

ON THE GEOMETRIC CONVERGENCE OF BYZANTINE-RESILIENT DISTRIBUTED OPTIMIZATION ALGORITHMS*

KANANART KUWARANANCHAROEN[†] AND SHREYAS SUNDARAM[†]

Abstract. The problem of designing distributed optimization algorithms that are resilient to Byzantine adversaries has received significant attention. For the Byzantine-resilient distributed optimization problem, the goal is to (approximately) minimize the average of the local cost functions held by the regular (nonadversarial) agents in the network. In this paper, we provide a general algorithmic framework for Byzantine-resilient distributed optimization which includes some state-of-the-art algorithms as special cases. We analyze the convergence of algorithms within the framework, and derive a geometric rate of convergence of all regular agents to a ball around the optimal solution (whose size we characterize). Furthermore, we show that approximate consensus can be achieved geometrically fast under some minimal conditions. Our analysis provides insights into the relationship among the convergence region, distance between regular agents' values, step size, and properties of the agents' functions for Byzantine-resilient distributed optimization.

Key words. consensus algorithm, convex optimization, distributed algorithms, distributed optimization, fault tolerant systems, linear convergence, multiagent systems, network security

MSC codes. 52A41, 93A14, 93A16, 90C25, 90C35, 65K05, 68M15, 68W15, 68W40

DOI. 10.1137/23M1573410

1. Introduction. Distributed optimization problems pertain to a setting where each node in a network has a local cost function, and the goal is for all agents in the network to agree on a minimizer of the average of the local cost functions. In the distributed optimization literature, there are two main paradigms: client-server and peer-to-peer. Motivated by settings where the client-server paradigm may suffer from a single point of failure or communication bottleneck, there is a growing amount of work on the peer-to-peer setting where the agents in the network are required to send and receive information only from their neighbors. A variety of algorithms have been proposed to solve such problems in peer-to-peer architectures (e.g., see [25, 5, 32, 24, 30]). The works [27, 38, 37] summarize the recent advances in the field of (peer-to-peer) distributed optimization.

These aforementioned works typically make the assumption that all agents are trustworthy and cooperative (i.e., they follow the prescribed protocol). However, it has been shown that the regular agents fail to reach an optimal solution even if a single misbehaving (or “Byzantine”) agent is present [35, 33]. Thus, designing distributed optimization algorithms that allow all the regular agents' states in the network to stay close to the minimizer of the sum of regular agents' functions regardless of the adversaries' actions has become a prevailing problem. Nevertheless, compared to the client-server setting, there have been only a few works on Byzantine-resilient algorithms in the peer-to-peer setup.

*Received by the editors May 18, 2023; accepted for publication (in revised form) October 14, 2024; published electronically January 13, 2025.

<https://doi.org/10.1137/23M1573410>

Funding: This work was supported by the National Science Foundation CAREER award 1653648.

[†]Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (kkuwaran@alumni.purdue.edu, sundara2@purdue.edu).

Contributions. In this work, we consider Byzantine-resilient (peer-to-peer) distributed deterministic optimization problems. Our contributions are as follows.

- (i) We introduce an algorithmic framework called REDGRAF, a generalization of BRIDGE in [7], which includes some state-of-the-art Byzantine-resilient distributed optimization algorithms as special cases.
- (ii) We propose a novel contraction property that provides a general method for proving geometric convergence of algorithms in REDGRAF. To the best of our knowledge, this is the first work to demonstrate a geometric rate of convergence of all regular agents' states to a ball containing the true minimizer for resilient algorithms under strong convexity. We also explicitly characterize both the convergence rate and the size of the convergence region.
- (iii) We introduce a novel mixing dynamics property used to derive approximate consensus results for algorithms in REDGRAF, explicitly characterizing both the convergence rate and final consensus diameter.
- (iv) Using our framework, we analyze the contraction and mixing dynamics properties of some state-of-the-art algorithms, leading to convergence and consensus results for each algorithm. Our work is the first to show that these algorithms satisfy such properties.
- (v) We demonstrate and compare the performance of the algorithms through numerical simulations to corroborate the theoretical results for convergence and approximate consensus.

