Distributed Optimization Under Adversarial Nodes

Shreyas Sundaram  Bahman Gharesifard

Abstract—We investigate the vulnerabilities of consensus-based distributed optimization protocols to nodes that deviate from the prescribed update rule (e.g., due to failures or adversarial attacks). We first characterize certain fundamental limitations on the performance of any distributed optimization algorithm in the presence of adversaries. We then propose a resilient distributed optimization algorithm that guarantees that the non-adversarial nodes converge to the convex hull of the minimizers of their local functions under certain conditions on the graph topology, regardless of the actions of a certain number of adversarial nodes. In particular, we provide sufficient conditions on the graph topology to tolerate a bounded number of adversaries in the neighborhood of every non-adversarial node, and necessary and sufficient conditions to tolerate a globally bounded number of adversaries. For situations where there are up to \( F \) adversaries in the neighborhood of every node, we use the concept of maximal \( F \)-local sets of graphs to provide lower bounds on the distance-to-optimality of achievable solutions under any algorithm. We show that finding the size of such sets is NP-hard.

I. INTRODUCTION

In recent years, the topic of distributed optimization has become a canonical problem in the study of networked systems. In this setting, a group of agents equipped with individual objective functions are required to agree on a state that optimizes the sum of these functions. As in the classical consensus problem, the agents can only operate on local information obtained from their neighboring agents, described by a communication network. There is a vast literature devoted to designing distributed algorithms, both in discrete and continuous-time, that guarantee convergence to an optimizer of the sum of the objective functions under reasonable convexity and continuity assumptions \([1]–[12]\).

As outlined above, the predominant assumption in distributed optimization is that all agents cooperate to calculate the global optimizer. In particular, in typical distributed optimization protocols, the individuals update their state via a combination of an agreement term and an appropriately scaled gradient flow of their individual functions. Given the potential applications of distributed optimization algorithms in large-scale (and safety-critical) cyber-physical systems, and motivated by studies of resilience issues in consensus dynamics (e.g., see \([13]–[18]\)), it is reasonable to ask how vulnerable consensus-based distributed optimization algorithms are with respect to failure or malicious behavior by certain nodes.

In fact, as we argue in this paper, current consensus-based distributed optimization algorithms are easily disrupted by adversarial behavior. The main objective of this paper is hence to address the issue of resilience of consensus-based distributed optimization dynamics to failure and adversarial behavior, and to refine existing distributed optimization protocols to provide certain safety guarantees against different numbers and types of attackers. The recent work \([19]\) also considers the problem of distributed optimization with adversaries under different assumptions on the graph topology, faulty behavior and classes of functions than the ones that we consider here. The material in this paper substantially extends the conference papers \([20], [21]\) by providing complete proofs of the results, along with characterizations of the factors that affect the performance of distributed optimization algorithms under adversarial behavior. The contributions of this paper can be summarized as follows.

Statement of Contributions

The first contribution of this paper is to demonstrate fundamental limitations on the performance of any distributed optimization algorithm in the presence of adversaries. In particular, we show that it is impossible to develop an algorithm that always finds optimal solutions in the absence of adversaries and is at the same time resilient to carefully crafted attacks.

As our second contribution, we introduce a resilient version of the consensus-based distributed optimization protocol, which we term Local Filtering (LF) Dynamics, in which the nodes discard the most extreme values in their neighborhood at each time-step. We investigate the capabilities of such protocols under different classes of adversarial behavior, and under the assumption of having an upper bound \( F \) on either the total number of adversarial nodes in the network (termed the \( F \)-total model) or on the local number of adversarial nodes in the neighborhood of each non-adversarial node (termed the \( F \)-local model). In particular, we provide graph-theoretic sufficient conditions for consensus in scenarios with \( F \)-local Byzantine adversaries (which can send different values to different neighbors at each time-step), and necessary and sufficient conditions for scenarios with \( F \)-total malicious adversaries (which operate under the wireless broadcast model of communication). We utilize two different proof techniques for the two scenarios (each of which provides different insights and capabilities): the first proof relies on properties of products of stochastic matrices for rooted graphs, and relates the consensus value to the limiting left-eigenvector of the subgraph of regular nodes corresponding to eigenvalue 1. The second proof relies on characterizing the contracting behavior of the gap between the regular agents with extreme values, and applies even when the graphs are not rooted at each time-step (which
can occur under our dynamics, as we demonstrate).

Our third contribution is to provide a safety guarantee for the proposed LF-dynamics. When the sequence of gradient step-sizes decreases to zero and has finite 1-norm (a typical condition in gradient-based optimization dynamics [9]), we prove that the states of the non-adversarial nodes converge to the convex hull of the minimizers of the individual functions, regardless of the actions taken by the adversarial nodes.

As our last contribution, we characterize factors that affect the performance of resilient distributed optimization algorithms. We provide a bound which shows that for graphs with large so-called maximum $F$-local sets, the performance of resilient algorithms can be poor under the $F$-local adversary model. As a by-product, we prove that the complexity of finding the size of the maximum $F$-local set is NP-hard. Several examples demonstrate our results.

Organization

Section II introduces various mathematical preliminaries. In Section III we review the standard consensus-based distributed optimization algorithm. We describe the adversary model in Section IV illustrate vulnerabilities in existing algorithms, and provide fundamental limitations on any distributed optimization algorithms under such adversarial behavior. We then introduce a class of resilient distributed optimization algorithms in Section VII we provide our main results on consensus under this algorithm in Section VI and provide safety guarantees on this algorithm in Section VII. We identify factors that affect the performance of resilient distributed optimization algorithms in Section VIII and conclude in Section IX.

II. Mathematical Notation and Terminology

Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{N}$ denote the real, nonnegative real, and natural numbers, respectively, $\| \cdot \|$ the Euclidean norm on $\mathbb{R}^n$, $1 = [1 \ 1 \ \cdots \ 1]^T$, $0 = [0 \ 0 \ \cdots \ 0]^T$, and $I_n$ the identity matrix in $\mathbb{R}^{n \times n}$ with nonnegative entries is called (row) stochastic if $A1 = 1$. Throughout this paper, we are concerned with stochastic matrices whose diagonal entries are bounded away from zero. For a locally Lipschitz function $f : \mathbb{R} \rightarrow \mathbb{R}$, we denote the set of subgradients at a given point $x \in \mathbb{R}$ by $\partial f(x)$.

A graph $\mathcal{G} = (V, \mathcal{E})$ consists of a set of vertices (or nodes) $V = \{v_1, v_2, \ldots, v_n\}$, and a set of edges $\mathcal{E} \subset V \times V$. The graph is said to be undirected if $(v_i, v_j) \in \mathcal{E} \Leftrightarrow (v_j, v_i) \in \mathcal{E}$, and directed otherwise. The in-neighbors and out-neighbors of vertex $v_i \in V$ are denoted by the sets $N_i^- \equiv \{v_j \in V \mid (v_j, v_i) \in \mathcal{E}\}$ and $N_i^+ \equiv \{v_j \in V \mid (v_i, v_j) \in \mathcal{E}\}$, respectively. The in-degree and out-degree of vertex $v_i \in V$ are denoted by $d_i^- \equiv \|N_i^-\|$ and $d_i^+ \equiv \|N_i^+\|$, respectively. For undirected graphs, we denote $N_i = N_i^- = N_i^+$ as the neighbors of vertex $v_i \in V$, and $\delta_i = d_i^- = d_i^+$ as the degree. We denote time-varying graphs, edge sets, and neighbor sets by appending a time-index to those quantities.

A path from vertex $v_i \in V$ to vertex $v_j \in V$ is a sequence of vertices $v_{k_1}, v_{k_2}, \ldots, v_{k_l}$ such that $v_{k_1} = v_i$, $v_{k_l} = v_j$ and $(v_{k_r}, v_{k_{r+1}}) \in \mathcal{E}$ for $1 \leq r \leq l - 1$. A graph $\mathcal{G} = (V, \mathcal{E})$ is said to be rooted at vertex $v_i \in V$ if for all vertices $v_j \in V \setminus \{v_i\}$, there a path from $v_i$ to $v_j$. A graph is said to be rooted if it is rooted at some vertex $v_i \in V$. A graph is strongly connected if there is a path from every vertex to every other vertex in the graph.

For any $r \in \mathbb{N}$, a subset $S \subset V$ of vertices is said to be $r$-local if $|N_i^- \cap S| \leq r$ for all $v_i \in V \setminus S$. In other words, if $S$ is $r$-local, then there are at most $r$ vertices from $S$ in the in-neighborhood of any vertex from $V \setminus S$. A maximum $r$-local set is an $r$-local set of largest cardinality (i.e., there are no $r$-local sets of larger size). A subset $S \subset V$ of vertices is said to be $r$-reachable if there exists a vertex $v_i \in S$ such that $|N_i^- \setminus S| \geq r$. In other words, $S$ is $r$-reachable if it contains a vertex that has at least $r$ in-neighbors from outside $S$.

The following definitions of robust graphs will play a role in our analysis.

Definition 2.1 ($r$-robust graphs): For $r \in \mathbb{N}$, graph $\mathcal{G}$ is said to be $r$-robust if for all pairs of disjoint nonempty subsets $S_1, S_2 \subset V$, at least one of $S_1$ or $S_2$ is $r$-reachable.

Definition 2.2 ($(r, s)$-robust graphs): For $r, s \in \mathbb{N}$, a graph is said to be $(r, s)$-robust if for all pairs of disjoint nonempty subsets $S_1, S_2 \subset V$, at least one of the following conditions holds:

(i) All nodes in $S_1$ have at least $r$ neighbors outside $S_1$.
(ii) All nodes in $S_2$ have at least $r$ neighbors outside $S_2$.
(iii) There are at least $s$ nodes in $S_1 \cup S_2$ that each have at least $r$ neighbors outside their respective sets.

The above definitions capture the idea that given any two disjoint nonempty subsets of nodes in the network, there are a certain number of nodes within those sets that each have a sufficient number of neighbors outside their respective sets. This notion will play a key role in the resilient dynamics that we propose in this paper, where nodes choose to discard a certain number of their neighbors in order to mitigate adversarial behavior. Note that $(r, 1)$-robustness is equivalent to $r$-robustness. The following result (from Lemma 6 and Lemma 7 in [13]) will be useful for our analysis.

