimization where no central coordinating agent exists, and we are the first to characterize the structures of the convex coefficients of the achievable global objectives.**

Dealing with Byzantine faults is very challenging. For example, in contrast to fault-free networks, reaching Byzantine-resilient agreement even in the simplest setting is far from trivial. We take a step towards solving the proposed Byzantine-resilient multi-agent optimization problem by focusing on scalar local cost functions. Our results might provide useful insights for the general local cost functions.

I. Introduction

Networked multi-agent systems consist of a group of agents that perform collaborative tasks. The problem of multi-agent optimization typically assumes that each agent in the network has a *local* cost function, and the overarching goal of the networked agents is to minimize a global objective that properly aggregates these local costs. One standard choice of such global objective is the average of the local cost functions [4].

Multi-agent optimization over adversary-free networks is well-studied [4]–[6]. In this paper, we study adversary-prone networks. In particular, we consider the scenario wherein an unknown subset of agents suffer Byzantine faults – a canonical fault model in distributed computing [7] – and thus behave adversarially against the good agents. Unfortunately, having the average of the local cost functions as the global objective makes the multi-agent network extremely vulnerable to Byzantine faults; the average can be completely controlled by even a single adversarial agent. Observing this, in our problem, the common goal of the good agents is to collaboratively minimize a global objective that properly aggregates the local cost functions at the *good agents*. The problem formulation (including the network model, the Byzantine fault model, and the family of desired global objectives) can be found in Section III. To the best of our knowledge, we are among the first to study Byzantine-resilient optimization where no central coordinating agent exists, and we are the first to characterize the structures of the convex coefficients of the achievable global objectives.

Dealing with Byzantine faults is very challenging; in contrast to fault-free networks, reaching Byzantine-resilient agreement even in the simplest setting is far from trivial. We take a step towards solving the proposed Byzantine-resilient multi-agent optimization problem by focusing on scalar local cost functions. Attempts have been made to generalize our results to a broader family of local cost functions [8]. We would like to leave the general local cost functions as one important future direction. Our contributions can be summarized as follows:

- We propose (in Section III-C) a Byzantine-resilient multi-agent optimization problem, wherein the global objective is a convex combination of the local cost functions at the good agents and the structure of the convex coefficients is quantified by a metric named (β,γ)–admissibility (see Definition 1). Notably, our problem formulation is also valid for general local functions.
- We show in Theorem 1 that no algorithms can guarantee (β,γ)–admissibility with $\gamma > n-\phi-b$, where n is the number of agents, ϕ is the actual number of agents suffering Byzantine faults, and b is the maximum number of Byzantine faults.
- We propose an algorithm that is provably resilient to Byzantine faults (i.e., Algorithm 1). We characterize the (β,γ) tuples it can guarantee when the network satisfies certain topological conditions (in Theorem 3). When the network is a complete graph (e.g., logically fully connected), Algorithm 1 can guarantee to achieve (β,γ) for $\beta = \frac{1}{2(n-\phi-b)}$ and $\gamma = n-\phi-b$, matching the bound in the impossibility result in terms of γ. More importantly, given $\gamma = n-\phi-b$, the achieved $\beta = \frac{1}{2(n-\phi-b)}$ matches the optimal β up to a multiplicative factor $\frac{1}{2}$. The results for complete graphs are summarized in Fig. 1. Notably, adversary-prone networks with complete graphs are distributed systems. In fact, many distributed systems assume complete graphs

Part of the results of this manuscript were presented in the unpublished technical report [1], and part of the results appeared in PODC 2016 [2]. The impossibility result was presented in the preliminary work [3]. This research is supported in part by National Science Foundation awards NSF 1329681 and 1421918.

Lili Su is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 (lilisu@mit.edu).

Nitin H. Vaidya is with the Department of Computer Science, Georgetown University, Washington, DC 20007 (nitin.vaidya@georgetown.edu).

[7].

This paper unifies the results scattered in our unpublished technical reports and the preliminary conference works [2], [9]. In particular, the impossibility result first appeared in [9] (which also appeared in the unpublished technical report [3]), and the algorithm was presented in [2] and the unpublished technical report [1], [10]. In contrast to [2] whose focus is on complete graphs, in this paper, we focus on the general networks which include complete networks as special cases.

II. Related Work

Byzantine-resilient consensus: There is a significant body of work on Byzantine-resilient consensus [11]–[17]. Readers are referred to [11] for comprehensive survey. Next we give a brief review on this line of work.

The Byzantine fault-tolerance problem was first introduced in [18], and has been one of the most fundamental problems in distributed computing for decades. Fisher, Lynch, and Paterson showed that the fault-tolerant consensus problem cannot be solved in an *asynchronous* system [19]. As one way to circumvent this impossibility result, the notion of *approximate consensus* was introduced [12], which only requires that processes agree with each other approximately rather than exactly in finite time. Reaching approximate consensus is of interest in *synchronous* systems as well [12]–[14]. The discussion in this paper applies to synchronous systems.

For undirected networks, approximate consensus can be achieved if and only if the network connectivity is at least $2b+1$ and $n > 3b$ [20]; recall that b is the maximum number of Byzantine faults. For directed networks, a sufficient and necessary condition on the network structures was characterized in [21]. There has been increasing interest in designing iterative approximate Byzantine consensus algorithms wherein only local communication is allowed [13], [14], [22], [23]. In particular, [23] studied the convergence rate over complete networks, and [13], [14] considered arbitrary directed networks and derived necessary and sufficient topological conditions. Recently, Byzantine consensus subject to differential privacy requirements was considered in [24].

All the above work focuses on scalar inputs. Multi-dimensional inputs have been studied recently [16], [25], [26]. Complete graphs were considered in [25], [26], where tight conditions on the number of agents were identified. Incomplete graphs were studied in [16].

