2023 American Control Conference (ACC)
San Diego, CA, USA. May 31 - June 2, 2023

Resilient Distributed Optimization*

Jingxuan Zhu Yixuan Lin Alvaro Velasquez Ji Liu

Abstract— **This paper considers a distributed optimization problem in the presence of Byzantine agents capable of introducing untrustworthy information into the communication network. A resilient distributed subgradient algorithm is proposed based on graph redundancy and objective redundancy. It is shown that the algorithm causes all non-Byzantine agents' states to asymptotically converge to the same optimal point under appropriate assumptions. A partial convergence rate result is also provided.**

I. Introduction

Distributed optimization has attracted considerable attention and achieved remarkable success in both theory and practice. The distributed convex optimization problem was first studied in [2] where a distributed subgradient algorithm was proposed. After this, various distributed optimization algorithms have been crafted and studied; see survey papers [3]–[5]. Distributed optimization techniques are also widely applied to decentralized deep learning [6].

Information exchange between neighboring agents is necessary for a multi-agent network for distributed optimization. However, agents' states may be corrupted and they may not adhere to the designed algorithm due to faulty processes or external attacks. An agent is called *Byzantine* if it updates its state in an arbitrary, unknown manner, and can send conflicting values to different neighbors [7]. Such attacking agents can know global information of the network, play arbitrarily and strategically, and even be coordinated. Consider a network of agents in which Byzantine agents exist. An ideal resilient algorithm is the one which can lead non-Byzantine (or normal) agents to cooperatively solve the corresponding distributed optimization problem in the presence of Byzantine agents as if they do not exist. Such a resilient algorithm is highly desirable for the safety and security of multi-agent systems as faulty processes and external attacks are inevitable.

*Proofs of the main results in this paper are omitted due to space limitations; they can be found in [1] and will be included in an expanded version of the paper.

This material is based upon work supported in part by the National Science Foundation under Grant No. 2230101, by the Air Force Office of Scientific Research under award number FA9550-23-1-0175, and by Stony Brook University's Office of the Vice President for Research through a Seed Grant. J. Liu acknowledges Air Force Research Laboratory Information Directorate (AFRL/RI); part of this work took place while he was there via the 2022 Visiting Faculty Research Program (VFRP).

J. Zhu and Y. Lin are with the Department of Applied Mathematics and Statistics at Stony Brook University ({jingxuan.zhu,yixuan.lin.1}@stonybrook.edu).
A. Velasquez is with the Department of Computer Science at University of Colorado Boulder (alvaro.velasquez@colorado.edu). J. Liu is with the Department of Electrical and Computer Engineering at Stony Brook University (ji.liu@stonybrook.edu).

Resilient distributed optimization has recently received increasing attention, probably originating from the work of [8]. Almost all the existing works cannot guarantee full resilience; what they can guarantee is all normal agents' states converge to a bounded neighborhood of the desired optimal point whose bound is not controllable [9]–[13], or an optimal point of an unspecified convex combination of all normal agents' objective functions [8], [14], [15], or a convex combination of all normal agents' local optimal points [16]. The only exceptions are [17]–[19] in which the underlying communication graph is assumed to be a complete graph, namely, each agent is allowed to communicate with all other agents. All [17]–[19] rely on the idea of "objective function redundancy". The idea has also been applied to the federated setting and achieved full resilience [20], [21]. In the federated setting, a central coordinator agent is able to communicate with all worker agents, which is more or less equivalent to a complete graph in the distributed setting (or sometimes called decentralized setting). It is worth noting that [8], [14]–[16], [19] only consider special one-dimensional optimization.

Resilient distributed optimization is also related to resilient federated optimization/learning in the coordinator-workers setting (e.g., [21]–[23]), which has attracted increasing attention recently. The key problem is how the central coordinator aggregates the received information to eliminate or attenuate the effects of Byzantine worker agents. Various Byzantine-resilient information aggregation methods have been proposed for high-dimensional optimization/learning, focusing on stochastic gradient descent (SGD); see an overview of recent developments in this area in [24]. It is doubtable that these methods can be applied to achieve full resilience in the distributed setting.

From the preceding discussion, and to the best of our knowledge, a fully resilient distributed optimization algorithm for general non-complete communication graphs does not exist, even for one-dimensional optimization problems. This gap is precisely what we study in this paper. We consider a distributed convex optimization problem in the presence of Byzantine agents and propose a fully resilient distributed subgradient algorithm based on the ideas of objective redundancy (cf. Definition 1) and graph redundancy (cf. Definition 2). The algorithm is shown to cause all non-Byzantine agents' states to asymptotically converge to the same desired optimal point under appropriate assumptions. The proposed algorithm works theoretically for multi-dimensional optimization but practically not for high-dimensional optimization, as will be explained and discussed in the concluding remarks.

This work is motivated by two recent ideas. The first is the

quantified notion of objective function redundancy proposed in [25] where a couple of different definitions of objective redundancy are studied, based on which fully resilient distributed optimization algorithms have been crafted either for a federated setting [20], [21], [25] or a distributed setting over complete graphs [17]–[19]; such redundancy has been shown necessary for achieving full resilience in multi-agent optimization [20]. We borrow one notation in [25] and further develop it. It is worth emphasizing that the results in [17]–[19] rely on objective redundancy among non-Byzantine agents, whereas ours depend on objective redundancy among all agents. This subtle difference is important for equipping a multi-agent network with a certain level of redundancy at a network design stage as which agents are non-Byzantine cannot be assumed a priori.