Lemma 2.3: Suppose a graph $\mathcal{G}$ is $r$-robust. Let $\mathcal{G}'$ be a graph obtained by removing $r - 1$ or fewer incoming edges from each node in $\mathcal{G}$. Then $\mathcal{G}'$ is rooted.

Further details on the above notions of robustness can be found in [13], [22].

III. Review of Consensus-Based Distributed Optimization

Consider a network consisting of $n$ agents $V = \{v_1, \ldots, v_n\}$ whose communication topology is a potentially time-varying graph $\mathcal{G}(t) = (V, \mathcal{E}(t))$. An edge $(v_i, v_j) \in \mathcal{E}(t)$
indicates that \(v_j\) can receive information from \(v_i\) at time-step \(t \in \mathbb{N}\). For each \(i \in \{1, \ldots, n\}\), let \(f_i : \mathbb{R} \to \mathbb{R}\) be locally Lipschitz and convex, and only available to agent \(v_i\). The objective is for the agents to solve, in a distributed way (i.e., by exchanging information only with their immediate neighbors), the global optimization problem:

\[
\text{minimize } f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x).
\]

A common approach to solve this problem is to use a synchronous iterative consensus-based protocol in which agents use a combination of consensus dynamics and gradient flow to find a minimizer of \(f\) \cite{1, 6, 23}. Specifically, at each time-step \(t \in \mathbb{N}\), each agent \(v_i \in V\) has an estimate \(x_i(t) \in \mathbb{R}\) of the solution to the problem \(1\). Each agent \(v_i \in V\) sends its estimate to its out-neighbors, receives the estimates of its in-neighbors, and updates its estimate as:

\[
x_i(t+1) = a_{ii}(t)x_i(t) + \sum_{v_j \in N_i^{-}(t)} a_{ij}(t)x_j(t) - \alpha_t d_i(t). \quad (2)
\]

In the above update rule, \(a_{ij}(t), v_j \in \{v_i\} \cup N_i^{-}(t)\), are a set of nonnegative real numbers satisfying \(a_{ii}(t) + \sum_{v_j \in N_i^{-}(t)} a_{ij}(t) = 1\). In other words, the first portion of the right hand side is a consensus step, representing a weighted average of the estimates at node \(v_i\)’s neighborhood. The quantity \(d_i(t)\) is a subgradient of \(f_i\), evaluated at \(a_{ii}(t)x_i(t) + \sum_{v_j \in N_i^{-}(t)} a_{ij}(t)x_j(t)\). Finally, \((\alpha_t)_{t \in \mathbb{N}}\) is the step-size sequence corresponding to the influence of the subgradient on the update rule at each time-step. In this sense, the last term in the above expression represents a gradient step.

The dynamics \(2\) can be represented compactly as follows. Let

\[
x(t) \triangleq [x_1(t) x_2(t) \ldots x_n(t)]' \in \mathbb{R}^n,
\]

\[
d(t) \triangleq [d_1(t) d_2(t) \ldots d_n(t)]' \in \mathbb{R}^n
\]

be the vector of states and subgradients of the nodes at time-step \(t\), respectively. Let \(A(t) \in \mathbb{R}^{n \times n}\) be the matrix such that for each \((v_j, v_i) \in E(t)\), the \((i, j)\)-th entry of \(A(t)\) is \(a_{ij}(t)\) given in \(2\), the diagonal elements of \(A(t)\) are the self-weights \(a_{ii}(t)\), and all other entries are set to zero. Then \(2\) can be written as

\[
x(t+1) = A(t)x(t) - \alpha_t d(t),
\]

for \(t \in \mathbb{N}\). Note that each row of \(A(t)\) sums to 1 at each time-step, and thus \(A(t)\) is row-stochastic. It is easy to observe that

\[
x(t+1) = A(t)A(t-1)\cdots A(0)x(0)
\]

\[
- \sum_{s=1}^{t} A(t)A(t-1)\cdots A(s)\alpha_{s-1}d(s-1) - \alpha_t d(t).
\]

For notational convenience, we define \(\Phi(t, s) \triangleq A(t)A(t-1)\cdots A(s)\) for \(t \geq s\), and \(\Phi(t, s) \triangleq 0\) for \(t < s\). Thus, \(4\) becomes

\[
x(t + 1) = \Phi(t, 0)x(0) - \sum_{s=1}^{t} \Phi(t, s)\alpha_{s-1}d(s-1) - \alpha_t d(t).
\]

There are some commonly-used assumptions that are made on the weights in \(2\), which we encapsulate below.

**Assumption 3.1 (Lower Bounded Weights):** There exists a constant \(\eta > 0\) such that for all \(t \in \mathbb{N}\) and \(v_i \in V\), if \(v_j \notin \{v_i\} \cup N_i^{-}(t)\), then \(a_{ij}(t) \geq \eta\).

**Assumption 3.2 (Double Stochasticity):** For all \(t \in \mathbb{N}\) and \(v_i \in V\), the weights satisfy \(a_{ii}(t) + \sum_{v_j \in N_i^{-}(t)} a_{ij}(t) = 1\).

The following result is a special case of the results of \(6\) for graphs that are strongly connected at each time-step.

**Proposition 3.3:** Suppose the network \(G(t)\) is strongly connected at each time-step. Suppose the subgradients of each of the local functions \(f_i\) are bounded, i.e., there exists \(L \in \mathbb{R}_{>0}\) such that \(\|d\| \leq L\), for all \(d \in \partial f_i(x)\) and \(x \in \mathbb{R}\). Consider the update rule \(2\), and suppose the weights satisfy Assumption \(3.1\) and Assumption \(3.2\). Let the step-sizes satisfy \(\sum_{t \in \mathbb{N}} \alpha_t = \infty\) and \(\sum_{t \in \mathbb{N}} \alpha_t^2 < \infty\). Then there is a minimizer \(x^*\) of \(1\) such that

\[
\lim_{t \to \infty} \|x(t) - x^*\| = 0,
\]

for all \(v_i \in V\).

The above result shows that the update rule \(2\) allows the nodes in the network to distributively solve the global optimization problem \(1\). Our main objective in this paper is to investigate the vulnerabilities of such protocols to nodes that deviate from the prescribed update rule (e.g., due to failures or adversarial attacks), and to develop a resilient distributed optimization algorithm that has provable safety guarantees in the presence of such deviations. To do this, it will be helpful to first generalize the above analysis to handle cases where the weights are not doubly-stochastic.

**A. Scenarios with Non-Doubly-Stochastic Weights**

Here, we will establish convergence of the node states under the dynamics \(2\) under certain classes of non-doubly-stochastic consensus weights. At each time-step \(t \in \mathbb{N}\), let \(A(t) \in \mathbb{R}^{n \times n}\) be the matrix containing the weights \(a_{ij}(t)\). Note that \(a_{ij}(t) = 0\) if \(v_j \notin \{v_i\} \cup N_i^{-}(t)\). Suppose there exists some constant \(\beta > 0\) such that at each time-step \(t \in \mathbb{N}\), \(A(t)\) has a rooted subgraph that has edge-weights lower-bounded by \(\beta\), and diagonal elements lower-bounded by \(\beta\). Let \(\Phi(t, s) \triangleq A(t)A(t-1)\cdots A(s)\) for \(t \geq s\geq 0\). Using the fact that \(A(t)\) has a rooted subgraph, and with an argument similar to the one in \(24\) which we omit here, for each \(s \in \mathbb{N}\), there exists a stochastic vector \(q_s\), such that

\[
\lim_{t \to \infty} \Phi(t, s) = q_s q_s',
\]
Noting that $\Phi(t, s) = \Phi(t, s + 1)A(s)$, we have that
\[ q_s^i = q_{s+1}^i A(s), \tag{6} \]
for all $s \in \mathbb{N}$.

For each $t \in \mathbb{N}$, let $x(t) \in \mathbb{R}^n$ be the state vector for the network, and define the quantity
\[ y(t) \triangleq q_t^i x(t) \tag{7} \]
(i.e., $y(t)$ is a convex combination of the states of the nodes at time-step $t$). Using the above definition, we have the following convergence result. The proof of this result closely follows the proof for doubly-stochastic weights provided in [6], with the main difference being in the use of the vector $q_t^i$ at appropriate points.

**Lemma 3.4**: Consider the network $\mathcal{G}(t) = (V, \mathcal{E}(t))$. Suppose that the functions $f_s$, $v_i \in V$, have subgradients bounded by some constant $L$, and that the nodes run the dynamics (2). Assume that there exists a constant $\beta > 0$ such that at each time-step $t \in \mathbb{N}$, the weight matrix $A(t)$ has diagonal elements lower bounded by $\beta$ and contains a root subgraph whose edge weights are lower bounded by $\beta$. Let $y(t)$ be the corresponding sequence defined in (7).

(i) If $\alpha_t \to 0$ as $t \to \infty$, then
\[ \lim_{t \to \infty} \sup_{t \in \mathbb{N}} \| x(t) - 1 y(t) \| = 0. \]

(ii) If $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$, then
\[ \sum_{t=1}^{\infty} \alpha_t \| x(t) - 1 y(t) \| < \infty. \]

(iii) If each matrix $A(t)$, $t \in \mathbb{N}$ has a common left-eigenvector $q^i$ corresponding to eigenvalue 1, and the step-sizes satisfy $\sum \alpha_t = \infty$ and $\sum \alpha_t^2 < \infty$, then
\[ \lim_{t \to \infty} \| x_i(t) - x^* \| = 0 \]
for all $v_i \in V$, where $x^*$ is a minimizer of $\sum_{i=1}^{n} q_i f_i$, with $q_i$ being the $i$-th entry of $q^i$.

Note that if the matrices $A(t)$ do not have a common left-eigenvector, convergence to a constant value is not guaranteed under the dynamics (2) (unlike in standard consensus dynamics without the gradient terms). To see this, consider two row-stochastic matrices $A_1$ and $A_2$, each with rooted subgraphs and nonzero diagonal elements, with different left eigenvectors $q_1^i$ and $q_2^i$, respectively, for eigenvalue 1. Select the functions for the nodes such that $\sum q_{1i} f_i$ and $\sum q_{2i} f_i$ have different minimizers, where $q_{ij}$ is the $j$-th component of $q_i$. Then, if the dynamics evolve according to matrix $A_1$ for a sufficiently large period of time, all nodes will approach the minimizer of $\sum q_{1i} f_i$, regardless of the initial conditions. Similarly, if the dynamics evolve according to the matrix $A_2$ for a sufficiently large period of time, all nodes will approach the minimizer of $\sum q_{2i} f_i$, again regardless of the initial conditions. Thus, by appropriately switching between the matrices $A_1$ and $A_2$, the nodes will oscillate between the two different minimizers.