Our work is most relevant to the line of work on *approximate* synchronous Byzantine consensus over *arbitrary graphs* [13], [14]. In our work, in each iteration, each good agent combines an approximate Byzantine consensus update with its local gradient descent update. Intuitively speaking, an approximate consensus update is used as a mechanism for each of the good agents to "robustly collect" information from others in the presence of Byzantine agents. This is in sharp contrast to the exact Byzantine consensus protocols that are involved in Blockchains wherein exact Byzantine consensus is used as a "selection mechanism" to determine among the multiple blocks proposed which one should be appended to the blockchain.

Distributed optimization: Our work goes beyond Byzantine consensus in that the local inputs are functions, and the goal is to reach an agreement on a value that is a minimizer of some weighted average of these local cost functions.

Distributed optimization has a long history. The seminal works [27], [28] considered separable global objectives for which the local decision variables at different agents are allowed to be different. Nedic and Ozdaglar [4] studied the setting wherein the global objective is the average of these local cost functions, and the local variables at the agents are require to reach consensus asymptotically. Many follow-up works are inspired; see [29], [30] for comprehensive surveys. Nevertheless, little attention has been paid to adversary-prone networks.

Multi-agent optimization over adversary-prone networks, to the best of our knowledge, was first considered by Sundaram and Gharesifard [31] and our technical reports [1], [3], [10], [32] with different fault models and global objectives. In contrast to the Byzantine fault model which assumes that a bad agent can send differently-valued messages to different neighbors, in [31] an adversarial agent is only allowed to send identically-valued (broadcast) messages. This difference is significant as in complete graphs, for the faults in [31], there exists a consensus algorithm that can tolerate less than $\frac{1}{2}$ of the agents to be faulty; in contrast, it is well-known that no consensus algorithms can tolerate more than $\frac{1}{3}$ agents to be Byzantine [33]. It might be enough to consider the fault model in [31] when the networked agents communicate with each other via wireless communication. In fact, the more structure on the adversarial behaviors, the easier to secure multi-agent optimization. Additionally, [31] considered the family of global objectives in the form of convex combinations of the local cost functions at the good agents, and no additional structures on the convex coefficients are required. Consequently, the local estimates at the good agent in [31] are only guaranteed to converge to a convex combination of the minima of those local functions. In contrast, in addition to being in the convex hull, we also characterize a structure of the convex coefficients.

III. Problem Formulation

A. Network Model

A multi-agent network consists of a collection of n agents/nodes that interact with each other through a

directed communication network $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \cdots, n\}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of edges. An unknown subset of agents, denoted by $\mathcal{A}$, are *adversarial*. This set is chosen by the system adversary and $|\mathcal{A}| \leq b$. For ease of exposition, let $\phi \triangleq |\mathcal{A}|$. Clearly, $\phi \leq b$. Additionally, we assume $n \geq 3b + 1$.

Let $\mathcal{N}_i$ be the set of incoming neighbors of agent i with $d_i \triangleq |\mathcal{N}_i|$. Each agent can send messages to itself. However, for convenience, we *exclude self-loops* from set $\mathcal{E}$, i.e., $(i, i) \notin \mathcal{E}$. The specific network structure under consideration is summarized later in Assumption 1, which ensures Byzantine-resilient consensus.

B. Fault Model

We use Byzantine fault model [7] to capture the system threat. Byzantine fault model assumes that there exists a system adversary that can choose a subset of agents $\mathcal{A}$ to compromise and control. While $\mathcal{A}$ is unknown to the good agents, a standard assumption is that the value of b is common knowledge [7]. We refer to an agent suffering Byzantine fault as Byzantine agent.

The system adversary is very powerful in that it has complete knowledge of the network, including the network structures, the local program at each good agent, the current status and running history of the multi-agent network. The Byzantine agents can collude with each other and deviate from their pre-specified local programs to arbitrarily misrepresent information to the good agents [7]. In particular, Byzantine agents can mislead the good agents by sending possibly inconsistent messages: letting $m_{ij}[t]$ be the message sent from a Byzantine agent $i \in \mathcal{A}$ to a good agent $j \in \mathcal{V} \setminus \mathcal{A}$ at iteration t, it is possible that $m_{ij}[t] \neq m_{ij'}[t]$ for $j \neq j' \in \mathcal{V} \setminus \mathcal{A}$.

Due to the freedom given to Byzantine agents as well as the system asymmetry caused by them, dealing with Byzantine faults is challenging. In particular, it is well-known that even for complete graphs and scalar local inputs, no Byzantine-resilient consensus algorithms can tolerate more than $\frac{1}{3}$ of the agents to be Byzantine [33].

C. Byzantine-Resilient Multi-Agent Optimization

Let $\mathcal{X} \subseteq \mathbb{R}$ be a nonempty, closed and convex set. We say a function $f : \mathcal{X} \to \mathbb{R}$ is *admissible* if (i) f is convex, continuously differentiable, and has L-Lipschitz gradient, (ii) f has bounded gradient, i.e., $|f'(x)| \leq L$ for each $x \in \mathcal{X}$, and (iii) $\arg\min_{x \in \mathcal{X}} f(x)$ is non-empty and compact (i.e., bounded and closed). Notably, the bounded gradient assumption holds automatically when $\mathcal{X}$ is compact. In our multi-agent optimization problem, we assume each agent i has an *admissible* local cost function f_i. Similar assumptions are adopted in [34].

In fault-free networks (i.e., $b = 0$) one commonly adopted global objective [4], [35], [36] is

$$\frac{1}{n} \sum_{i=1}^{n} f_i(x). \tag{1}$$

This objective is popular for two reasons:
(A): In statistical learning, the local functions f_is are often $i.i.d.$ generated from the same but unknown distribution p, and the goal is to minimize the population risk $\mathbb{E}[f_1]$. By standard concentration results [37], it can be shown that with high probability $\frac{1}{n}\sum_{i=1}^{n} f_i$ is uniformly close to $\mathbb{E}[f_1]$, and the minimizer of (1), under some regularity conditions, well approximates that of $\mathbb{E}[f_1]$.
(B): In general, f_is are not necessarily $i.i.d.$ generated. For example, in resource allocation [4], f_i is local cost of agent i and the convex coefficients of f_is, as a whole, can be viewed as a metric of "fairness". Alternatively, the local functions encode local information, and the convex coefficients can be interpreted as how much the "local information" at different agents are utilized in determining a global decision. In particular, with equal coefficients, the "local information" at different agents can be viewed as being utilized equally.