The second idea is so-called "Byzantine vector consensus" [26] whose goal is, given a set of both Byzantine and non-Byzantine vectors, to pick a vector lying in the convex hull of the non-Byzantine vectors, based on Tverberg's theorem [27], [28]. The idea has been very recently improved in [29] which can be used to achieve resilient multi-dimensional consensus exponentially fast. Exponential consensus is critical in the presence of diminishing disturbance [30]. We are prompted by this improved idea and utilize a resilient vector picking process, simplified from that of [29, Algorithm 1]. There are other recent approaches appealing to the idea of centerpoint [31], [32]. We expect that these approaches can also be applied to resilient optimization, provided that exponential consensus is guaranteed, e.g., in [32].

II. Problem Formulation

Consider a network consisting of n agents, labeled 1 through n for the purpose of presentation. The agents are not aware of such global labeling, but can differentiate between their neighbors. The neighbor relations among the n agents are characterized by a directed graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ whose vertices correspond to agents and whose directed edges (or arcs) depict neighbor relations, where $\mathcal{V} = \{1, \ldots, n\}$ is the vertex set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the directed edge set.[1] Specifically, agent j is an neighbor of agent i if $(j, i) \in \mathcal{E}$. Each agent can receive information from its neighbors. Thus, the directions of edges represent the directions of information flow. We use $\mathcal{N}_i$ to denote the neighbor set of agent i, excluding i, i.e., $\mathcal{N}_i = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$.

Each agent $i \in \mathcal{V}$ has a "private" convex (not necessarily differentiable) objective function, $f_i : \mathbb{R}^d \to \mathbb{R}$, only known to agent i. There exist Byzantine agents in the network which are able to transmit arbitrary values to others and capable of sending conflicting values to different neighbors at any time. The set of Byzantine agents is denoted by $\mathcal{F}$ and the set of normal (non-Byzantine) agents is denoted by $\mathcal{H}$. Which agents are Byzantine is unknown to normal agents. It is assumed that each agent may have at most β Byzantine neighbors.

[1]We use $\mathcal{A} \subset \mathcal{B}$ to denote that $\mathcal{A}$ is a subset of $\mathcal{B}$.

The goal of the normal agents is to cooperatively minimize the objective functions

$$f_{\mathcal{H}}(x) = \sum_{i \in \mathcal{H}} f_i(x) \quad \text{and} \quad f(x) = \sum_{i \in \mathcal{V}} f_i(x).$$

We will show that minimizing the above two objective functions can be achieved simultaneously with appropriate redundancy in objective functions (cf. Definition 1 and Corollary 1). It is assumed that the set of optimal solutions to f, denoted by $\mathcal{X}^*$, is nonempty and bounded.

Since each f_i is not necessarily differentiable, the gradient descent method may not be applicable. Instead, the subgradient method [33] can be applied. For a convex function $h : \mathbb{R}^d \to \mathbb{R}$, a vector $g \in \mathbb{R}^d$ is called a subgradient of h at point x if

$$h(y) \geq h(x) + g^\top (y - x) \text{ for all } y \in \mathbb{R}^d. \tag{1}$$

Such a vector g always exists and may not be unique. In the case when h is differentiable at point x, the subgradient g is unique and equals $\nabla h(x)$, the gradient of h at x. Thus, the subgradient can be viewed as a generalization of the notion of the gradient. From (1) and the Cauchy-Schwarz inequality, $h(y) - h(x) \geq -G\|y - x\|$, where G is an upper bound for the 2-norm of the subgradients of h at both x and y. We will use this fact without special mention in the sequel. Throughout this paper, we use $\|\cdot\|$ for the 2-norm.

The subgradient method was first proposed in [33] and the first distributed subgradient method was proposed in [2] for undirected graphs. Its extension to directed graphs has been studied in [34] and recently further analyzed in [35].

A. Redundancy

To make the resilient distributed optimization problem solvable, certain redundancy is necessary. We begin with objective redundancy.

Definition 1: An n-agent network is called k-redundant, $k \in \{0, 1, \ldots, n-1\}$, if for any subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$ with $|\mathcal{S}_1| = |\mathcal{S}_2| = n - k$, there holds[2]

$$\arg\min_x \sum_{i \in \mathcal{S}_1} f_i(x) = \arg\min_x \sum_{i \in \mathcal{S}_2} f_i(x).$$

The above definition of objective redundancy originated in [25, Definition 2]. It has the following properties.