With these results on distributed optimization in hand, we now turn our attention to the effect of adversaries on the optimization dynamics.

**IV. Adversary Model and Vulnerabilities of Distributed Optimization Algorithms**

Henceforth, we will assume that the underlying graph $\mathcal{G}$ is time-invariant in order to focus on issues pertaining to resilience to adversarial behavior. However, as we will see later, our proposed algorithm will utilize time-varying (and state-dependent) weights which can be viewed as inducing time-varying subgraphs of the underlying graph $\mathcal{G}$.

**A. Adversary Model**

We partition the set of nodes $V$ into two subsets: a set of adversarial nodes $A$, and a set of regular nodes $R = V \setminus A$. The system undergoes the following sequence of steps:

(i) Each node $v_i \in V$ draws a private function $f_i$ that is locally Lipschitz and convex.

(ii) A set of nodes $A \subset V$ is selected by an attacker to be adversarial. The attacker is allowed to know the entire network topology and the private functions assigned to all of the nodes when selecting the set $A$.

(iii) The regular nodes commence running the distributed optimization algorithm.

The regular nodes will exactly follow any algorithm that is prescribed. The adversarial nodes, on the other hand, can update their states in a completely arbitrary (potentially worst-case and coordinated) manner. We will classify adversaries in terms of their number, locations, and types of misbehavior, as follows.

**Definition 4.1 (F-total vs. F-local)**: For $F \in \mathbb{N}$, we say that the set of adversaries $A$ is an $F$-total set if $|A| \leq F$, and an $F$-local set if $|N_i^- \cap A| \leq F$, for all $v_i \in R$.

**Definition 4.2 (Malicious vs. Byzantine)**: We say that an adversarial node is **malicious** if it sends the same value to all of its out-neighbors at each time-step (i.e., it follows the wireless broadcast model of communication). We say that an adversarial node is **Byzantine** if it is capable of sending different values to different neighbors at each time-step (i.e., it follows the wired point-to-point model of communication).

Note that malicious adversaries are a special case of Byzantine adversaries, and similarly, $F$-total adversaries are a special case of $F$-local adversaries.

**B. Attacking Consensus-Based Distributed Optimization Algorithms**

We start with the following result showing that it is extremely simple for even a single adversarial node (either malicious or Byzantine) to disrupt dynamics of the form (2).
Proposition 4.3: Consider the network $G = (V,E)$, and let there be a single adversarial node $A = \{v_n\}$. Suppose the network is rooted at $v_n$. Then if $v_n$ keeps its value fixed at some constant $\bar{x} \in \mathbb{R}$ and the step-sizes satisfy $\alpha_t \to 0$, all regular nodes will asymptotically converge to $\bar{x}$ when following the distributed optimization dynamics (2).

Proof: Since the adversarial node keeps its value fixed for all time, its update can be modeled as

$$x_n(t + 1) = x_n(t)$$

for all $t \in \mathbb{N}$, with $x_n(0) = \bar{x}$. Thus, the global distributed optimization dynamics take the form shown in (3), with

$$A(t) = \begin{bmatrix} 1 \quad 0 \\ \bar{A}_{v_n} \end{bmatrix},$$

where $A_{v_n}(t)$ is the matrix containing the weights placed by regular nodes on other regular nodes during the update (2), and $\bar{A}_{v_n}(t)$ is a vector containing the weights placed by regular nodes on the adversarial node’s value. Since (i) the graph contains a spanning tree rooted at $v_n$, (ii) all weights used by the regular nodes on their neighbors (and own values) are bounded away from zero, and (iii) all matrices $A(t)$ have a common left-eigenvector $q^* = \begin{bmatrix} 0_{1 \times n-1} & 1 \end{bmatrix}$, the first part of Lemma 3.4 indicates that all regular nodes will converge to $y(t) = q^* x(t) = x_n(t) = \bar{x}$.

The above phenomenon is entirely analogous to the behavior that occurs under “stubborn” agents in standard consensus dynamics (e.g., [25], [26]).

C. Fundamental Limitations on Any Resilient Distributed Optimization Algorithm

The previous result shows that consensus-based distributed optimization algorithms can be co-opted by an adversary simply fixing its value at some constant. It is plausible that this type of simple misbehavior can be detected via an appropriate mechanism. However, it is easy to argue as follows that under mild conditions on the class of objective functions at each node, an adversary can always behave in any way as to avoid detection, while arbitrarily affecting the outcome of the distributed optimization.

Theorem 4.4: Suppose the local objective functions at each node are convex with bounded subgradients, but otherwise completely arbitrary. Suppose $\Gamma$ is a distributed algorithm that guarantees that all nodes calculate the global optimizer of problem (1) when there are no adversarial nodes. Then a single adversary can cause all nodes to converge to any arbitrary value when they run algorithm $\Gamma$, and furthermore, will remain undetected.

Proof: Without loss of generality, let $v_n$ be an adversarial node. Let each node $v_i \in V$ have local function $f_i$. Suppose node $v_n$ wishes all nodes to calculate some value $\bar{x}$ as an outcome of running the algorithm $\Gamma$. Node $v_n$ chooses a convex function $\bar{f}_n$ such that $\bar{f}_n(\bar{x}) = -\sum_{v_i \in V \setminus \{v_n\}} \partial f_i(\bar{x})$, with gradient capped at a sufficiently large value. Thus, the global minimizer of the function $\frac{1}{n} \left( \sum_{v_i \in V \setminus \{v_n\}} f_i + \bar{f}_n \right)$ is $\bar{x}$. Now, node $v_n$ participates in algorithm $\Gamma$ by pretending its local function is $\bar{f}_n$ instead of $f_n$. Since $\bar{f}_n$ is a legitimate function that could have been assigned to $v_n$, this scenario is indistinguishable from the case where $v_n$ is a regular node, and thus this misbehavior cannot be detected. Thus, algorithm $\Gamma$ must cause all nodes to calculate $\bar{x}$ under this misbehavior.

The above theorem applies to any algorithm that is guaranteed to output the globally optimum value in the absence of adversaries. The takeaway point is that there is a tradeoff between optimality and resilience: any algorithm that always finds optimal solutions in the absence of adversaries (under mild assumptions on the class of local functions) can also be arbitrarily co-opted by an adversary.

In the next section, we build on the insights gained from the above characterizations of fundamental limitations, and propose a modification of the standard consensus-based distributed optimization algorithm that provides certain safety guarantees in the face of arbitrary adversarial behavior.

V. A Resilient Consensus-Based Distributed Optimization Protocol

Suppose that the adversarial nodes are restricted to form an $F$-local set, where $F$ is a nonnegative integer. The regular nodes do not know which (if any) of their neighbors are adversarial. Suppose that at each time-step $t \in \mathbb{N}$, each regular node $v_i \in \mathcal{R}$ performs the following actions in parallel with the other regular nodes:

(i) Node $v_i$ gathers the states $\{x_j(t), v_j \in \mathcal{N}_i^{-}\}$ of its in-neighbors.

(ii) Node $v_i$ sorts the gathered values and removes the $F$ highest and $F$ smallest values that are larger and smaller than its own value, respectively. If there are fewer than $F$ values higher (resp. lower) than its own value, $v_i$ removes all of those values. Ties in values are broken arbitrarily. Let $\mathcal{J}_i(t) \subset \mathcal{N}_i^{-}$ be the set of in-neighbors of $v_i$ whose states were retained by $v_i$ at time-step $t$.

(iii) Node $v_i$ updates its state as

$$x_i(t + 1) = a_{ii}(t)x_i(t) + \sum_{v_j \in \mathcal{J}_i(t)} a_{ij}(t)x_j(t) - \alpha_td_i(t),$$

where $d_i(t)$ is a subgradient of $f_i$ evaluated at $a_{ii}(t)x_i(t) + \sum_{v_j \in \mathcal{J}_i(t)} a_{ij}(t)x_j(t)$, and $\{\alpha_t\}_{t \in \mathbb{N}}$ is a nonnegative step-size sequence. At each time-step $t$ and for each $v_i \in \mathcal{R}$, the weights $a_{ij}(t), v_j \in \{v_i\} \cup \mathcal{J}_i(t)$, are lower-bounded by some strictly positive real number $\eta$ and sum to 1 (i.e., they specify a convex combination).

The adversarial nodes are allowed to update their states however they wish. Note that the above dynamics are purely-local in the sense that they do not require the regular nodes to know anything about the network topology (other than their own in-neighbors). Also note that even when the underlying network $G$ is time-invariant, the filtering operation induces
state-dependent switching (i.e., the effective in-neighbor set
\( J_i(t) \) is a function of the states of the in-neighbors of \( v_i \) at
time-step \( t \)). In case a regular node \( v_i \) has a Byzantine neighbor
\( v_j \), we abuse notation and take the value \( x_j(t) \) in the update
equation \( 8 \) to be the value received from node \( v_j \) (i.e., it
does not have to represent the true state of node \( v_j \)).

We will refer to the above dynamics as Local Filtering (LF) Dynamics with parameter \( F \). Local filtering operations of the above form have been previously studied in the context of resilient consensus dynamics (i.e., outside of distributed optimization) in \[18, 27\]. However, the presence of the gradient terms in the dynamics \( 8 \) adds additional complexity that precludes the proof techniques from \[18\] from being directly applied, and thus we will analyze these dynamics in
the remainder of the paper, and show that they are resilient to
adversarial behavior under certain conditions on the network
topology.

A. Mathematically Equivalent Representation of Local Filtering Dynamics

Since we are concerned with understanding the evolution of the states of the regular nodes in our analysis, it will be useful to consider a mathematically equivalent representation of the dynamics \( 8 \) that only involves the states of the regular nodes. The key idea of the proof of the following proposition is from \[28\], which considered a slightly different version of the local filtering dynamics in the context of distributed consensus. Here, we provide a somewhat simpler proof, adapted for the version of the dynamics that we are considering.