Unfortunately, when $b > 0$, the global objective (1) cannot be minimized. This is because initially the function f_i is the local information known to agent i only, and a Byzantine agent can lie arbitrarily about f_i. Observing this, we formulate a robust multi-agent optimization as follows: For a given tuple (β, γ), where $\beta \in (0, 1)$ and $\gamma \in \mathbb{N}$, the common goal of the good agents is to identify an output such that

$$\widetilde{x} \in \arg\min_{x \in \mathcal{X}} \sum_{i \in \mathcal{V} \setminus \mathcal{A}} \alpha_i f_i(x) \tag{2}$$

where

$$\alpha_i \geq 0, \quad \sum_{i \in \mathcal{V} \setminus \mathcal{A}} \alpha_i = 1, \quad \text{and}$$
$$\sum_{i \in \mathcal{V} \setminus \mathcal{A}} \mathbf{1}\{\alpha_i \geq \beta\} \geq \gamma.$$

Here $\mathbf{1}\{\alpha_i \geq \beta\} = 1$ if $\alpha_i \geq \beta$, and $\mathbf{1}\{\alpha_i \geq \beta\} = 0$ otherwise. Notably, the above formulation is valid even for the general setting where $\mathcal{X} \subseteq \mathbb{R}^d$ for $d \geq 1$. A similar formulation is *independently* proposed in [38, Eq. (4) on page 12] under a statistical learning setup.

Characterizing the structure of the convex coefficients is important despite the fact that different applications favor different structures on the convex coefficients. In this paper, we require that sufficiently many coefficients be not too small, i.e., are nontrivially lower bounded away from 0. Intuitively, if the coefficient assigned to a local cost function is not too small, then the information encoded is utilized in determining the global decision.

Such convex coefficients structure is crucial especially for the setting where f_is are not $i.i.d.$ generated.

Example 1. *Suppose that* $n = 4$, $\mathcal{A} = \{4\}$, $\mathcal{X} = [-1, 1]$, $f_1(x) = \frac{1}{2}x^2$,

$$f_2(x) = \begin{cases} \frac{(x-1)^2}{2}, & x > 1; \\ 0, & 0 \le x \le 1; \\ \frac{1}{2}x^2, & x < 0, \end{cases} \qquad f_3(x) = \begin{cases} \frac{1}{2}x^2, & x > 0; \\ 0, & -1 \le x \le 0; \\ \frac{(x+1)^2}{2}, & x < -1. \end{cases}$$

If we can solve (2) for $\gamma \ge 2$ *and* $\beta > 0$*, then* $\widetilde{x} = 0$.

Example 2 (Vector example)**.** *Consider the distributed state estimation problem wherein each good agent gets access to some local linear measurement* y_i *of the true state* $\theta^* \in \mathbb{R}^d$*, i.e.,* $y_i = \langle w_i, \theta^* \rangle$*, where* $w_i \in \mathbb{R}^d$ *is the local observation vector. Suppose that* $n = 4$, $\mathcal{A} = \{4\}$, $\theta^* \in \mathbb{R}^2$*, and the local observation vectors of the good agents are* $w_1^\top = [1, 0]$, $w_2^\top = [0, 1]$*, and* $w_3^\top = [1, 1]$*. Suppose we can solve (2) for* $\gamma \ge 2$ *and* $\beta > 0$*. For any such convex coefficients* α_i*s, it is easy to see that* $\sum_{i=1}^{3} \alpha_i w_i w_i^\top$ *is of full rank. Thus,* $\widetilde{x} = \theta^*$.

We use the following notion to describe the structure of the convex coefficients.

Definition 1. (β, γ)**–admissibility:** *Given* $\beta > 0$ *and* $\gamma \in \mathbb{N}$, $\boldsymbol{\alpha}^\top = [\alpha_1, \cdots, \alpha_n]$ *is* (β, γ)*-admissible if: (1)* $\sum_{i \in \mathcal{V} \setminus \mathcal{A}} \alpha_i = 1$, $\alpha_i \ge 0$*, for each* $i \in \mathcal{V} \setminus \mathcal{A}$*; and (2) at least* γ *elements of* $\boldsymbol{\alpha}$ *are lower bounded by* β.

In this paper, we take a step towards solving the Byzantine-resilient multi-agent optimization problem in (2) by focusing on scalar local cost functions. Our results might provide insights for general local cost functions.

Clearly, the problem in (2) cannot be solved for all (β, γ) tuples. The following impossibility result was originally presented in [9] (a preliminary conference version) and the unpublished technical report [3].

Theorem 1. *It is impossible to guarantee that more than* $|\mathcal{V} \setminus \mathcal{A}| - b$ *entries in vector* $\boldsymbol{\alpha}$ *are non-zero. That is, for any* $\beta > 0$*, it is impossible to guarantee that* $\boldsymbol{\alpha}$ *is* (β, γ)*-admissible with* $\gamma > |\mathcal{V} \setminus \mathcal{A}| - b$.

Proof of Theorem 1 is omitted here for brevity, and can be found in [3]. This proof is based on an indistinguishability argument that is frequently adopted in distributed computing [7]. Specifically, a collection of admissible functions that are hard to be dealt with were constructed. Notably, if additional structures on the local functions are available, the impossibility results in Theorem 1 might no longer hold.