Lemma 1: If an n-agent network is k-redundant, then for any subsets $\mathcal{S}, \mathcal{L} \subset \mathcal{V}$ with $|\mathcal{S}| = n - k$ and $|\mathcal{L}| \geq n - k$,

$$\arg\min_x \sum_{i \in \mathcal{S}} f_i(x) = \arg\min_x \sum_{i \in \mathcal{L}} f_i(x).$$

Proof of Lemma 1: Let $\mathcal{Z} = \arg\min_x \sum_{i \in \mathcal{S}} f_i(x)$ and $\mathcal{Q} = \{\mathcal{P} : \mathcal{P} \subset \mathcal{L},\ |\mathcal{P}| = n - k\}$. From Definition 1, $\arg\min_x \sum_{i \in \mathcal{P}} f_i(x) = \mathcal{Z}$ for any $\mathcal{P} \in \mathcal{Q}$. For each $i \in \mathcal{L}$, let $\mathcal{Q}_i = \{\mathcal{P} : \mathcal{P} \subset \mathcal{L},\ |\mathcal{P}| = n - k,\ i \in \mathcal{P}\}$. It is easy to see that for each $i \in \mathcal{L}$,[3]

$$|\mathcal{Q}_i| = q \triangleq \binom{|\mathcal{L}| - 1}{n - k - 1}.$$

[2]We use $|\mathcal{S}|$ to denote the cardinality of a set $\mathcal{S}$.

[3]$\binom{n}{k}$ denotes the number of k-combinations from a set of n elements.

Then,

$$\sum_{\mathcal{P}\in\mathcal{Q}}\sum_{i\in\mathcal{P}} f_i(x) = q\sum_{i\in\mathcal{L}} f_i(x). \tag{2}$$

Pick any $z \in \mathcal{Z}$. From (2),

$$\begin{aligned}\min_x q\sum_{i\in\mathcal{L}} f_i(x) &= \min_x \sum_{\mathcal{P}\in\mathcal{Q}}\sum_{i\in\mathcal{P}} f_i(x) \geq \sum_{\mathcal{P}\in\mathcal{Q}} \min_x \sum_{i\in\mathcal{P}} f_i(x)\\ &= \sum_{\mathcal{P}\in\mathcal{Q}}\sum_{i\in\mathcal{P}} f_i(z) = q\sum_{i\in\mathcal{L}} f_i(z),\end{aligned}$$

which implies that $z \in \arg\min_x \sum_{i\in\mathcal{L}} f_i(x)$, and thus $\mathcal{Z} \subset \arg\min_x \sum_{i\in\mathcal{L}} f_i(x)$.

To prove the lemma, it is sufficient to prove that $\arg\min_x \sum_{i\in\mathcal{L}} f_i(x) \subset \mathcal{Z}$. Suppose that, to the contrary, there exists a y such that $y \in \arg\min_x \sum_{i\in\mathcal{L}} f_i(x)$ and $y \notin Z$. Since $y, z \in \arg\min_x \sum_{i\in\mathcal{L}} f_i(x)$, from (2),

$$\begin{aligned}\sum_{i\in\mathcal{L}} f_i(y) = \sum_{i\in\mathcal{L}} f_i(z) &= \frac{1}{q}\sum_{\mathcal{P}\in\mathcal{Q}}\sum_{i\in\mathcal{P}} f_i(z)\\ &< \frac{1}{q}\sum_{\mathcal{P}\in\mathcal{Q}}\sum_{i\in\mathcal{P}} f_i(y) = \sum_{i\in\mathcal{L}} f_i(y),\end{aligned}$$

which is impossible. Thus, $\arg\min_x \sum_{i\in\mathcal{L}} f_i(x) \subset \mathcal{Z}$. ■

The following corollaries are immediate consequences of Lemma 1.

Corollary 1: If an n-agent network is k-redundant, then for any subsets $\mathcal{S} \subset \mathcal{V}$ with $|\mathcal{S}| \geq n - k$,

$$\arg\min_x \sum_{i\in\mathcal{S}} f_i(x) = \mathcal{X}^*.$$

Corollary 2: If an n-agent network is $(k+1)$-redundant with $k \geq 0$, then it is k-redundant.

We also need redundancy in graph connectivity.

A vertex i in a directed graph $\mathbb{G}$ is called a root of $\mathbb{G}$ if for each other vertex j of $\mathbb{G}$, there is a directed path from i to j. Thus, i is a root of $\mathbb{G}$ if it is the root of a directed spanning tree of $\mathbb{G}$. We will say that $\mathbb{G}$ is rooted at i if i is in fact a root. It is easy to see that a rooted graph $\mathbb{G}$ has a unique strongly connected component whose vertices are all roots of $\mathbb{G}$.

Definition 2: An (r, s)-reduced graph of a directed graph $\mathbb{G}$ with n vertices, with $r, s \geq 0$ and $r + s \leq n - 1$, is a subgraph of $\mathbb{G}$ obtained by first picking any vertex subset $\mathcal{S} \subset \mathcal{V}$ with $|\mathcal{S}| = n - r$ and then removing from each vertex of the subgraph induced by $\mathcal{S}$, $\mathbb{G}_\mathcal{S}$, arbitrary s incoming edges in $\mathbb{G}_\mathcal{S}$. A directed graph $\mathbb{G}$ is called (r, s)-resilient if all its (r, s)-reduced graphs are rooted.

It is easy to see that if a directed graph is (r_1, s_1)-resilient, then for any nonnegative $r_2 \leq r_1$ and $s_2 \leq s_1$, the graph is also (r_2, s_2)-resilient.

In the case when $r = s = \beta$, the resilient graph is equivalent to rooted "reduced graph" in [36] which was used to guarantee resilient one-dimension consensus; see Definition 4 and Theorem 2 in [36]. Thus, the definition here can be viewed as a simple generalization of the rooted "reduced graph".