**Proposition 5.1:** Consider the network \( G = (V, E) \), with
a set of regular nodes \( R \) and a set of adversarial nodes \( A \). Suppose that \( A \) is an \( F \)-local set, and that each regular node has at least \( 2F + 1 \) in-neighbors. Then the update rule \( 8 \) for
each node \( v_i \in R \) is mathematically equivalent to

\[ x_i(t+1) = \bar{a}_{ii}(t)x_i(t) + \sum_{v_j \in N^-_i \cap R} \bar{a}_{ij}(t)x_j(t) - \alpha_i d_i(t), \quad (9) \]

where the nonnegative weights \( \bar{a}_{ij}(t) \) satisfy the following properties at each time-step \( t \):

(i) \( \bar{a}_{ii}(t) + \sum_{v_j \in N^-_i \cap R} \bar{a}_{ij}(t) = 1 \).

(ii) \( \bar{a}_{ii}(t) \geq \eta \) and at least \( |N^-_i| - 2F \) of the other weights
are lower bounded by \( \frac{\eta}{2} \).

**Proof:** Consider a regular node \( v_i \in R \). We will prove the result by providing a procedure to construct the weights \( \bar{a}_{ij}(t) \) described in the statement, starting from the weights \( a_{ij}(t) \) in the LF dynamics \( 8 \). To facilitate this, we define two different partitions of the in-neighbors of \( v_i \). For the first partition, define the sets \( U_i(t), J_i(t) \) and \( L_i(t) \), where \( U_i(t) \) (resp. \( L_i(t) \)) contains the nodes with the highest (resp. lowest) values that were removed by node \( v_i \) after the filtering operation. For the second partition, define the sets \( U_i(t), \bar{J}_i(t) \) and \( \bar{L}_i(t) \), where \( U_i(t) \) and \( \bar{L}_i(t) \) contain the highest and lowest \( F \)
values in node \( v_i \)'s neighborhood at time-step \( t \), respectively.

The set \( J_i(t) \) contains the remaining values. Thus, we have
\( U_i(t) \subseteq U_i(t), J_i(t) \subseteq J_i(t), \) and \( L_i(t) \subseteq L_i(t) \).

Define \( \bar{a}_{ii}(t) = a_{ii}(t) \) and \( \bar{a}_{ij}(t) = a_{ij}(t) \) for
\( v_j \in J_i(t) \cap R \). Set \( \bar{a}_{i\bar{j}}(t) = 0 \) for \( v_j \in R \setminus J_i(t) \).

If there are no adversarial nodes in \( J_i(t) \) (i.e., \( J_i(t) = J_i(t) \cap R \)), then the construction of the weights \( \bar{a}_{ij}(t) \) for
node \( v_i \) is complete. Specifically, we have

\[ \bar{a}_{ii}(t) + \sum_{v_j \in N^-_i \cap R} \bar{a}_{ij}(t) = a_{ii}(t) + \sum_{v_j \in J_i(t)} a_{ij}(t) = 1, \]

which satisfies the first condition in the proposition. Furthermore,
since \( |J_i(t)| \geq |N^-_i| - 2F \) and each of the weights are
lower bounded by \( \eta \), this satisfies the second condition in the
proposition.

Now consider the case where there are one or more adversarial
nodes in \( J_i(t) \). We consider adversarial nodes in
\( J_i(t) \setminus \bar{J}_i(t) \) and \( \bar{J}_i(t) \) separately.

Consider any adversarial node \( v_m \in J_i(t) \setminus \bar{J}_i(t) \), and let
\( x_m(t) \) be the value received by node \( v_m \) from \( v_i \). Since \( v_i \) did
not discard \( v_m \)'s value, it must be the case that there are either \( F \)
values that are higher than \( x_m(t) \) in \( v_i \)'s neighborhood, or
\( v_i \)'s own value is higher than \( x_m(t) \). Similarly, there
must either be \( F \) values that are lower than \( x_m(t) \) in \( v_i \)'s
neighborhood, or \( v_i \)'s own value is lower than \( x_m(t) \). Since
there are at most \( F \) adversarial nodes in \( v_i \)'s neighborhood, we see that there is a pair of regular nodes \( v_u, v_l \in N^-_i \setminus \{v_i\} \)
with \( x_l(t) \leq x_m(t) \leq x_u(t) \). Thus, the term \( a_{im}(t)x_m(t) \) in
\( 8 \) can be written as

\[ a_{im}(t)x_m(t) = a_{im}(t)\gamma_m x_u(t) + a_{im}(t)(1 - \gamma_m)x_l(t) \]

for some \( \gamma_m \in [0, 1] \). By updating the weights \( \bar{a}_{il}(t) \) and
\( \bar{a}_{il}(t) \) as \( \bar{a}_{il}(t) \leftarrow \bar{a}_{il}(t) + a_{im}(t)\gamma_m \) and
\( \bar{a}_{il}(t) \leftarrow \bar{a}_{il}(t) + a_{im}(t)(1 - \gamma_m) \), respectively, the contribution of the adversarial node \( v_m \in J_i(t) \setminus \bar{J}_i(t) \) in \( 8 \) is transformed into contributions by two regular nodes. We do this for each adversarial node in
\( J_i(t) \setminus \bar{J}_i(t) \).

Now consider the set \( \bar{J}_i(t) \), containing \( |N^-_i| - 2F \) nodes. If
there are no adversarial nodes in \( \bar{J}_i(t) \), then the construction of the weights \( \bar{a}_{ij}(t) \) is complete and both conditions in the
proposition are satisfied (since the weights assigned to the
regular nodes in \( \bar{J}_i(t) \) satisfy the second condition in the
proposition by each being larger than \( \eta \)).

Thus suppose that there are \( K \) adversarial nodes in the set
\( \bar{J}_i(t) \), where \( 1 \leq K \leq F \) (recall that the set of
adversarial nodes is assumed to be \( F \)-local). Then there must
be at least \( K \) regular nodes in the set \( U_i(t) \), and at least
\( K \) regular nodes in the set \( \bar{L}_i(t) \). Label the \( K \) adversarial
nodes in \( \bar{J}_i(t) \) as \( \{v_{m1}, v_{m2}, \ldots, v_{mK}\} \), with corresponding
states \( x_{m1}(t), x_{m2}(t), \ldots, x_{mK}(t) \).

Pick any \( K \) regular nodes in \( U_i(t) \) and any \( K \) regular nodes in \( \bar{L}_i(t) \), and label them as \( \{v_{l1}, v_{l2}, \ldots, v_{lK}\} \), and
\( \{v_{l1}, v_{l2}, \ldots, v_{lK}\} \), respectively. We will label the states of these nodes as
\( x_{l1}(t), x_{l2}(t), \ldots, x_{lK}(t) \), and \( x_{u1}(t), x_{u2}(t), \ldots, x_{uK}(t) \), respectively.

By definition, we have \( x_{l1}(t) \leq x_{l2}(t) \leq x_{u1}(t) \)}
for all $1 \leq j \leq K$. Thus for each $j \in \{1, 2, \ldots, K\}$, we can write
\[ x_{m_j}(t) = \gamma_j x_{i_j}(t) + (1 - \gamma_j) x_{u_j}(t), \]
where $0 \leq \gamma_j \leq 1$. In other words, the state of the adversarial node $v_{m_j}$ is a convex combination of the states of the regular nodes $v_{i_j}$ and $v_{u_j}$. Note that either $\gamma_j$ or $(1 - \gamma_j)$ must be at least equal to 0.5.

As before, update the weights $\bar{a}_{ij}(t)$ and $\bar{a}_{iu}(t)$ as $\bar{a}_{ij}(t) \leftarrow \bar{a}_{ij}(t) + a_{ij}(t)$ and $\bar{a}_{iu}(t) \leftarrow \bar{a}_{iu}(t) + a_{iu}(t)(1 - \gamma_j)$ for $j \in \{1, 2, \ldots, K\}$. In other words, we split the value of the weight that was assigned to the adversarial node $m_j$ among the regular nodes $i_j$ and $u_j$, according to the proportions $\gamma_j$ and $(1 - \gamma_j)$. Note that at least $K$ of the nodes in $\{v_{i_1}, v_{i_2}, \ldots, v_{i_K}\} \cup \{v_{u_1}, v_{u_2}, \ldots, v_{u_K}\}$ get assigned a weight that is lower bounded by $\frac{1}{2}$ (since either $\gamma_j$ or $(1 - \gamma_j)$ is at least 0.5). Since the weight associated to each adversarial node is split according to a convex combination to a pair of regular nodes in $\mathcal{N}_-^\prime \setminus \mathcal{J}_i(t)$, we see that the first condition in the proposition is satisfied. Finally, since $\bar{a}_{ij}(t) \geq \bar{a}_{iu}(t) \geq \eta$ for $v_j \in \mathcal{J}_i(t) \cap \mathcal{R}$, this ensures that $|\mathcal{J}_i(t) - \mathcal{R}| = |\mathcal{N}_-^\prime| - 2F - K$ weights are lower bounded by $\eta$. As discussed above, the splitting of the adversarial nodes’ weights ensures that an additional $K$ regular nodes are assigned a weight that is lower bounded by $\frac{1}{2}$. Thus, in total, there are at least $|\mathcal{N}_-^\prime| - 2F$ weights (other than $\bar{a}_{ii}(t)$) that are lower bounded by $\frac{1}{2}$, concluding the proof.

We emphasize again that the regular nodes run the dynamics (8) (which does not require them to know which of their neighbors is adversarial); the dynamics (9) are mathematically equivalent to the dynamics (8) due to the nature of the local filtering that is done by each regular node, and will lead to certain insights that we will leverage.

Henceforth, we assume without loss of generality that the regular nodes are arranged first in the ordering of the nodes, and define
\[
\begin{align*}
x_{\mathcal{R}}(t) & \triangleq [x_1(t) \ x_2(t) \ \cdots \ x_{|\mathcal{R}|}(t)]', \\
d_{\mathcal{R}}(t) & \triangleq [d_1(t) \ d_2(t) \ \cdots \ d_{|\mathcal{R}|}(t)]',
\end{align*}
\]
to be the vectors of states and subgradients of the regular nodes, respectively. Based on Proposition 5.1, the dynamics of the regular nodes under the LF dynamics can be written as
\[
x_{\mathcal{R}}(t + 1) = \bar{A}(t)x_{\mathcal{R}}(t) - \alpha_{i}d_{\mathcal{R}}(t), \tag{10}
\]
where $\bar{A}(t) \in \mathbb{R}^{(|\mathcal{R}|)^2 \times |\mathcal{R}|}$ contains the weights $\bar{a}_{ij}(t)$ from (9).