IV. Preliminaries

A. Projection

Let $Dist\,(x, \mathcal{X})$ be the distance of x from set $\mathcal{X}$, i.e.,

$$Dist\,(x, \mathcal{X}) = \inf_{y \in \mathcal{X}} |x - y|. \tag{3}$$

We use $P_{\mathcal{X}}[x]$ to denote the projection of the point x onto the set $\mathcal{X}$, i.e., $P_{\mathcal{X}}[x] = \arg\min_{z \in \mathcal{X}} |z - x|$. Recall from Section III-C that $\mathcal{X} \subseteq \mathbb{R}$ is a nonempty, closed and convex set. In our analysis, we use the non-expansiveness property of projection, i.e.,

$$|P_{\mathcal{X}}[x] - P_{\mathcal{X}}[y]| \le |x - y| \quad \forall x, y \in \mathbb{R}. \tag{4}$$

B. Valid Global Objectives

Definition 2. *For a given tuple* (β, γ)*, let* $\mathcal{C}(\beta, \gamma)$ *be the collection of functions defined as follows:*

$$\mathcal{C}(\beta, \gamma) \triangleq \Bigg\{ p : p = \sum_{i \in \mathcal{V} \setminus \mathcal{A}} \alpha_i f_i,\ \alpha_i \ge 0,\ \sum_{i \in \mathcal{V} \setminus \mathcal{A}} \alpha_i = 1 \text{ and } \sum_{i \in \mathcal{V} \setminus \mathcal{A}} \mathbf{1}\{\alpha_i \ge \beta\} \ge \gamma \Bigg\}. \tag{5}$$

We refer to the functions in $\mathcal{C}(\beta, \gamma)$ as (β, γ)*–valid global objectives*. Define $X(\beta, \gamma)$ as

$$X(\beta, \gamma) \triangleq \cup_{p \in \mathcal{C}(\beta, \gamma)} \arg\min_{x \in \mathcal{X}} p(x). \tag{6}$$

Lemma 1. *If* $\beta \le \frac{1}{|\mathcal{V} \setminus \mathcal{A}|}$ *and* $\gamma \le |\mathcal{V} \setminus \mathcal{A}|$*, the set* $X(\beta, \gamma)$ *is convex and closed.*

Lemma 1 is proved in our full version [39]. The following quantity is used in proving Lemma 1: Define

$$\bar{X}(\beta, \gamma) \triangleq \cup_{p \in \mathcal{C}(\beta, \gamma)} \arg\min_{x \in \mathbb{R}} p(x). \tag{7}$$

We first show that $\bar{X}(\beta, \gamma)$ is convex and closed. The proof of Lemma 1 then follows easily by carefully examining two cases: (1) $\bar{X}(\beta, \gamma) \cap \mathcal{X} = \emptyset$ and (2) $\bar{X}(\beta, \gamma) \cap \mathcal{X} \neq \emptyset$.

To show an algorithm solves (2) for a given tuple (β, γ), it is enough to show the convergence of the local estimates at the good agents to set $X(\beta, \gamma)$. Notably, by requiring a common output at the good agents in (2), we implicitly require consensus among the good agents.

C. Byzantine Consensus

In this subsection, we briefly review some relevant definitions and results on Byzantine consensus.

Definition 3. *[22] For a given* $G(\mathcal{V}, \mathcal{E})$*, a reduced graph* $\mathcal{H}$ *is a subgraph obtained by removing all the Byzantine agents along with their edges; and (ii) by removing any additional up to* b *incoming edges of each good agent.*

Denote the collection of all such reduced graphs by $\mathcal{R}(G)$ with $|\mathcal{R}(G)| \triangleq \tau$.

Definition 4. *A source component S of a given graph is the collection of agents each of which has a directed path to every other agent in the graph.*

Notably, a graph has at most one source component.

Theorem 2. *[22] Byzantine-resilient consensus with scalar inputs can be achieved on $G(\mathcal{V}, \mathcal{E})$ if and only if every reduced graph has a non-empty source component.*

Throughout this paper, we assume that $G(\mathcal{V}, \mathcal{E})$ satisfies the following assumption so that Byzantine-resilient consensus with scalar inputs can be achieved.

Assumption 1. *Every reduced graph of $G(\mathcal{V}, \mathcal{E})$ has a non-empty source component.*

As stated in Theorem 2, the graph condition in Assumption 1 is both necessary and sufficient for Byzantine-resilient consensus. A similar necessary and sufficient graph condition is assumed by Sundaram and Gharesifard [31] but is stated for the restricted fault model wherein, in each iteration, a malicious agent can send only identically-valued messages to its neighbors. As a result of this difference, the graph condition in Theorem 2 is more restrictive than the graph condition stated in [31]. Note that Assumption 1 immediately implies that $d_i \geq 2b+1$ for all $i \in \mathcal{V}$ [22]. Otherwise, we might be able to find an isolated node in a reduced graph of $G(\mathcal{V}, \mathcal{E})$, contradicting Assumption 1.

V. Byzantine-Resilient Gradient Descent over Networks

We present a collaborative gradient descent method, formally described in Algorithm 1 for a good agent; a Byzantine agent may deviate from Algorithm 1 by sending out malicious and possibly inconsistent messages. In Algorithm 1, the good agents exchange both local estimates and local gradients with others. Though exchanging local gradients might not be necessary, it simplifies the exposition of our results.

In Algorithm 1, the maximum number of Byzantine faults b and the sequence of stepsizes $\{\lambda[t]\}_{t=1}^{\infty}$ are known to each good agent as a priori. Here, the stepsizes used satisfy the following conditions: (1) $\lambda[t] \geq \lambda[t+1]$ for $t \geq 1$, (2) $\sum_{t=1}^{\infty} \lambda[t] = \infty$, and (3) $\sum_{t=1}^{\infty} \lambda^2[t] < \infty$. In iteration $t \geq 1$, a good agent i sends its local estimate $x_i[t-1]$ and the local gradient $f_i'(x_i[t-1])$ to its outgoing neighbors. Define $\mathcal{D}_i^x[t]$ and $\mathcal{D}_i^g[t]$ as

$$\mathcal{D}_i^x[t] \triangleq \text{ set of estimates received from } \mathcal{N}_i,$$
$$\mathcal{D}_i^g[t] \triangleq \text{ set of received gradients } \cup \ \{f_i'(x_i[t-1])\}.$$