Definition 2 implicitly requires that each vertex of an (r, s)-resilient graph have at least $r + s$ neighbors. More can be said.

Lemma 2: If a directed graph is (r, s)-resilient, then each of its vertices has at least $(r + s + 1)$ neighbors.

Proof of Lemma 2: Suppose that, to the contrary, there exists a vertex i in $\mathbb{G}$ whose $|\mathcal{N}_i| \leq r + s$. If $|\mathcal{N}_i| < r + s$, it is easy to see that $\mathbb{G}$ does not satisfy Definition 2. We thus consider the case when $|\mathcal{N}_i| = r + s$. Let $\mathcal{R}$ be the set of arbitrary r neighbors of vertex i, and $\mathcal{S} = \mathcal{V} \setminus \mathcal{R}$, where $\mathcal{V}$ is the vertex set of $\mathbb{G}$.[4] It is clear that $|\mathcal{S}| = n - r$, and in the subgraph induced by $\mathcal{S}$, $\mathbb{G}_\mathcal{S}$, vertex i has exactly s neighbors. Then, after vertex i removes s incoming edges in $\mathbb{G}_\mathcal{S}$, and each out-neighbor[5] of vertex i in $\mathbb{G}_\mathcal{S}$, if any, removes its incoming edge from i, vertex i becomes isolated. But it is impossible for an (r, s)-resilient graph. ■

III. Algorithm

To describe our algorithm, we need the following notation.

Let $\mathcal{A}_i$ denote the collection of all those subsets of $\mathcal{N}_i$ whose cardinality is $(d + 1)\beta + 1$. It is obvious that the number of all such subsets is

$$a_i \triangleq \binom{|\mathcal{N}_i|}{(d+1)\beta+1}, \tag{3}$$

and label them $\mathcal{A}_{i1}, \ldots, \mathcal{A}_{ia_i}$. For each $j \in \{1, \ldots, a_i\}$, let $\mathcal{B}_{ij}$ denote the collection of all those subsets of $\mathcal{A}_{ij}$ whose cardinality is $d\beta + 1$. For any subset of agents $\mathcal{S} \subset \mathcal{V}$, let $\mathcal{C}_\mathcal{S}(t)$ denote the convex hull of all $x_i(t)$, $i \in \mathcal{S}$.

Algorithm: At each discrete time $t \in \{0, 1, 2, \ldots\}$, each agent i first picks an arbitrary point

$$y_{ij}(t) \in \bigcap_{\mathcal{S}\in\mathcal{B}_{ij}} \mathcal{C}_\mathcal{S}(t) \tag{4}$$

for each $j \in \{1, \ldots, a_i\}$, and then updates its state by setting

$$v_i(t) = \frac{1}{1 + a_i}\Big(x_i(t) + \sum_{j=1}^{a_i} y_{ij}(t)\Big), \tag{5}$$

$$x_i(t + 1) = v_i(t) - \alpha(t)g_i(v_i(t)), \tag{6}$$

where $\alpha(t)$ is the stepsize and $g_i(\cdot)$ is a subgradient of $f_i(\cdot)$.

In the special one-dimensional case with $d = 1$, it is not hard to check that the steps (4) and (5) simplifies to the resilient scalar consensus algorithm in [36], which is essentially equivalent to the trimmed mean method and has been improved in [37].

The convergence and correctness of the proposed algorithm rely on the following assumptions.

Assumption 1: $\mathcal{X}^*$ has a nonempty interior.

It is easy to see that Assumption 1 implies that $f(x)$ is differentiable at any $x \in \mathrm{int}(\mathcal{X}^*)$, where $\mathrm{int}(\cdot)$ denotes the interior of a set. More can be said.

[4] We use $\mathcal{A} \setminus \mathcal{B}$ to denote the set of elements that are in $\mathcal{A}$ but not in $\mathcal{B}$.

[5] A vertex i is called an out-neighbor of vertex j if the latter is a neighbor of the former.

Lemma 3: Under Assumption 1, if the n-agent network is k-redundant with $k \geq 1$, then $f_i(x)$ is differentiable at x with $\nabla f_i(x) = 0$ for all $i \in \mathcal{V}$ and $x \in \text{int}(\mathcal{X}^*)$.

Proof of Lemma 3: Since $\text{int}(\mathcal{X}^*)$ is nonempty, for any $x^* \in \text{int}(\mathcal{X}^*)$, there exist a positive number r and an open ball in $\text{int}(\mathcal{X}^*)$ centered at x^* with radius r, denoted as $\mathcal{B}(x^*, r) \subset \text{int}(\mathcal{X}^*)$. Let h_j be a vector in $\mathbb{R}^d$ whose jth entry is ϵ and the remaining entries all equal zero. Since $x^* + h_j \in \mathcal{B}(x_0, r) \subset \text{int}(\mathcal{X}^*)$ for sufficiently small ϵ,

$$\frac{\partial}{\partial x_j} f(x^*) = \lim_{\epsilon \to 0} \frac{f(x^* + h_j) - f(x^*)}{\epsilon} = \lim_{\epsilon \to 0} \frac{\sum_{i \in \mathcal{V}} (f_i(x^* + h_j) - f_i(x^*))}{\epsilon} = 0. \quad (7)$$