VI. CONVERGENCE TO CONSENSUS

In this section, we study the convergence properties of the LF dynamics (3). In particular, we provide sufficient conditions for consensus for scenarios with $F$-local Byzantine adversaries (i.e., the most general class of adversaries that we consider), and necessary and sufficient conditions for scenarios with $F$-total malicious adversaries.

A. A Sufficient Condition for Consensus Under $F$-local Byzantine Adversaries

Theorem 6.1: Consider the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with regular nodes $\mathcal{R}$ and an $F$-local set of Byzantine nodes $\mathcal{A}$. Suppose the network is $(2F + 1)$-robust, that the functions $f_i, \ v_i \in \mathcal{R}$, have subgradients bounded by some constant $L$, and that the regular nodes run the LF dynamics (8) with parameter $F$. Further suppose that $\alpha_t \rightarrow 0$ as $t \rightarrow \infty$. Then, there exists a sequence of stochastic vectors $q_t, \ t \in \mathbb{N}$, such that
\[
\limsup_{t \rightarrow \infty} \|x_{\mathcal{R}}(t) - y(t)\| = 0,
\]
where $y(t) = q_t' x_{\mathcal{R}}(t)$.

Proof: Consider the LF dynamics (3), and their equivalent matrix representation (10). By Proposition 5.1 we know the following facts about the dynamics matrix $A(t)$ at each time-step $t \in \mathbb{N}$: each diagonal element is lower bounded by $\eta$, and for each row $i \in \{1, 2, \ldots, |\mathcal{R}|\}$, at least $|\mathcal{N}_-| - 2F$ elements are lower bounded by $\frac{1}{2}$. Consider the graph $\mathcal{G}$, and remove all edges whose weights are smaller than $\frac{1}{2}$ in $\bar{A}(t)$; note that this removes all edges from adversarial nodes to regular nodes (since they do not show up at all in $\bar{A}(t)$). For each regular node $v_i \in \mathcal{R}$, note that at most $2F$ incoming edges are removed, again since at least $|\mathcal{N}_-| - 2F$ elements are lower bounded by $\frac{1}{2}$. Now, from Lemma 2.3, we see that if the graph $\mathcal{G}$ is $(2F + 1)$-robust, the subgraph consisting of regular nodes will be rooted after removing $2F$ or fewer edges from each regular node. Thus, $\bar{A}(t)$ is rooted for each $t \in \mathbb{N}$, with a tree whose edge-weights are all lower bounded by $\frac{1}{2}$ (and whose diagonal elements are also lower bounded by $\frac{1}{2}$). The theorem then follows by applying the first part of Lemma 5.4.

The above proof relied on the fact that in $(2F + 1)$-robust networks, the weight matrix $\bar{A}(t)$ corresponding to the regular nodes is rooted at each time-step (under the $F$-local adversary model). This is only a sufficient condition; we now show that under the $F$-total malicious model, one can in fact give a necessary and sufficient condition on the graph topology in order to guarantee consensus, but that rootedness is no longer guaranteed at each time-step under such conditions. We will then provide an alternate proof of convergence to consensus for such graphs.

B. A Necessary and Sufficient Condition for Consensus Under $F$-total Malicious Adversaries

We start with the following example showing that when the network is not $(2F + 1)$-robust, the graph induced by the filtering operation may not be rooted at each time-step.

Example 6.2: Consider the graph of Figure 11(a), where all nodes are regular and use the LF dynamics (8) with $F = 1$. Let us assume that all nodes have identical objective functions given by $f(x) = |x|$, and that the initial values of the nodes are as displayed inside the circles. One can verify that this graph is only 2-robust: if we take each of the nodes with value 1 to be the sets $S_1$ and $S_2$, then no node in either set has more
than 2 neighbors outside its set. Thus Theorem 6.1 cannot be
equalled to apply to prove consensus. Indeed, we will show that graph
induced by the LF dynamics may not be rooted at each time-
step. Figure 1(b) shows the information that is used by each
node after the filtering operation. For example, the node with
value 2 has disregarded one of its neighbors with value 0,
which is lower than its own value. However, since the node
with value 2 does not have any neighbors with values larger
than its own value, it does not remove any other values. Similarly
each node with value 1 removes the value 2 and the value 0,
as they are the single highest and single lowest values in its
neighborhood at this time-step. The directed graph induced
by the filtering operation is clearly not rooted; nevertheless, as we
show later in Theorem 6.4, the regular nodes are guaranteed
to achieve consensus in this network under the dynamics (8),
even if any single node becomes malicious.

This example motivates us to use a different strategy for
establishing the convergence properties of the LF dynamics
8. More importantly, our alternate approach will allow us to
show that the notion of $(r, s)$-robustness given in Definition 2.2
yields a necessary and sufficient condition for consensus
in scenarios with $F$-total malicious adversaries. In order to
establish this result, we need to define the following quantities:

$$M(t) \triangleq \max_{v_i \in \mathcal{R}} x_i(t), \quad m(t) \triangleq \min_{v_i \in \mathcal{R}} x_i(t),$$

and

$$D(t) \triangleq M(t) - m(t).$$

For each $t \in \mathbb{N}$, we set

$$\delta_t \triangleq \sup_{\ell \geq t} |\alpha_{\ell}| L,$$

where $L$ is the upper bound on the magnitude of the subgra-
dients. Clearly $|\alpha_{\ell} d_i(\ell)| \leq \delta_t$ for all $\ell \geq t$. For any $\gamma \in \mathbb{R}$ and
t, $\bar{t} \in \mathbb{N}$ with $\bar{t} \geq t$, define the sets

$$\mathcal{M}_M(t, \bar{t}, \gamma) \triangleq \{ v_i \in V \mid x_i(\bar{t}) > M(t) - \gamma \}$$

and

$$\mathcal{M}_m(t, \bar{t}, \gamma) \triangleq \{ v_i \in V \mid x_i(\bar{t}) < m(t) + \gamma \}.$$

A key to the proof will be the following simple fact: at any
time-step $t$, no regular node will ever use a value larger than
$M(t)$ or smaller than $m(t)$ in its update equation (8). This is
easy to see as follows. If the filtering operation by regular node
$v_{i1}$ removes all of the adversarial nodes in its neighborhood
at time-step $t$, then clearly all remaining nodes in $\mathcal{J}_i(t)$ are
regular, and thus have values in the interval $[m(t), M(t)]$. On

the other hand, if a regular node uses a value of an adversarial
node, then under the $F$-local model, there must be at least one
regular node in $v_i$’s neighborhood that had value larger than
the value of the adversarial node, and at least one regular node
in the neighborhood that had value smaller than the adversarial
node’s (these values could potentially be $v_i$’s own value). Thus
again, we see that all of the values used by $v_i$ at time-step $t$
are in the interval $[m(t), M(t)]$. We are now ready to show
the following result.

**Proposition 6.3:** Consider the network $\mathcal{G} = (V, E)$, with
regular nodes $\mathcal{R}$ and adversarial nodes $\mathcal{A}$. Suppose the
adversarial nodes are $F$-total malicious and the network is
$(F + 1, F + 1)$-robust. Further suppose that the functions $f_i$,
v_i \in \mathcal{R}$ have subgradients bounded by some constant $L$, and
that the regular nodes run the Local Filtering dynamics (8)
with parameter $F$ and with weights lower bounded by $\eta$. Then
for any $t \in \mathbb{N}$, we have

$$\eta \leq (1 - \frac{\eta |\mathcal{R}|}{2}) D(t) + 2|\mathcal{R}| \delta_t. \quad (11)$$

**Proof:** Consider any time-step $t \in \mathbb{N}$. Define $\gamma_0 = \frac{D(t)}{2}$. Note that the sets $\mathcal{M}_M(t, t, \gamma_0)$ and $\mathcal{M}_m(t, t, \gamma_0)$ are disjoint.

By the definition of these sets, they each contain at least
one regular node when $D(t) > 0$ (i.e., the nodes that have
value $M(t)$ and $m(t)$, respectively). Since the graph is $(F +
1, F + 1)$-robust, and since there are at most $F$ adversarial
nodes, there is at least one regular node in either $\mathcal{M}_M(t, t, \gamma_0)$
or $\mathcal{M}_m(t, t, \gamma_0)$ (or both) that has $F + 1$ neighbors outside
its set. Since each regular node only discards up to $F$ values that
are smaller (or larger) than its own value, there will be at least
one regular node that uses the value of a node from outside
its set. Suppose that there is such a regular node $v_i$ in the set
$\mathcal{M}_M(t, t, \gamma_0)$. Then, the value of this node at the next time-step
is upper bounded as

$$x_i(t + 1) \leq (1 - \eta) M(t) + \eta (M(t) - \gamma_0) + \delta_t = M(t) - \eta \gamma_0 + \delta_t.$$

The above bound is obtained by noting that the smallest
possible weight that a node can assign to a used value is $\eta$
(according to the description of the LF dynamics (8)). Note
that the above expression is also an upper bound for any
regular node that is not in $\mathcal{M}_M(t, t, \gamma_0)$, since such a node
will use its own value in its update.

Similarly, if there is a regular node $v_j \in \mathcal{M}_m(t, t, \gamma_0)$ that
uses the value of a node outside that set, then its value at the
next time-step is lower-bounded as

$$x_j(t + 1) \geq (1 - \eta) m(t) + \eta (m(t) + \gamma_0) - \delta_t = m(t) + \eta \gamma_0 - \delta_t.$$

Again, this is also a lower bound for the value of any regular
node that is not in the set $\mathcal{M}_m(t, t, \gamma_0)$.