Note that the definitions of $\mathcal{D}_i^x[t]$ and $\mathcal{D}_i^g[t]$ are slightly different: $\mathcal{D}_i^x[t]$ does not contain the local estimate, and $\mathcal{D}_i^g[t]$ contains the local gradient. Due to the fact that a Byzantine agent can send differently-valued messages to different outgoing neighbors [7], [20], even in complete graphs it is possible that

$$\mathcal{D}_i^x[t] \neq \mathcal{D}_j^x[t], \quad \text{and} \quad \mathcal{D}_i^g[t] \neq \mathcal{D}_j^g[t], \quad \forall i \neq j \in \mathcal{V} \setminus \mathcal{A}.$$

If agent i does not receive a message from an incoming neighbor in iteration t, it must be true that this incoming neighbor is Byzantine. Thus, a default value is assumed for such missing messages. An algorithmic function, named $\mathsf{Trim}(\cdot)$, is used in our gradient descent method, and its description can be found in Algorithm 1. $\mathsf{Trim}(\cdot)$ takes a multi-set of size at least $2b+1$ as input, and removes the b largest and the b smallest elements (breaking ties arbitrarily). Note that in Algorithm 1, the averaging strategies used to compute $\widetilde{x}_i$ and $\widetilde{g}_i$ are also slightly different. In particular, $\widetilde{x}_i$ is the average of $d_i - 2b + 1$ elements in $\mathsf{Trim}(\mathcal{D}_i^x[t]) \cup \{x_i[t-1]\}$, and $\widetilde{g}_i$ is the average of the max and min elements in $\mathsf{Trim}(\mathcal{D}_i^g[t])$, where the local gradient may possibly be trimmed away. As can be seen later, these averaging strategies ensure some desired structures on $\widetilde{x}_i$ and $\widetilde{g}_i$, respectively.

Algorithm 1: Byzantine-resilient gradient descent (local program at agent $i \in \mathcal{V} \setminus \mathcal{A}$)

Input: b and $\{\lambda[t]\}_{t=1}^{\infty}$;
Initialization: Set $x_i[0]$ to an arbitrary value in $\mathcal{X}$.
for $t = 1, \ldots$ **do**
- Compute the local gradient $f_i'(x_i[t-1])$;
- Send $(x_i[t-1], f_i'(x_i[t-1]))$ to its outgoing neighbors;
- $\mathcal{D}_i^x[t] \leftarrow$ set of estimates received from incoming neighbors;
- $\mathcal{D}_i^g[t] \leftarrow$ set of received gradients $\cup \{f_i'(x_i[t-1])\}$;
- $\widetilde{x}_i[t] \leftarrow \mathsf{Average}(\mathsf{Trim}(\mathcal{D}_i^x[t]) \cup \{x_i[t-1]\})$;
- $\widetilde{g}_i[t] \leftarrow \frac{1}{2}\left(\max\{\mathsf{Trim}(\mathcal{D}_i^g[t])\} + \min\{\mathsf{Trim}(\mathcal{D}_i^g[t])\}\right)$;
- Compute $x_i[t] \leftarrow P_{\mathcal{X}}\left[\widetilde{x}_i[t] - \lambda[t]\widetilde{g}_i[t]\right]$;

end

Function $\mathsf{Trim}(\cdot)$
Input: Multi-set $\mathcal{D}$ of size $\geq 2b+1$;
- Sort the elements in $\mathcal{D}$ in a non-decreasing order (breaking ties arbitrarily);
- Remove the smallest b elements and the largest b elements;

Output: Trimmed set $\mathcal{D}$ (after $2b$ elements were removed in the previous step).

With the computed $\widetilde{x}_i[t]$ and $\widetilde{g}_i[t]$, the local estimate x_i is updated via a projected gradient descent update in the last step of the **for** loop in Algorithm 1:

$$x_i[t] \leftarrow P_{\mathcal{X}}\left[\widetilde{x}_i[t] - \lambda[t]\widetilde{g}_i[t]\right].$$

Theorem 3 (main result)**.** *Suppose Assumption 1 holds. Let* $|\mathcal{N}_i \cap \mathcal{A}| \triangleq \phi_i$, $\widetilde{\gamma} \triangleq \min_{i\in\mathcal{V}\setminus\mathcal{A}}(d_i + 1 - \phi_i - b)$, *and* $\widetilde{\beta} \triangleq \min\left\{\frac{1}{2\max_{i\in\mathcal{V}\setminus\mathcal{A}}(d_i+1-\phi_i-b)}, \frac{1}{n-\phi}\right\}$. *Then* $\lim_{t\to\infty} Dist\left(x_i[t], X(\widetilde{\beta},\widetilde{\gamma})\right) = 0$ *for* $i \in \mathcal{V}\setminus\mathcal{A}$.

Notably, Assumption 1 implies that $\max_{i\in\mathcal{V}\setminus\mathcal{A}}(d_i + 1 - \phi_i - b) > 0$.

VI. Main Analysis

In this section, we provide a proof sketch of Theorem 3. All the missing detailed proofs can be found in the full version [39]. The proof of Theorem 3 consists of two parts: achievability of consensus (Section VI-A) and convergence to set $X(\widetilde{\beta},\widetilde{\gamma})$ (Section VI-B).

A. Achievability of Consensus

Define the projection error as

$$e_i[t] = P_{\mathcal{X}}\left[\widetilde{x}_i[t] - \lambda[t]\widetilde{g}_i[t]\right] - \left(\widetilde{x}_i[t] - \lambda[t]\widetilde{g}_i[t]\right). \quad (8)$$

Thus, the update of x_i in Algorithm 1 can be written as

$$x_i[t] = \widetilde{x}_i[t] - \lambda[t]\widetilde{g}_i[t] + e_i[t]. \quad (9)$$

It turns out that the projection error diminishes over time.

Proposition 1. *For each* $i \in \mathcal{V}\setminus\mathcal{A}$ *and each* $t \geq 1$, *the projection error* $e_i[t]$ *satisfies* $|e_i[t]| \leq \lambda[t]\, L$.