For each $i \in \mathcal{V}$, since $f_i(x)$ is convex, both

$$\lim_{\epsilon \to 0^-} \frac{f_i(x^* + h_j) - f_i(x^*)}{\epsilon} \text{ and } \lim_{\epsilon \to 0^+} \frac{f_i(x^* + h_j) - f_i(x^*)}{\epsilon}$$

exist and

$$\lim_{\epsilon \to 0^-} \frac{f_i(x^* + h_j) - f_i(x^*)}{\epsilon} \leq \lim_{\epsilon \to 0^+} \frac{f_i(x^* + h_j) - f_i(x^*)}{\epsilon}$$

for all $j \in \{1, \ldots, d\}$ [38, Theorem 24.1]. It follows that

$$\sum_{k \in \mathcal{V}} \lim_{\epsilon \to 0^-} \frac{f_k(x^* + h_j) - f_k(x^*)}{\epsilon} \leq \sum_{k \in \mathcal{V}} \lim_{\epsilon \to 0^+} \frac{f_k(x^* + h_j) - f_k(x^*)}{\epsilon}.$$

Note that from (7),

$$\sum_{k \in \mathcal{V}} \lim_{\epsilon \to 0^-} \frac{f_k(x^* + h_j) - f_k(x^*)}{\epsilon} = \sum_{k \in \mathcal{V}} \lim_{\epsilon \to 0^+} \frac{f_k(x^* + h_j) - f_k(x^*)}{\epsilon}.$$

Thus,

$$\lim_{\epsilon \to 0^-} \frac{f_i(x^* + h_j) - f_i(x^*)}{\epsilon} = \lim_{\epsilon \to 0^+} \frac{f_i(x^* + h_j) - f_i(x^*)}{\epsilon},$$

i.e., $\partial f_i(x^*)/\partial x_j$ exists for all $i \in \mathcal{V}$ and $j \in \{1, \ldots, d\}$.

To proceed, let $h_i(x) = \sum_{k \in \mathcal{V},\, k \neq i} f_k(x)$ for all $i \in \mathcal{V}$. From Corollary 1, $\arg\min_x h_i(x) = \mathcal{X}^*$. Since $x^* \in \text{int}(\mathcal{X}^*)$, both $f(x)$ and $h_i(x)$ are differentiable at x^*, implying that $\frac{\partial f}{\partial x_j}(x^*) = \frac{\partial h_i}{\partial x_j}(x^*) = 0$ for all $i \in \mathcal{V}$ and $j \in \{1, \ldots, d\}$. Since $f_i(x) = f(x) - h_i(x)$, $\frac{\partial f_i}{\partial x_j}(x^*) = 0$ for all $i \in \mathcal{V}$ and $j \in \{1, \ldots, d\}$. Note that this holds for all $x^* \in \text{int}(\mathcal{X}^*)$. From [39, Section 8.4.2], $f_i(x)$ is differentiable at x^* with $\nabla f_i(x^*) = 0$ for all $i \in \mathcal{V}$. ■

Lemma 3 has the following important implication.

Corollary 3: Under Assumption 1, if the n-agent network is k-redundant with $k \geq 1$, then for all $i \in \mathcal{V}$,

$$\mathcal{X}^* \subset \arg\min_x f_i(x).$$

Corollary 3 immediately implies that

$$\bigcap_{i \in \mathcal{V}} \arg\min_x f_i(x) = \mathcal{X}^*.$$

Proof of Corollary 3: Suppose that, to the contrary, there exist $x^* \in \mathcal{X}^*$ and $i \in \mathcal{V}$ such that $x^* \notin \arg\min_x f_i(x)$. Pick a $z \in \text{int}(\mathcal{X}^*)$. From Lemma 3, $z \in \arg\min_x f_i(x)$. It is then clear that $f_i(x^*) > f_i(z)$. Let $h_i(x) = \sum_{k \in \mathcal{V},\, k \neq i} f_k(x)$. From Corollary 1, $\arg\min_x h_i(x) = \mathcal{X}^*$, and thus $h_i(x^*) = h_i(z)$. It follows that $f(x^*) = f_i(x^*) + h_i(x^*) > f_i(z) + h_i(z) = f(z)$, which contradicts the fact that $x^* \in \mathcal{X}^*$. ■

Assumption 2: The subgradients of all f_i, $i \in \mathcal{V}$, are uniformly bounded, i.e., there exists a positive number D such that $\|g_i(x)\| \leq D$ for all $i \in \mathcal{V}$ and $x \in \mathbb{R}^d$.

Assumption 3: The step-size sequence $\{\alpha(t)\}$ is positive, non-increasing, and satisfies $\sum_{t=0}^{\infty} \alpha(t) = \infty$ and $\sum_{t=0}^{\infty} \alpha^2(t) < \infty$.

The above two assumptions are standard for subgradient methods.

To state our main results, we need the following concepts. For a directed graph $\mathbb{G}$, we use $\mathcal{R}_{r,s}(\mathbb{G})$ to denote the set of all (r, s)-reduced graphs of $\mathbb{G}$. For a rooted graph $\mathbb{G}$, we use $\kappa(\mathbb{G})$ to denote the size of the unique strongly connected component whose vertices are all roots of $\mathbb{G}$; in other words, $\kappa(\mathbb{G})$ equals the number of roots of $\mathbb{G}$. For any (r, s)-resilient graph $\mathbb{G}$, let

$$\kappa_{r,s}(\mathbb{G}) \triangleq \min_{\mathbb{H} \in \mathcal{R}_{r,s}(\mathbb{G})} \kappa(\mathbb{H}).$$

which is well-defined and denotes the smallest possible number of roots in any (r, s)-reduced graphs of $\mathbb{G}$.