Now, define the quantity $\gamma_1 = \eta \gamma_0 - \delta_t$ and note that this
is smaller than $\gamma_0$. Thus, the sets $\mathcal{M}_M(t, t + 1, \gamma_1)$ and $\mathcal{M}_m(t, t + 1, \gamma_1)$ are disjoint.
Since the former set is empty; this means that this bound also holds for any regular node that is not in $X$. At each time step are nonempty, then again by the fact that the graph is $(F + 1, F + 1)$-robust, there is at least one regular node in at least one of these sets that has $F + 1$ neighbors outside the set. Suppose that $v_i \in X_M(t, t + 1, \gamma_1) \cap R$ is such a node. As above, this node’s value at the next time-step is upper bounded as

$$x_i(t + 2) \leq (1 - \eta)M(t + 1) + \eta(M(t) - \gamma_1) + \delta_t$$

$$\leq (1 - \eta)(M(t) + \delta_t) + \eta(M(t) - \gamma_1) + \delta_t$$

$$= M(t) + (2 - \eta)\delta_t - \eta\gamma_1$$

$$= M(t) + 2\delta_t - \eta^2\gamma_0.$$  

where the first inequality holds since the smallest possible weight that node $v_i$ can assign to the (undiscounted) value of a neighbor outside $X_M(t, t + 1, \gamma_1)$ is $\eta$, and the value of this neighbor, by construction, is at most $M(t) - \gamma_1$. Again, this upper bound also holds for any regular node that is not in $X_M(t, t + 1, \gamma_1) \cap R$. Similarly, if there is a node $v_j \in X_m(t, t + 1, \gamma_1) \cap R$ that has $F + 1$ neighbors outside that set, its next value is lower bounded as

$$x_j(t + 2) \geq (1 - \eta)m(t + 1) + \eta(m(t) + \gamma_1) - \delta_t$$

$$\geq (1 - \eta)(m(t) - \delta_t) + \eta(m(t) + \gamma_1) - \delta_t$$

$$= m(t) - (2 - \eta)\delta_t + \eta\gamma_1$$

$$= m(t) - 2\delta_t + \eta^2\gamma_0.$$  

This bound also holds for any regular node that is not in the set $X_m(t, t + 1, \gamma_1) \cap R$. We continue in this manner by defining $\gamma_k = \eta^k\gamma_0 - k\delta_t$. At each time step $t + k$, if both $X_M(t, t + k, \gamma_k) \cap R$ and $X_m(t, t + k, \gamma_k) \cap R$ are nonempty, then at least one of these sets will shrink in the next time-step. If either of the sets is empty, then it will stay empty at the next time-step, since every regular node outside that set will have its upper bound updated by $M(t) - \gamma_k$ (or lower bounded by $m(t) + \gamma_k$). After $|R|$ time-steps, at least one of the sets $X_M(t, t + |R|, \gamma_k) \cap R$ or $X_m(t, t + |R|, \gamma_k) \cap R$ is guaranteed to be empty. Suppose the former set is empty; this means that

$$M(t + |R|) \leq M(t) - \gamma_{|R|}.$$  

Since $m(t + |R|) \geq m(t) - |R|\delta_t$, we obtain

$$D(t + |R|) \leq D(t) - \gamma_{|R|} + |R|\delta_t$$

$$= \left(1 - \frac{\eta^{|R|}}{2}\right) D(t) + 2|R|\delta_t.$$  

The same expression arises if the set $X_m(t, t + |R|, \gamma_{|R|}) \cap R$ is empty, concluding the proof.

The above proposition leads to the following result for consensus of the gradient-based distributed optimization dynamics under local-filtering rules.

**Theorem 6.4:** Consider the network $G = (V, E)$, with regular nodes $R$ and an $F$-total set of malicious nodes $A$. Suppose that the functions $f_r$, $v_i \in R$, have subgradients bounded by some constant $L$, and that the regular nodes run the Local Filtering dynamics (8) with parameter $F$ and weights lower bounded by $\eta$. Suppose further that the step-sizes satisfy $\alpha_t \to 0$. Then the regular nodes are guaranteed to reach consensus despite the actions of the adversaries, initial values, and local functions if and only if the graph is $(F + 1, F + 1)$-robust. \hfill \Box

**Proof:** The proof of sufficiency follows immediately from Proposition 6.3 by noting that when $\alpha_t \to 0$, then $\delta_t \to 0$ as $t \to 0$. By input-to-state stability, we have $D(t) \to 0$ as $t \to 0$, proving consensus.

For necessity, suppose that the network is not $(F + 1, F + 1)$-robust. Then there exist two disjoint nonempty sets $S_1, S_2 \subset V$ such that (i) there is at least one node in $S_1$ that has at most $F$ neighbors outside $S_1$, (ii) there is at least one node in $S_2$ that has at most $F$ neighbors outside $S_2$, and (iii) there are at most $F$ nodes in $S_1 \cup S_2$ that have $F + 1$ or more neighbors outside their respective sets. Choose the nodes in $S_1 \cup S_2$ that each have $F + 1$ or more neighbors outside their respective sets to be the adversarial set $A$; clearly $A$ is an $F$-total set. Now, assign all of the nodes in set $S_1$ to have function $f_1$, and assign all of the nodes in set $S_2$ to have function $f_2$, where the minimizer of $f_2$ is strictly larger than the minimizer of $f_1$. Now let all of the nodes in set $V \{S_1 \cup S_2\}$ have function $f_3$, selected to have gradient equal to zero in the entire interval bracketed by the minimizers of $f_1$ and $f_2$. Let all nodes in $S_1$ and $S_2$ (including the adversarial nodes) be initialized at their local minimizers, and let all nodes in $V \{S_1 \cup S_2\}$ be initialized at a value strictly between the minimizers of $f_1$ and $f_2$. Furthermore, let the malicious nodes never change their values. In this case, all regular nodes in $S_1$ will discard all of their neighbors’ values from outside $S_1$ (since they each have at most $F$ neighbors outside $S_1$), and similarly all regular nodes in $S_2$ will discard all of their neighbors’ values from outside $S_2$. As the values of nodes in $V \{S_1 \cup S_2\}$ will always remain strictly between the minimizers of $f_1$ and $f_2$, the regular nodes in $S_1$ and $S_2$ will never deviate from their initial values, and thus consensus will not be reached. \hfill \blacksquare

The above result shows that the network considered in Example 6.4 is guaranteed to facilitate consensus among the regular nodes despite the presence of any single malicious node (since the network is $(2, 2)$-robust), even though the graph induced by the filtering operation is not rooted at each time-step.

**Remark 6.5:** As illustrated by Theorems 6.1 and 6.4, the properties of $r$-robustness and $(r, s)$-robustness play a key role in consensus-based optimization dynamics of the form (8). While these properties are stronger than other graph properties such as $r$-minimum degree and $r$-connectivity, all of these properties occur simultaneously in various commonly studied models for large-scale networks [22]. There are also various simple techniques to construct $r$-robust networks for any given $r \in \mathbb{N}$, as discussed in [18]. \hfill \Box
VII. A SAFETY CONDITION: CONVERGENCE TO THE
CONVEX HULL OF THE LOCAL MINIMIZERS

In the previous section, we provided graph properties that
guaranteed consensus for the regular nodes under the LF
dynamics (8) (under the condition that the step-sizes
asymptotically go to zero). In this section, we provide a safety
guarantee on these dynamics under additional conditions on
the step-sizes, as detailed in the following theorem.

Theorem 7.1: Suppose that one of the following conditions
holds:

(i) The adversarial nodes are F-total malicious and the
network is \((F + 1, F + 1)\)-robust; or
(ii) The adversarial nodes are F-local Byzantine and the
network is \((2F + 1)\)-robust.

Suppose that all regular nodes follow the LF dynamics (8) with
parameter F. For each node \(v_i \in \mathcal{R}\), let the local function \(f_i\) be
convex, continuous and have subgradients bounded by \(L\). Let
the set of minimizers of \(f_i\) be denoted by the closed convex
set \(\mathcal{M}_i \subseteq \mathbb{R}\). Define \(\overline{M} = \max_{v_i \in \mathcal{R}} \max\{x \mid x \in \mathcal{M}_i\}\) and
\(\underline{M} = \min_{v_i \in \mathcal{R}} \min\{x \mid x \in \mathcal{M}_i\}\). If the step-sizes satisfy
\(\sum \alpha_t = \infty\) and \(\alpha_t \to 0\), then \(\limsup_{t \to \infty} x_i(t) \leq \overline{M}\) and
\(\liminf_{t \to \infty} x_i(t) \geq \underline{M}\) for all \(v_i \in \mathcal{R}\), regardless of the actions
of the adversarial nodes and the initial values.

Proof: Let \(M(t)\) and \(m(t)\) be the maximum and minimum
values of the regular nodes at time-step \(t\), respectively.
Under the conditions of the theorem, Theorems 6.1 and 6.4
indicate that \(M(t) - m(t) \to 0\). Now consider the local
filtering dynamics (8). Since no regular node ever adopts
a neighbor’s value larger than \(M(t)\) in its update, we have
\[
x_i(t + 1) = a_{ii}(t)x_i(t) + \sum_{v_j \in \mathcal{N}_i(t)} a_{ij}(t)x_j(t) - \alpha_t d_i(t),
\]
\[
\leq a_{ii}(t)M(t) + \sum_{v_j \in \mathcal{N}_i(t)} a_{ij}(t)M(t) - \alpha_t d_i(t)
\]
\[
= M(t) - \alpha_t d_i(t),
\]
for each regular node \(v_i \in \mathcal{R}\). In particular, we have
\[
M(t + 1) \leq M(t) - \alpha_t \min_{v_i \in \mathcal{R}} d_i(t).
\] (12)

Iterating, we obtain for any \(T \in \mathbb{Z}_{\geq 1}\)
\[
M(t + T) \leq M(t) - \sum_{j = t}^{t+T-1} \alpha_j \min_{v_i \in \mathcal{R}} d_i(j).
\] (13)

Now suppose by way of contradiction that \(\limsup_{t \to \infty} M(t) = \overline{M} + \delta\) for some \(\delta > 0\). Let \(t_0\) be
such that the following three conditions are satisfied:

(i) \(\overline{M} + {\delta\over 2} \leq M(t_0) \leq \overline{M} + 2\delta\),
(ii) \(M(t) - m(t) \leq {\delta\over 2}\) for all \(t \geq t_0\), and
(iii) \(\alpha_t L \leq {\delta\over 4}\) for all \(t \geq t_0\).

Such a \(t_0\) is guaranteed to exist by the convergence of \(M(t) - m(t)\)
to zero and the definition of \(\delta\). Define
\[
G = \min_{v_i \in \mathcal{R}} \frac{df_i}{dx} \bigg|_{\overline{M} + {\delta\over 2}}.
\]
If \(f_i\) is not differentiable at \(\overline{M} + {\delta\over 2}\), we consider the infimum
of its subgradients at that point (note that all such subgradients
will be positive and bounded away from zero). Thus, we have
\(d_i(t) \geq G > 0\) whenever \(m(t) \geq \overline{M} + {\delta\over 2}\). By the definition of \(t_0\) and using (13), we have
\[
M(t_0 + T) \leq M(t_0) - G \sum_{j = t_0}^{t_0+T-1} \alpha_j,
\]
for any \(T\) such that \(M(t) \geq \overline{M} + {\delta\over 2}\) for all \(t \in [t_0, t_0 + T)\). Thus, using the fact that \(\sum_{j = t_0}^{t_0+T-1} \alpha_j\) is unbounded in \(T\), we see
that \(M(t_0 + T) \leq \overline{M} + 2\delta\) for sufficiently large \(T\). Let \(t_1\)
be that point in time.