Without loss of generality, assume that agents indexed from 1 through $n-\phi$ are good agents, and agents indexed from $n-\phi+1$ to n are Byzantine faulty. Let $\mathbf{x} \in \mathbb{R}^{n-\phi}$ be the vector that stacks the local estimates of the good agents with $\mathbf{x}[t]$ being these estimates at the end of iteration t, and $\mathbf{x}_i[t] = x_i[t]$ for $i \in \mathcal{V}\setminus\mathcal{A}$. Similarly, let $\widetilde{\mathbf{g}}[t] \in \mathbb{R}^{n-\phi}$ be the vector that stacks the aggregated gradients adopted by the good agents at iteration t (i.e., $\widetilde{g}_i[t], \forall\ i \in \mathcal{V}\setminus\mathcal{A}$), and let $\mathbf{e}[t] \in \mathbb{R}^{n-\phi}$ be the vector that stacks the projection errors (defined in (8)) at the good agents at iteration t.

It was shown in [22] that when $G(\mathcal{V},\mathcal{E})$ satisfies the condition in Assumption 1, the update of $\mathbf{x} \in \mathbb{R}^{n-\phi}$ can be written compactly in a matrix form:

$$\mathbf{x}[t] = \mathbf{M}[t]\mathbf{x}[t-1] - \lambda[t]\widetilde{\mathbf{g}}[t] + \mathbf{e}[t], \quad (10)$$

where $\mathbf{M}[t]$ is a row-stochastic matrix, and its dependency on t arises from the fact that the Byzantine agents can behave differently from iterations to iterations. In particular, $\widetilde{x}_i[t]$ can be written as a convex combination of $x_j[t-1]$s at agent i's good incoming neighbors, i.e.,

$$\widetilde{x}_i[t] = \sum_{j=1}^{n-\phi} \boldsymbol{M}_{ij}[t]x_j[t-1]. \quad (11)$$

The adjacency matrix $\mathbf{H}$ of a reduced graph $\mathcal{H} \in \mathcal{R}(G)$ is defined as

$$\mathbf{H}_{ij} = \begin{cases} 1, & \text{if } i = j \text{ or edge } (i,j) \text{ is contained in } \mathcal{H};\\ 0, & \text{otherwise.}\end{cases}$$

Henceforth, with a little abuse of terminology and notation, we do not distinguish a reduced graph and its adjacency matrix. The matrix $\mathbf{M}[t]$ in (11) satisfies: for every $t \geq 1$, there exists a reduced graph $\mathbf{H}[t]$ such that

$$\mathbf{M}[t] \geq \xi\, \mathbf{H}[t], \quad (12)$$

where $\xi \triangleq \frac{1}{2(d_{\max}+1-2b)}$, and $d_{\max} \triangleq \max_{i\in\mathcal{V}} d_i$.

Let $\boldsymbol{\Phi}(t,r) \triangleq \mathbf{M}[t]\mathbf{M}[t-1]\ldots\mathbf{M}[r]$ be a backward product, $\boldsymbol{\Phi}(t,t) \triangleq \mathbf{M}[t]$ and $\boldsymbol{\Phi}(t,t+1) \triangleq \mathbf{I}$. Equation (10) can be expanded out as

$$\mathbf{x}[t] = \boldsymbol{\Phi}(t,1)\mathbf{x}[0] - \sum_{r=1}^{t}\lambda[r]\boldsymbol{\Phi}(t,r+1)\widetilde{\mathbf{g}}[r] + \sum_{r=1}^{t}\boldsymbol{\Phi}(t,r+1)\mathbf{e}[r]. \quad (13)$$

Proposition 2. *[14, Proof of Theorem 1] Suppose Assumption 1 holds. Then for* $i,j,\ell \in \mathcal{V}\setminus\mathcal{A}$, *and* $r \leq t$, *it holds that* $|\boldsymbol{\Phi}_{i\ell}(t,r) - \boldsymbol{\Phi}_{j\ell}(t,r)| \leq \theta^{\lceil\frac{t-r+1}{\nu}\rceil}$, *where* $\nu \triangleq \tau(n-\phi)$ *and* $\theta \triangleq 1-\xi^{\nu}$.

Recall that $|\mathcal{R}(G)| = \tau$ and $|\mathcal{A}| = \phi \leq b$.

Lemma 2. *For all* $i,j \in \mathcal{V}\setminus\mathcal{A}$ *and for each* $t \geq 1$,

$$|x_i[t] - x_j[t]| \leq (n-\phi)\theta^{\lceil\frac{t}{\nu}\rceil}\max_{\ell\in\mathcal{V}\setminus\mathcal{A}}|x_\ell[0]| + 2L(n-\phi)\sum_{r=1}^{t}\lambda[r]\theta^{\lceil\frac{t-r}{\nu}\rceil}.$$

The second term in the above upper bound concerns the convolution of an exponentially convergent sequence with a convergent sequence. It is well-known that

$$L(n-\phi)\sum_{r=1}^{t}\lambda[r]\theta^{\lceil\frac{t-r}{\nu}\rceil} \to 0 \quad \text{as } t\to\infty.$$

A proof was given in our technical report [10] for completeness. Thus, an immediate consequence of Lemma 2 is the achievability of asymptotic consensus.

Corollary 1. *For all* $i,j \in \mathcal{V}\setminus\mathcal{A}$, $|x_i[t] - x_j[t]| \to 0$, *as* $t\to\infty$.

In addition, a convergence rate of $|x_i[t] - x_j[t]|$ for $i,j \in \mathcal{V}\setminus\mathcal{A}$ can be obtained by plugging in the specific choice of stepsizes $\{\lambda[t]\}_{t=1}^{\infty}$, such as $\lambda[t] = \frac{1}{t}\ \ \forall\, t \geq 1$.

B. Convergence Analysis

Lemma 3. *At each iteration t, for each $i \in \mathcal{V} \setminus \mathcal{A}$, there exists a valid function $p^{it} = \sum_{i \in \mathcal{V} \setminus \mathcal{A}} \alpha_j^{it} f_j$ in $\mathcal{C}(\widetilde{\beta}, \widetilde{\gamma})$, where $\{\alpha_j^{it}\}_{j \in \mathcal{V} \setminus \mathcal{A}}$ are the $(\widetilde{\beta}, \widetilde{\gamma})$–admissible convex coefficients, such that*

$$\widetilde{g}_i[t] = \sum_{j \in \mathcal{V} \setminus \mathcal{A}} \alpha_j^{it} f_j'(x_j[t-1]). \tag{14}$$

In (14), the local cost functions are evaluated at different estimates. Moreover, the valid function involved in Equation (14) is time-varying, and the local gradients at different good agent is might correspond to different valid functions ps. This is because that the Byzantine agents might behave maliciously.