Theorem 1: Under Assumptions 1–3, if $\mathbb{G}$ is $(\beta, d\beta)$-resilient and the n-agent network is $(n - \kappa_{\beta,d\beta}(\mathbb{G}))$-redundant, then all $x_i(t)$, $i \in \mathcal{H}$ will asymptotically reach a consensus at a point in $\mathcal{X}^*$.

The following example shows that $(n - \kappa_{\beta,d\beta}(\mathbb{G}))$-redundancy is necessary. For simplicity, set $d = 1$. Consider a 4-agent network whose neighbor graph is the 4-vertex complete graph $\mathbb{C}$, which is $(1, 1)$-resilient. Suppose that agent 4 is Byzantine and the other three are normal. It is possible that, with a carefully crafted attack strategy of the Byzantine agent, the three normal agents update their states mathematically equivalent to the case as if their neighbor graph is the 3-vertex $(1, 1)$-reduced graph with the arc set $\{(1, 2), (1, 3), (2, 3)\}$, which is rooted (cf. Lemma 6 in [1]). In this case, since vertex 1 is the only root and agent 1 does not have any neighbor, it follows the single-agent subgradient algorithm, and thus its state will converge to a minimum point of $f_1(x)$, denoted x^*. Since all normal agents will eventually reach a consensus (cf. Lemma 9 in [1]), both states of agents 2 and 3 will converge to x^*. To guarantee the resilient distributed optimization problem is solvable in this case, there must hold that $x^* \in \arg\min_x f_i(x)$, $i \in \{1, 2, 3\}$, which implies that the network needs to be 3-redundant. It is easy to see that $\kappa_{1,1}(\mathbb{C}) = 1$, and thus $n - \kappa_{1,1}(\mathbb{G}) = 3$.

Theorem 1 shows that the proposed algorithm achieves full resiliency. We next partially characterize the convergence rate of the algorithm.

Theorem 2: Under Assumptions 1 and 2, if $\mathbb{G}$ is $(\beta, d\beta)$-resilient, the n-agent network is $(n-\kappa_{\beta,d\beta}(\mathbb{G}))$-redundant, and $\alpha(t)=1/\sqrt{T}$ for $T>0$ steps, i.e., $t\in\{0,1,\ldots,T-1\}$, then there exist a subset of normal agents $\mathcal{S}\subset\mathcal{H}$ with $|\mathcal{S}|\geq\kappa_{\beta,d\beta}(\mathbb{G})$, a positive constant $C\geq 1$, and a time subsequence $\mathcal{T}\subset\{0,1,\ldots,T-1\}$ with $|\mathcal{T}|\geq T/C$ such that for any $j\in\mathcal{H}$ and $x^*\in\mathcal{X}^*$,

$$\sum_{i\in\mathcal{S}} f_i\Big(\frac{\sum_{t\in\mathcal{T}} x_j(t)}{|\mathcal{T}|}\Big)-\sum_{i\in\mathcal{S}} f_i(x^*)\leq O\Big(\frac{1}{\sqrt{T}}\Big). \quad (8)$$

The existing distributed convex optimization literature (without Byzantine agents) typically characterizes convergence rates by bounding the difference between $\sum_{i\in\mathcal{V}} f_i(\frac{1}{T}\sum_{t=0}^{T-1} x_i(t))$ and $\sum_{i\in\mathcal{V}} f_i(x^*)$. The above theorem can be viewed as a "partial" convergence rate result in that it only reckons a subset $\mathcal{S}$ of normal agents and a subsequence $\mathcal{T}$ in a finite time horizon. Notwithstanding this, it is worth noting that $\min\sum_{i\in\mathcal{S}} f_i(x)$ is equivalent to $\min\sum_{i\in\mathcal{V}} f_i(x)$ in the setting here with Byzantine agents (cf. Corollary 1) and that $|\mathcal{T}|=O(T)$. Therefore, the theorem still to some extent evaluates the convergence rate of the resilient distributed subgradient algorithm under consideration. It is well known that the optimal convergence rate of subgradient methods for convex optimization is $O(1/\sqrt{t})$. Whether $f_{\mathcal{H}}(\cdot)=\sum_{i\in\mathcal{H}} f_i(\cdot)$ converges at this optimal rate or not, has so far eluded us.

Theorem 2 is an immediate consequence of the following proposition.

Proposition 1: Under Assumptions 1 and 2, if $\mathbb{G}$ is $(\beta, d\beta)$-resilient, the n-agent network is $(n-\kappa_{\beta,d\beta}(\mathbb{G}))$-redundant, and $\alpha(t)=1/\sqrt{T}$ for $t\in\{0,1,\ldots,T-1\}$, then for any integer $b\in[\kappa_{\beta,d\beta}(\mathbb{G}), n-|\mathcal{F}|]$, there exist a subset of normal agents $\mathcal{S}\subset\mathcal{H}$ with $b\geq|\mathcal{S}|\geq\kappa_{\beta,d\beta}(\mathbb{G})$ and a time subsequence $\mathcal{T}\subset\{0,1,\ldots,T-1\}$ with $|\mathcal{T}|\geq T/\sum_{k=\kappa_{\beta,d\beta}(\mathbb{G})}^{b}\binom{n-|\mathcal{F}|}{k}$ such that (8) holds for any $j\in\mathcal{H}$ and $x^*\in\mathcal{X}^*$.