Now we show that \(M(t)\) will never exceed \(\overline{M} + {3\delta\over 4}\) after
time \(t_1\). Specifically, if \(M(t) \leq \overline{M} + {\delta\over 2}\) at some time \(t \geq t_1\),
then by (12), we have
\[
M(t + 1) \leq M(t) + \alpha_t L \leq \overline{M} + {\delta\over 2} + {\delta\over 4} = \overline{M} + {3\delta\over 4}.
\]
Similarly, if \(M(t) \geq \overline{M} + {\delta\over 2}\) at some time \(t \geq t_1\), then by
(12), we have \(M(t + 1) \leq M(t) - \alpha_t G\), and thus \(M(t)\) will
monotonically decrease until it is below \(\overline{M} + {\delta\over 2}\). Thus, \(M(t)\)
will eventually be upper bounded by \(\overline{M} + {3\delta\over 4}\), contradicting the definition of \(\delta\). Thus, \(\limsup_{t \to \infty} M(t) \leq \overline{M}\). An identical
argument holds for the lower bound.

A. Lack of Convergence to a Constant Value Under Adversarial
Behavior

As shown in the previous result, the LF dynamics guarantee
consensus within the convex hull of the local minimizers and
prevent the adversarial nodes from driving the states of regular
nodes to arbitrarily large values under appropriate conditions
on the network topology. However, a single malicious node can
still prevent the regular nodes from converging to a constant
value under certain classes of step-sizes. This is illustrated in
the following example.

Example 7.2: Consider a complete graph \(G\) with five nodes
\(\{v_1, v_2, v_3, v_4, v_5\}\). Suppose \(v_1, v_2\) and \(v_3\) all have local functions
\(f_a(x) = x^2\), and \(v_4\) has local function \(f_b(x) = (x - 9)^2\) (with the magnitude of their gradients capped at \(L\), for some
sufficiently large \(L\)). Suppose node \(v_5\) is malicious.

Let all regular nodes start at their local minimizers and
run the dynamics (8) with step-sizes satisfying \(\sum \alpha_t = \infty\) and
\(\sum \alpha_t^2 < \infty\). Let the malicious node behave as follows
(illustrated in Figure 2). It starts by keeping its value the same
as the regular nodes \(v_3, v_2\) and \(v_1\). In this case, those regular
nodes all discard node \(v_4\)’s value as being too extreme, and
thus all regular nodes converge towards the minimizer of \(f_a\),
namely 0. When node \(v_4\)’s value is sufficiently close to 0, the
malicious node switches its value to be larger than \(v_4\)’s value
(as shown just after time-step 100 in Figure 2). At this point,
all regular nodes discard \(v_5\)’s value as being too extreme.
functions are drawn, and suppose that local set. Let be any distributed optimization algorithm that guarantees that all
regular nodes reach consensus on a value in the convex hull of the minimizers of the regular nodes’ objective functions. Let be the true minimizer of the average of the functions held by all regular nodes, and let be the value computed by the regular nodes under . Then, under the -local adversary model, there is an allocation of functions to nodes such that , where is the value of the average of the functions held by the regular nodes evaluated at .

Proof: We consider two scenarios. In the first scenario, let each node in have the local function , and let each node in have the local function . Let all nodes be regular. The minimizer of the average of all functions is given by . Let .

In the second scenario, the nodes in set are also assigned the function , but are adversarial and execute the algorithm by pretending their local functions are . Since guarantees that all regular nodes reach consensus in the convex hull of the minimizers of the regular nodes’ functions, all regular nodes must obtain the value after executing algorithm .

Since the two scenarios are indistinguishable from the perspective of , the algorithm must also cause all regular nodes to calculate as the first scenario. Thus, the difference of the value output by and the true minimizer of the regular nodes’ functions is , and the difference in achieved costs is .

Example 8.2: Consider the network shown in Figure where is some positive integer, and . We define the vertex sets and . One can verify that this network is -robust and that is a maximum -local set. Suppose each node in is assigned the function , and each node in is assigned the function . By Proposition any algorithm that guarantees to output a value in the convex hull of the regular nodes’ minimizers must produce as a solution. In this case, we have and .

Given the fact that the performance of resilient distributed optimization algorithms heavily depends on the size of maximum -local sets in the network (under the -local adversary

Fig. 3: Graph constructed on node sets and . All nodes in set are connected to each other (the edges are not shown in the interest of clarity). Each node in set connects to three unique vertices in set . This graph is -robust.

Fig. 2: An illustration of lack of convergence to a constant value under adversarial behavior.

VIII. FACTORS THAT AFFECT THE PERFORMANCE OF RESILIENT DISTRIBUTED OPTIMIZATION ALGORITHMS

The proof of Theorem 4.4 indicates that the nature of the individual optimization functions (together with the network topology) will play a role in determining the performance that is achievable under adversarial behavior. For example, suppose that all individual objective functions are drawn from a certain class of functions . In the trivial case where all functions in have the same minimizer, each node can calculate the globally optimal value simply by calculating the minimizer of its own function, and thus resilience to any number of adversarial nodes is guaranteed. On the other hand, when the class of functions is sufficiently rich so that the function held by each node contributes to the global minimizer, then the number and location of adversarial nodes will play a larger role in determining the achievable performance. One such bound on performance is provided by the following result.

Proposition 8.1: Consider a network with nodes and let be a maximum -local set. Let be the set from which the local objective functions are drawn, and suppose that , where and for . Let be any distributed optimization algorithm that guarantees that all functions can be modified to have their gradients capped at sufficiently large values, so as to not affect the minimizer of any convex combination of the functions.

Both functions can be modified to have their gradients capped at sufficiently large values, so as to not affect the minimizer of any convex combination of the functions.
model), it is natural to ask how easy it is to find such maximum sets. To answer this, we first define the problem formally and then characterize its complexity.

**Definition 8.3:** Let \( r, k \) be positive integers. The \( r \)-Local Set Problem is to determine whether a given graph has an \( r \)-local set of size at least \( k \).

**Theorem 8.4:** The \( r \)-Local Set Problem is NP-complete.

The proof of the above theorem is given in Appendix A.

Although finding maximum \( F \)-local sets in graphs is difficult in general (unless \( P = NP \)), one can characterize the size of such sets in certain specific classes of graphs. For instance, the maximum \( F \)-local set in complete graphs has size exactly \( F \). Similarly, consider Erdős-Rényi random graphs where each edge between each pair of nodes is added independently with a certain probability \( p(n) \) (which could depend on the number of nodes in the graph). It was shown in \([22], [29] \) that if the edge probability satisfies

\[
p(n) = \frac{\ln(n) + F \ln \ln(n) + g(n)}{n},
\]

where \( g(n) \to \infty \) as \( n \to \infty \), the size of the largest \( F \)-local set is in \( \mathcal{O}(n^{\gamma(n)}) \) with high probability, where \( \gamma(n) \) is any function satisfying \( \ln \ln(n) = o(\gamma(n) \ln n) \). For instance, \( \gamma(n) = \frac{\ln \ln(n)^{1+\epsilon}}{\ln n} \) satisfies this for any \( \epsilon > 0 \). Thus, with high probability, the fraction of nodes that are in the maximum \( F \)-local set goes to zero as \( n \to \infty \) in Erdős-Rényi random graphs for the above regime of edge probabilities. This means that the limitation identified in Proposition 8.1 will not play a major role in such graphs. An interesting avenue for further research is to identify whether there are other graph theoretic obstructions to the performance of resilient distributed optimization algorithms (including the LF dynamics we have presented in this paper).

**IX. DIRECTIONS FOR FUTURE RESEARCH**

In this paper, we proposed a consensus-based distributed optimization algorithm that is resilient to adversarial behavior under certain conditions on the network topology, in the sense that the regular nodes will always asymptotically converge to the convex hull of the minimizers of the regular nodes’ functions, despite the actions of any \( F \)-local set of adversaries. We also identified topological properties (in the form of maximum \( F \)-local sets) that affect the performance of the algorithm. There are many interesting directions for future research, including a more explicit characterization of the distance-to-optimality of such algorithms (with corresponding conditions on the network topology), along with a characterization of classes of functions that lead to near-optimal solutions.

**REFERENCES**

[1] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[2] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” in *Symposium on Information Processing of Sensor Networks*, Berkeley, CA, Apr. 2004, pp. 20–27.

[3] L. Xiao and S. Boyd, “Optimal scaling of a gradient method for distributed resource allocation,” *Journal of Optimization Theory & Applications*, vol. 129, no. 3, pp. 409–488, 2006.

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

[5] P. Wan and M. D. Lemmon, “Event-triggered distributed optimization in sensor networks,” in *Symposium on Information Processing of Sensor Networks*, San Francisco, CA, 2009, pp. 49–60.

[6] A. Nedic, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[7] B. Johansson, M. Rabi, and M. Johansson, “A randomized incremental subgradient method for distributed optimization in networked systems,” *SIAM Journal on Control and Optimization*, vol. 20, no. 3, pp. 1157–1170, 2009.

[8] 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, 2012.

[9] J. Wang and N. Elia, “Control approach to distributed optimization,” in *Allerton Conf. on Communications, Control and Computing*, Monticello, IL, Oct. 2010, pp. 557–561.

[10] ——, “A control perspective for centralized and distributed convex optimization,” in *IEEE Conf. on Decision and Control*, Orlando, Florida, 2011, pp. 3800–3805.

[11] 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, 2014.

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

[13] L. Lampert, R. Shostak, and M. Pease, “The Byzantine Generals problem,” *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, July 1982.

[14] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann Publishers, Inc., 1996.

[15] S. Sundaram and C. N. Hadjicostis, “Distributed function calculation via linear iterative strategies in the presence of malicious agents,” *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1495–1508, 2011.

[16] N. H. Vaidya, L. Tseng, and G. Liang, “Iterative approximate Byzantine consensus in arbitrary directed graphs,” in *ACM Symposium on Principles of Distributed Computing*, 2012, pp. 365–374.