To show Algorithm 1 solves (2), it is enough to show the local estimates at the good agents converge to set $X(\widetilde{\beta}, \widetilde{\gamma})$. Towards this goal, we introduce an auxiliary estimate sequence $\{z[t]\}_{t=1}^{\infty}$, defined as follows:

$$z[t] = x_{j_t}[t], \text{ where } j_t \in \arg\max_{j \in \mathcal{V} \setminus \mathcal{A}} Dist\left(x_j[t], X(\widetilde{\beta}, \widetilde{\gamma})\right).$$

By this definition, there is a sequence of agents $\{j_t\}_{t=1}^{\infty}$ associated with the auxiliary sequence $\{z[t]\}_{t=1}^{\infty}$.

Lemma 4. *The auxiliary estimate sequence $\{z[t]\}_{t=1}^{\infty}$ is asymptotically trapped in set $X(\widetilde{\beta}, \widetilde{\gamma})$, i.e., $\lim_{t \to \infty} Dist\left(z[t], X(\widetilde{\beta}, \widetilde{\gamma})\right) = 0$.*

Lemma 4 immediately implies Theorem 3.

Next we provide a high level sketch of the proof of Lemma 4. It can be shown that

$$\begin{aligned} &Dist\left(z[t], X(\widetilde{\beta}, \widetilde{\gamma})\right) \\ &\leq \max_{i \in \mathcal{V} \setminus \mathcal{A}} Dist\left(x_i[t-1] - \lambda[t]\widetilde{g}_{j_t}[t],\ X(\widetilde{\beta}, \widetilde{\gamma})\right). \end{aligned}$$

By Lemma 3, there exists a valid global objective function $p^{j_t t} = \sum_{k \in \mathcal{V} \setminus \mathcal{A}} \alpha_k f_k \in \mathcal{C}(\widetilde{\beta}, \widetilde{\gamma})$ such that

$$\widetilde{g}_{j_t}[t] = \sum_{k \in \mathcal{V} \setminus \mathcal{A}} \alpha_k f_k'(x_k[t-1]). \tag{15}$$

Let $j_t' \in \arg\max_{i \in \mathcal{V} \setminus \mathcal{A}} Dist\left(x_i[t-1] - \lambda[t]\widetilde{g}_{j_t}[t],\ X(\widetilde{\beta}, \widetilde{\gamma})\right)$. We have

$$\begin{aligned} &Dist\left(z[t], X(\widetilde{\beta}, \widetilde{\gamma})\right) \\ &\leq \inf_{y \in X(\widetilde{\beta}, \widetilde{\gamma})} \left| x_{j_t'}[t-1] - \lambda[t] \left(p^{j_t t}\right)' \left(x_{j_t'}[t-1]\right) - y \right| \\ &\quad + \lambda[t] L \max_{i,j \in \mathcal{V} \setminus \mathcal{A}} |x_i[t-1] - x_j[t-1]|. \end{aligned} \tag{16}$$

By Lemma 2, we know the second term in the right-hand side of (16) is controllable. It remains to push a recursion out of (16). Towards that, we need to consider several different cases. In particular, to ease exposition, a notion of resilient points is introduced. Detailed proof can be found in our full version [39].

VII. Discussion

In this paper, we studied multi-agent optimization over adversary-prone networks. We introduced a metric, named (β, γ)–admissibility, to quantify the "quality" of a global objective. We took a step towards solving the proposed Byzantine-resilient multi-agent optimization problem by focusing on scalar local cost functions. Unfortunately, even for this simple setting, an impossibility result (i.e., Theorem 1) was shown. We presented an algorithm that is provably robust to Byzantine faults for a certain range of (β, γ) assuming that the communication network satisfies Assumption 1, which ensures Byzantine-resilient consensus can be achieved. For the special case when the network is complete, our results achieved the optimality implied by the impossibility result in Theorem 1, summarized in Fig. 1.

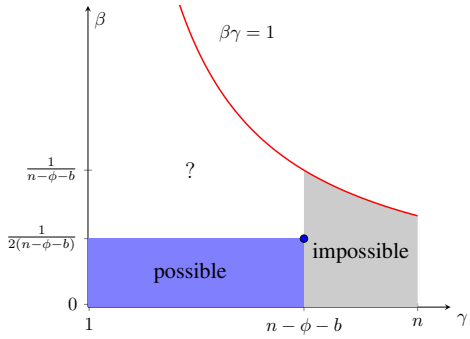

Fig. 1: Achievability of (β, γ) for complete graphs

Open problems:

- Even for complete graphs and scalar local cost functions, the problem is not completely solved. In particular, whether the white region in Fig. 1 with the question mark "?" is achievable or not is open.
- In this paper, we assumed Assumption 1. It would be interesting to investigate the minimal conditions on the communication graphs, and to study the trade-off between the (β, γ)–admissibility and the graph structures.
- We focused on asymptotic convergence. A finite-time convergence rate would be of highly practical interests.
- In our algorithm, for ease of analysis, both local estimates and local gradients are exchanged. Directly exchanging gradients might not be necessary because this gradient *information*, in a sense, is contained in agents' local estimates. It would be interesting to see whether it is possible to relax the requirement on gradient exchanges, and to quantify the gain in exchanging gradients, if any.