The proposition further quantifies a trade-off between the number of normal agents in $\mathcal{S}$ and the length of time subsequence $\mathcal{T}$. Roughly speaking, the fewer the normal agents involved in (8), the denser would the time subsequence be. In the special case when $b=\kappa_{\beta,d\beta}(\mathbb{G})$, the proposition simplifies to the following corollary.

Corollary 4: Under Assumptions 1 and 2, if $\mathbb{G}$ is $(\beta, d\beta)$-resilient, the n-agent network is $(n-\kappa_{\beta,d\beta}(\mathbb{G}))$-redundant, and $\alpha(t)=1/\sqrt{T}$ for $t\in\{0,1,\ldots,T-1\}$, then there exist a subset of normal agents $\mathcal{S}\subset\mathcal{H}$ with $|\mathcal{S}|=\kappa_{\beta,d\beta}(\mathbb{G})$ and a time subsequence $\mathcal{T}\subset\{0,1,\ldots,T-1\}$ with $|\mathcal{T}|\geq T/\binom{n-|\mathcal{F}|}{\kappa_{\beta,d\beta}(\mathbb{G})}$ such that (8) holds for any $j\in\mathcal{H}$ and $x^*\in\mathcal{X}^*$.

IV. Concluding Remarks

This paper has proposed a distributed subgradient algorithm which achieves full resilience in the presence of Byzantine agents, with appropriate redundancy in both graph connectivity and objective functions. The algorithm and convergence results can be easily extended to time-varying neighbor graphs, provided that the neighbor graph is $(\beta, d\beta)$-resilient all the time. One immediate next step is to relax Assumption 1, possibly appealing to gradient descent for differentiable convex functions. The concepts and tools developed in the paper are expected to be applicable to other consensus-based distributed optimization and computation problems.

Although the algorithm theoretically works for multi-dimensional convex optimization, it has the following limitations which preclude its applicability to high-dimensional optimization. First, from Lemma 2, the algorithm implicitly requires that each agent have at least $(d+1)\beta+1$ neighbors, which is impossible for high dimensions. Second, picking a point in the intersection of multiple convex hulls (cf. step (4) in the algorithm) can be computationally expensive in high dimensions, although the issue has been attenuated in [29, Algorithm 2] and [32, Section 5.1]. Last, building $(\beta, d\beta)$-resilient graphs is not an easy job, especially when d or β is large. Another practical issue of the algorithm, independent of dimensions, is how to measure and establish objective function redundancy. Studies of (r,s)-resilient graphs and k-redundant multi-agent networks are of independent interest.

Considering that nowadays distributed optimization algorithms in machine learning are frequently high-dimensional, there is ample motivation to design fully resilient high-dimensional distributed optimization algorithms. A future direction of this paper aims to tackle this challenging problem by combining the proposed algorithm with communication-efficient schemes in which each agent can transmit only low-dimensional signals. Possible approaches include entry-wise or block-wise updating [40], [41], limited information fusion [42], and dimension-independent filtering [17].

Acknowledgement

The authors wish to thank Wenhan Gao (Stony Brook University) for useful discussion and thank all the anonymous reviewers for their helpful comments.

References

[1] J. Zhu, Y. Lin, A. Velasquez, and J. Liu. Resilient distributed optimization. 2023. arXiv:2209.13095 [math.OC].

[2] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[3] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K.H. Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2019.

[4] A. Nedić and J. Liu. Distributed optimization for control. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:77–103, 2018.

[5] D.K. Molzahn, F. Dörfler, H. Sandberg, S.H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei. A survey of distributed optimization and control algorithms for electric power systems. *IEEE Transactions on Smart Grid*, 8(6):2941–2962, 2017.

[6] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[7] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[8] L. Su and N.H. Vaidya. Fault-tolerant multi-agent optimization: Optimal iterative distributed algorithms. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, pages 425–434, 2016.

[9] K. Kuwarancharoen, L. Xin, and S. Sundaram. Byzantine-resilient distributed optimization of multi-dimensional functions. In *Proceedings of the 2020 American Control Conference*, pages 4399–4404, 2020.

[10] Z. Yang and W.U. Bajwa. ByRDiE: Byzantine-resilient distributed coordinate descent for decentralized learning. *IEEE Transactions on Signal and Information Processing over Networks*, 5(4):611–627, 2019.

[11] C. Fang, Z. Yang, and W.U. Bajwa. BRIDGE: Byzantine-resilient decentralized gradient descent. *IEEE Transactions on Signal and Information Processing over Networks*, 8:610–626, 2022.

[12] L. He, S.P. Karimireddy, and M. Jaggi. Byzantine-robust decentralized learning via self-centered clipping. 2022. arXiv:2202.01545 [cs.LG].

[13] Z. Wu, T. Chen, and Q. Ling. Byzantine-resilient decentralized stochastic optimization with robust aggregation rules. 2022. arXiv:2206.04568 [cs.DC].