[17] F. Pasqualetti, A. Bicchi, and F. Bullo, “Consensus computation in unreliable networks: A system theoretic approach,” *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90–104, Jan. 2012.

[18] 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, April 2013.

[19] L. Su and N. Vaidya, “Byzantine multi-agent optimization,” *arXiv preprint arXiv:1506.04681*, 2015.

[20] S. Sundaram and B. Gharesifard, “Consensus-based distributed optimization with malicious nodes,” in *Allerton Conf. on Communications, Control and Computing*, 2015, to appear.

[21] ——, “Secure local filtering algorithms for distributed optimization,” in *IEEE Conf. on Decision and Control*, 2016, submitted.

[22] H. Zheg, E. Fata, and S. Sundaram, “A notion of robustness in complex networks,” *IEEE Transactions on Control of Network Systems*, vol. 2, no. 3, pp. 310–320, 2015.

[23] A. Nedić and A. Ozdaglar, “Cooperative distributed multi-agent optimization,” in *Convex optimization in signal processing and communications*, D. P. Palomar and Y. C. Eldar, Eds. Cambridge University Press, 2010, pp. 340–386.

[24] M. Cao, S. A. Morse, and B. D. O. Anderson, “Reaching a consensus in a dynamically changing environment: a graphical approach,” *SIAM Journal on Control and Optimization*, vol. 47, no. 2, pp. 575–600, 2008.

[25] J. Ghaderi and R. Srikant, “Opinion dynamics in social networks with stubborn agents: Equilibrium and convergence rate,” *Automatica*, vol. 50, no. 12, pp. 3209–3215, 2014.

[26] E. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, and A. Scaglione, “Binary opinion dynamics with stubborn agents,” *ACM Transactions on Economics and Computation*, vol. 1, no. 4, p. 19, 2013.

[27] D. Dolev, N. A. Lynch, S. Pinter, E. W. Stark, and W. E. Weihl, “Reaching approximate agreement in the presence of faults,” *Journal of the ACM*, vol. 33, pp. 499–516, May 1986.
Next, note that any vertex neighbors in $P$ vertices from the set of the Set Packing problem. This completes the construction of "yes." Then there exists a collection of at least $P$ such that no two of the subsets share an element. Let constructed instance of the $k \leq i \leq m$ can be in $P_v$. Thus $P_v$ is of size 1, contradicting the fact that it is a 1-local set of size at least 2.

Thus, $P_v$ can contain only vertices from the set $\{s_1, s_2, \ldots, s_m\}$. It is now easy to see that the subsets from the Set Packing problem corresponding to those vertices form a packing of size at least $k$, and thus the answer to the Set Packing problem is "yes."

The above reduction shows that the $r$-local Set Problem is NP-hard. Since this problem has a certificate for "yes" instances that can be verified in polynomial time (i.e., the actual $r$-local set of size at least $k$), the $r$-local Set Problem is in NP, and thus is NP-complete.

**APPENDIX A**

**PROOF OF THEOREM 8.4: COMPLEXITY OF FINDING MAXIMUM $r$-LOCAL SETS**

**Proof:** We will provide a reduction from the NP-complete Set Packing problem: given a collection of elements $U = \{u_1, u_2, \ldots, u_n\}$, a set of subsets $S = \{S_1, \ldots, S_m\}$ of $U$, and a positive integer $k$, do there exist $k$ subsets in $S$ that are mutually disjoint? Specifically, we will show that given any instance of the Set Packing problem, one can construct a graph $G = (V, E)$ in such a way that $G$ contains a 1-local set of size at least $k$ if and only if the answer to the given instance of the Set Packing problem is "yes." We assume throughout that $k \geq 2$, as the answer to the Set Packing problem for $k = 1$ is always "yes."

Construct the graph $G$ as follows. Define the vertex set $V$ to consist of $n + m$ vertices

$$V = \{u_1, u_2, \ldots, u_n, s_1, s_2, \ldots, s_m\},$$

where each vertex $u_i$ corresponds to an element of the set $U$, and each vertex $s_i$ corresponds to the subset $S_i \in S$.

Next, place an edge between each pair of vertices $u_i, u_j, j \neq i$. This creates a complete graph on the vertex set $\{u_1, \ldots, u_n\}$. For each vertex $s_i, 1 \leq i \leq m$, add an edge between $s_i$ and vertex $u_j$ if $u_j \in S_i$ in the given instance of the Set Packing problem. This completes the construction of the graph $G$.

Suppose that the answer to the Set Packing instance is "yes." Then there exists a collection of at least $k$ subsets such that no two of the subsets share an element. Let $P = \{S_i, S_{i_2}, \ldots, S_{i_k}\}$ be the corresponding collection, where $k' \geq k$. Let $P_v = \{s_i, s_{i_2}, \ldots, s_{i_k}\} \subset V$ be the corresponding vertices in graph $G$. Then it is easy to verify that $P_v$ forms a 1-local set of size $k'; k'$ none of the vertices $\{u_1, u_2, \ldots, u_n\}$ have more than one neighbor in $P_v$ (by the definition of the edges and the fact that $P_v$ corresponds to a packing), and none of the vertices $s_i$ share any edges with nodes in the set $P_v$. Thus, if the answer to the Set Packing instance is "yes", the answer to the constructed instance of the 1-local Set Problem is "yes."

We now show the converse. Suppose the answer to the constructed instance of the 1-local Set Problem is "yes," i.e., there exists a 1-local set $P_v \subset V$ of vertices, with cardinality $k' \geq k \geq 2$. We first claim that $P_v$ cannot contain any vertices from the set $\{u_1, u_2, \ldots, u_n\}$. To see this, note that $P_v$ cannot contain all of the vertices $\{u_1, u_2, \ldots, u_n\}$, for if it did, any vertex $s_i$ that is not in $P_v$ would contain at least two neighbors in $P_v$, contradicting the fact that it is a 1-local set. Next, note that $P_v$ cannot contain more than one node from $\{u_1, u_2, \ldots, u_n\}$, for if it did, any node $u_j$ that is not in $P_v$ would have more than one neighbor in $P_v$, again contradicting the fact that it is a 1-local set. Thus suppose $P_v$ contains a single vertex from $\{u_1, \ldots, u_n\}$, and take this vertex to be $u_i$. Then each vertex $u_j (j \neq i)$ already has a neighbor in $P_v$, and thus none of the vertices $s_i, 1 \leq i \leq m$ can be in $P_v$. Thus $P_v$ is of size 1, contradicting the fact that it is a 1-local set of size at least 2.

Next, place an edge between each pair of vertices $u_i, u_j, j \neq i$. This creates a complete graph on the vertex set $\{u_1, u_2, \ldots, u_n\}$. For each vertex $s_i, 1 \leq i \leq m$, add an edge between $s_i$ and vertex $u_j$ if $u_j \in S_i$ in the given instance of the Set Packing problem. This completes the construction of the graph $G$.

**APPENDIX A**

**PROOF OF THEOREM 8.4: COMPLEXITY OF FINDING MAXIMUM $r$-LOCAL SETS**

**Proof:** We will provide a reduction from the NP-complete Set Packing problem: given a collection of elements $U = \{u_1, u_2, \ldots, u_n\}$, a set of subsets $S = \{S_1, \ldots, S_m\}$ of $U$, and a positive integer $k$, do there exist $k$ subsets in $S$ that are mutually disjoint? Specifically, we will show that given any instance of the Set Packing problem, one can construct a graph $G = (V, E)$ in such a way that $G$ contains a 1-local set of size at least $k$ if and only if the answer to the given instance of the Set Packing problem is "yes." We assume throughout that $k \geq 2$, as the answer to the Set Packing problem for $k = 1$ is always "yes."

Construct the graph $G$ as follows. Define the vertex set $V$ to consist of $n + m$ vertices

$$V = \{u_1, u_2, \ldots, u_n, s_1, s_2, \ldots, s_m\},$$

where each vertex $u_i$ corresponds to an element of the set $U$, and each vertex $s_i$ corresponds to the subset $S_i \in S$.

Next, place an edge between each pair of vertices $u_i, u_j, j \neq i$. This creates a complete graph on the vertex set $\{u_1, \ldots, u_n\}$. For each vertex $s_i, 1 \leq i \leq m$, add an edge between $s_i$ and vertex $u_j$ if $u_j \in S_i$ in the given instance of the Set Packing problem. This completes the construction of the graph $G$.

Suppose that the answer to the Set Packing instance is "yes." Then there exists a collection of at least $k$ subsets such that no two of the subsets share an element. Let $P = \{S_i, S_{i_2}, \ldots, S_{i_k}\}$ be the corresponding collection, where $k' \geq k$. Let $P_v = \{s_i, s_{i_2}, \ldots, s_{i_k}\} \subset V$ be the corresponding vertices in graph $G$. Then it is easy to verify that $P_v$ forms a 1-local set of size $k'; k'$ none of the vertices $\{u_1, u_2, \ldots, u_n\}$ have more than one neighbor in $P_v$ (by the definition of the edges and the fact that $P_v$ corresponds to a packing), and none of the vertices $s_i$ share any edges with nodes in the set $P_v$. Thus, if the answer to the Set Packing instance is "yes", the answer to the constructed instance of the 1-local Set Problem is "yes."

We now show the converse. Suppose the answer to the constructed instance of the 1-local Set Problem is "yes," i.e., there exists a 1-local set $P_v \subset V$ of vertices, with cardinality $k' \geq k \geq 2$. We first claim that $P_v$ cannot contain any vertices from the set $\{u_1, u_2, \ldots, u_n\}$. To see this, note that $P_v$ cannot contain all of the vertices $\{u_1, u_2, \ldots, u_n\}$, for if it did, any vertex $s_i$ that is not in $P_v$ would contain at least two neighbors in $P_v$, contradicting the fact that it is a 1-local set. Next, note that $P_v$ cannot contain more than one node from $\{u_1, u_2, \ldots, u_n\}$, for if it did, any node $u_j$ that is not in $P_v$ would have more than one neighbor in $P_v$, again contradicting the fact that it is a 1-local set. Thus suppose $P_v$ contains a single vertex from $\{u_1, \ldots, u_n\}$, and take this vertex to be $u_i$. Then each vertex $u_j (j \neq i)$ already has a neighbor in $P_v$, and thus none of the vertices $s_i, 1 \leq i \leq m$ can be in $P_v$. Thus $P_v$ is of size 1, contradicting the fact that it is a 1-local set of size at least 2.