REFERENCES

[1] L. Su and N. H. Vaidya, "Fault-tolerant distributed optimization (Part IV): Constrained optimization with arbitrary directed networks," *arXiv preprint arXiv:1511.01821*, 2015. 1, 2

[2] ——, "Fault-tolerant multi-agent optimization: Optimal distributed algorithms," in *Proceedings of ACM Symposium on Principles of Distributed Computing*, July, 2016, pp. 425–434. 1, 2

[3] ——, "Byzantine multi-agent optimization: Part I," *arXiv preprint arXiv:1506.04681*, 2015. 1, 2, 4

[4] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *Automatic Control, IEEE Transactions on*, vol. 54, no. 1, pp. 48–61, Jan 2009. 1, 2, 3

[5] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 714–725, 2018. 1

[6] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić, "Optimal algorithms for distributed optimization," *arXiv preprint arXiv:1712.00232*, 2017. 1

[7] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann, 1996. 1, 2, 3, 4, 5

[8] Z. Yang and W. U. Bajwa, "Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning," *arXiv preprint arXiv:1708.08155*, 2017. 1

[9] L. Su and N. H. Vaidya, "Multi-agent optimization in the presence of byzantine adversaries: Fundamental limits," in *Proceedings of IEEE American Control Conference (ACC)*, July, 2016. 2, 4

[10] ——, "Byzantine multi-agent optimization: Part III," *Technical Report*, 2015. 2, 6

[11] R. Wattenhofer, *Blockchain Science: Distributed Ledger Technology*. Inverted Forest Publishing, 2019. 2

[12] D. Dolev, N. A. Lynch, S. S. Pinter, E. W. Stark, and W. E. Weihl, "Reaching approximate agreement in the presence of faults," *J. ACM*, vol. 33, no. 3, pp. 499–516, May 1986. [Online]. Available: http://doi.acm.org/10.1145/5925.5931 2

[13] H. J. LeBlanc, H. Zhang, S. Sundaram, and X. Koutsoukos, "Consensus of multi-agent networks in the presence of adversaries using only local information," in *Proceedings of the 1st International Conference on High Confidence Networked Systems*, ser. HiCoNS '12. New York, NY, USA: ACM, 2012, pp. 1–10. [Online]. Available: http://doi.acm.org/10.1145/2185505.2185507 2

[14] N. H. Vaidya, L. Tseng, and G. Liang, "Iterative approximate byzantine consensus in arbitrary directed graphs," in *Proceedings of the 2012 ACM symposium on Principles of distributed computing*. ACM, 2012, pp. 365–374. 2, 6

[15] R. Friedman, A. Mostefaoui, S. Rajsbaum, and M. Raynal, "Asynchronous agreement and its relation with error-correcting codes," *Computers, IEEE Transactions on*, vol. 56, no. 7, pp. 865–875, 2007. 2

[16] N. H. Vaidya, "Iterative byzantine vector consensus in incomplete graphs," in *Distributed Computing and Networking*. Springer, 2014, pp. 14–28. 2

[17] L. Tseng and N. H. Vaidya, "Iterative approximate consensus in the presence of byzantine link failures," in *Networked Systems*. Springer International Publishing, 2014, pp. 84–98. 2

[18] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *J. ACM*, vol. 27, no. 2, pp. 228–234, Apr. 1980. [Online]. Available: http://doi.acm.org/10.1145/322186.322188 2

[19] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, pp. 374–382, April 1985. [Online]. Available: http://doi.acm.org/10.1145/3149.214121 2

[20] M. J. Fischer, N. A. Lynch, and M. Merritt, "Easy impossibility proofs for distributed consensus problems," in *Proceedings of the fourth annual ACM symposium on Principles of distributed computing*, ser. PODC '85. New York, NY, USA: ACM, 1985, pp. 59–70. [Online]. Available: http://doi.acm.org/10.1145/323596.323602 2, 5

[21] L. Tseng and N. H. Vaidya, "Fault-tolerant consensus in directed graphs," in *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*. ACM, 2015, pp. 451–460. 2

[22] N. H. Vaidya, "Matrix representation of iterative approximate byzantine consensus in directed graphs," *arXiv preprint arXiv:1203.1888*, 2012. 2, 4, 5, 6

[23] A. D. Fekete, "Asymptotically optimal algorithms for approximate agreement," *Distributed Computing*, vol. 4, no. 1, pp. 9–29, 1990. 2

[24] D. Fiore and G. Russo, "Resilient consensus for multi-agent systems subject to differential privacy requirements," *Automatica*, vol. 106, pp. 18–26, 2019. 2

[25] H. Mendes and M. Herlihy, "Multidimensional approximate agreement in byzantine asynchronous systems," in *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, ser. STOC '13. New York, NY, USA: ACM, 2013, pp. 391–400. [Online]. Available: http://doi.acm.org/10.1145/2488608.2488657 2

[26] N. H. Vaidya and V. K. Garg, "Byzantine vector consensus in complete graphs," in *Proceedings of the 2013 ACM symposium on Principles of distributed computing*. ACM, 2013, pp. 65–73. 2

[27] J. N. Tsitsiklis, "Problems in decentralized decision making and computation." DTIC Document, Tech. Rep., 1984. 2

[28] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *Automatic Control, IEEE Transactions on*, vol. 31, no. 9, pp. 803–812, Sep 1986. 2

[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011. 2

[30] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 77–103, 2018. [Online]. Available: https://doi.org/10.1146/annurev-control-060117-105131 2

[31] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, 2018. 2, 5

[32] L. Su and N. H. Vaidya, "Byzantine multi-agent optimization: Part II," *CoRR*, vol. abs/1507.01845, 2015. [Online]. Available: http://arxiv.org/abs/1507.01845 2

[33] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982. 2, 3

[34] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2506–2517, 2009. 3

[35] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *Automatic Control, IEEE Transactions on*, vol. 60, no. 3, pp. 601–615, 2015. 3

[36] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec 2012, pp. 5453–5458. 3

[37] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. 3

[38] M. Charikar, J. Steinhardt, and G. Valiant, "Learning from untrusted data," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2017, pp. 47–60. 3

[39] L. Su and N. H. Vaidya, "Byzantine-resilient multi-agent optimization," *Technical report*. [Online]. Available: https://sites.google.com/site/lilisuece/home/byzantine-resilient-multi-agent-optimization 4, 6, 7