[14] S. Sundaram and B. Gharesifard. Distributed optimization under adversarial nodes. *IEEE Transactions on Automatic Control*, 64(3):1063–1076, 2018.

[15] L. Su and N.H. Vaidya. Byzantine-resilient multiagent optimization. *IEEE Transactions on Automatic Control*, 66(5):2227–2233, 2020.

[16] C. Zhao, J. He, and Q.-G. Wang. Resilient distributed optimization algorithm against adversarial attacks. *IEEE Transactions on Automatic Control*, 65(10):4308–4315, 2020.

[17] N. Gupta, T.T. Doan, and N.H. Vaidya. Byzantine fault-tolerance in decentralized optimization under $2f$-redundancy. In *Proceedings of the 2021 American Control Conference*, pages 3632–3637, 2021.

[18] N. Gupta and N.H. Vaidya. Byzantine fault-tolerance in peer-to-peer distributed gradient-descent. 2021. arXiv:2101.12316 [cs.DC].

[19] M. Kaheni, E. Usai, and M. Franceschelli. Resilient constrained optimization in multi-agent systems with improved guarantee on approximation bounds. *IEEE Control Systems Letters*, 6:2659–2664, 2022.

[20] S. Liu, N. Gupta, and N.H. Vaidya. Approximate Byzantine fault-tolerance in distributed optimization. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, pages 379–389, 2021.

[21] N. Gupta, T.T. Doan, and N. Vaidya. Byzantine fault-tolerance in federated local SGD under $2f$-redundancy. 2021. arXiv:2108.11769 [cs.DC].

[22] D. Data, L. Song, and S.N. Diggavi. Data encoding for Byzantine-resilient distributed optimization. *IEEE Transactions on Information Theory*, 67(2):1117–1140, 2021.

[23] C.A. Uribe, H.-T. Wai, and M. Alizadeh. Resilient distributed optimization algorithms for resource allocation. In *Proceedings of the 58th IEEE Conference on Decision and Control*, pages 8341–8346, 2019.

[24] Z. Yang, A. Gang, and W.U. Bajwa. Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the Byzantine threat model. *IEEE Signal Processing Magazine*, 37(3):146–159, 2020.

[25] N. Gupta and N. H. Vaidya. Resilience in collaborative optimization: Redundant and independent cost functions. 2020. arXiv:2003.09675v2 [cs.DC].

[26] N.H. Vaidya. Iterative Byzantine vector consensus in incomplete graphs. In *Proceedings of the 15th International Conference on Distributed Computing and Networking*, pages 14–28, 2014.

[27] H. Tverberg. A generalization of Radon's theorem. *Journal of the London Mathematical Society*, 41:123–128, 1966.

[28] P.K. Agarwal, M. Sharir, and E. Welzl. Algorithms for center and Tverberg points. In *Proceedings of the 20th Annual Symposium on Computational Geometry*, pages 61–67, 2004.

[29] X. Wang, S. Mou, and S. Sundaram. Resilience for distributed consensus with constraints. 2022. arXiv:2206.05662 [eess.SY].

[30] J. Liu, T. Başar, and A. Nedić. Input-output stability of linear consensus processes. In *Proceedings of the 55th IEEE Conference on Decision and Control*, pages 6978–6983, 2016.

[31] W. Abbas, M. Shabbir, J. Li, and X. Koutsoukos. Resilient distributed vector consensus using centerpoint. *Automatica*, 136:110046, 2022.

[32] J. Yan, X. Li, Y. Mo, and C. Wen. Resilient multi-dimensional consensus in adversarial environment. *Automatica*, 145:110530, 2022.

[33] B. Polyak. A general method for solving extremum problems. *Doklady Akademii Nauk*, 8(3):593–597, 1967.

[34] A. Nedić and A. Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2015.

[35] Y. Lin and J. Liu. Subgradient-push is of the optimal convergence rate. In *Proceedings of the 61st IEEE Conference on Decision and Control*, pages 5849–5856, 2022.

[36] N.H. Vaidya, L. Tseng, and G. Liang. Iterative approximate Byzantine consensus in arbitrary directed graphs. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, pages 365–374, 2012.

[37] H.J. Leblance, H. Zhang, X. Koutsoukos, and S. Sundaram. Resilient asymptotic consensus in robust networks. *IEEE Journal on Selected Areas in Communications*, 31(4):766–781, 2013.

[38] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 2015.

[39] V.A. Zorich and R. Cooke. *Mathematical Analysis I*. Mathematical Analysis. Springer, 2004.

[40] J. Liu and B.D.O. Anderson. Communication-efficient distributed algorithms for solving linear algebraic equations over directed graphs. In *Proceedings of the 59th IEEE Conference on Decision and Control*, pages 5360–5365, 2020.

[41] I. Notarnicola, Y. Sun, G. Scutari, and G. Notarstefano. Distributed big-data optimization via blockwise gradient tracking. *IEEE Transactions on Automatic Control*, 66(5):2045–2060, 2021.

[42] J. Zhu, Y. Lin, J. Liu, and A.S. Morse. Reaching a consensus with limited information. In *Proceedings of the 61st IEEE Conference on Decision and Control*, pages 4579–4584, 2022.