2. Related work. The survey paper [40] provides an overview of some Byzantine-resilient algorithms for both the client-server and peer-to-peer paradigms. Since we are focusing on resilient algorithms for peer-to-peer settings, we discuss the following research papers attempting to solve such problems. Papers [33, 35, 34] show that using the *distributed gradient descent (DGD)* equipped with a *trimmed mean filter* guarantees convergence to the convex hull of the local minimizers under scalar-valued objective functions. Adopting a similar algorithm, [9] gives the same guarantee for scalar-valued problems under *deception attacks*. The work [44] also considers the scalar version of such problems but relies on *trusted agents* which cannot be compromised by adversarial attacks. To tackle vector-valued problems, [39] proposes ByRDIE, a coordinate descent method for machine learning problems leveraging the algorithm in [33], while [7] presents BRIDGE, an algorithm framework for Byzantine-resilient distributed optimization problems. Even though [39] and [7] show the convergence to the minimizer with high probability (for certain specific algorithms), they require that the training data are independent and identically distributed (i.i.d.) across the agents in the network. While resilient algorithms with the trimmed mean filter are widely used, e.g., [35, 33, 34, 9, 44, 7], the convergence analysis for multivariate functions under general assumptions is still lacking. The work [31] proposes decentralized robust subgradient push, a resilient algorithm based on a subgradient-push method [26] equipped with a *maliciousness score* for detecting adversaries. However, the work requires that the regular agents' functions have common statistical characteristics, and does not provide any guarantees on the proposed algorithm. Papers [20, 21] introduce SDMMFD and SDFD, resilient algorithms for deterministic distributed convex optimization problems with multidimensional functions. These algorithms have an asymptotic convergence guarantee to a proximity of the true minimum. However, they do not provide the convergence rate for the proposed algorithms. In contrast, the work [11] offers an algorithm with provable *exact fault tolerance*, but it relies on

redundancy among the local functions and requires the underlying communication network to be complete.

For distributed stochastic optimization problems, [29] introduces a resilient algorithm based on a total variation norm penalty motivated from [2]. The recent paper [10] also considers stochastic problems, and proposes an algorithm utilizing a distance-based filter and objective value-based filter, but does not provide any performance guarantees. The recent paper [6] which also considers stochastic problems especially for machine learning, proposes a validation-based algorithm for both i.i.d. and non-i.i.d. settings. In particular, the work theoretically shows a convergence guarantee for the proposed algorithm under convex loss functions and i.i.d. data. The recent papers [36] and [13] propose algorithms which converge to a neighborhood of a stationary point for distributed stochastic nonconvex optimization problems.

As outlined in our contributions section, this paper addresses gaps in the existing literature by demonstrating the geometric rate of convergence of all regular agents' states to a ball containing the true minimizer for a class of resilient algorithms under the strong convexity assumption. We also explicitly characterize the size of this ball. Consequently, our work provides a convergence analysis under mild assumptions for four resilient algorithms: (i) algorithms employing the trimmed mean filter (referred to as CWTM), as studied in [35, 33, 34, 9, 44, 7]; (ii) SDMMFD, as considered in [20, 21]; (iii) SDFD, as introduced in [21]; and (iv) a resilient algorithm based on resilient vector consensus (referred to as RVO) [28, 1]. For detailed descriptions of each algorithm, please refer to subsection 4.3. In this section, we provide a comparative summary of assumptions and theoretical results between our work and previous studies, as outlined in Table 1.¹

The table demonstrates that prior works on Byzantine-resilient distributed optimization mostly considered the decreasing step-size regime, typically under convex local functions, leading to sublinear convergence at best. In contrast, our work explores constant step sizes under strongly convex local functions, which enables us to achieve a linear rate of convergence for both optimality distance and consensus distance. However, this comes at the expense of obtaining an approximate consensus, rather than an exact consensus.

3. Background and problem formulation.

3.1. Background. Let \mathbb{N} , \mathbb{Z} and \mathbb{R} denote the set of natural numbers (including zero), integers, and real numbers, respectively. Let \mathbb{Z}_+ , $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{> 0}$ denote the set of positive integers, nonnegative real numbers, and positive real numbers, respectively. For convenience, for an integer $N \in \mathbb{Z}_+$, we define $[N] := \{1, 2, \dots, N\}$. The cardinality of a set is denoted by $|\cdot|$. Given positive integers $F \in \mathbb{Z}_+$ and $s \geq F$, and a set of scalars $\mathcal{X} = \{x_1, x_2, \dots, x_s\}$, define $M_F(\mathcal{X})$ and $m_F(\mathcal{X})$ to be the F th largest element and F th smallest element, respectively, of the set \mathcal{X} .

3.1.1. Linear algebra. Vectors are taken to be column vectors, unless otherwise noted. We use $x^{(\ell)}$ to represent the ℓ th component of a vector \mathbf{x} . The Euclidean norm on \mathbb{R}^d is denoted by $\|\cdot\|$. We use $\mathbf{1}$ and \mathbf{I} to denote the vector of all ones and the identity matrix, respectively, with appropriate dimensions. We denote by $\langle \mathbf{u}, \mathbf{v} \rangle$ the Euclidean inner product of vectors \mathbf{u} and \mathbf{v} , i.e., $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v}$ and by $\angle(\mathbf{u}, \mathbf{v})$ the angle between vectors \mathbf{u} and \mathbf{v} , i.e., $\angle(\mathbf{u}, \mathbf{v}) = \arccos(\frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\| \|\mathbf{v}\|})$. The Euclidean ball in \mathbb{R}^d with center at $\mathbf{x}_0 \in \mathbb{R}^d$ and radius $r \in \mathbb{R}_{\geq 0}$ is denoted by

¹We use the terms “geometric convergence” and “linear convergence” interchangeably.

TABLE 1
Comparison of Assumptions and Theoretical Results between Our Work and Previous Studies.

	Byzantine-resilient distributed optimization algorithm						
Assumptions and results	CWTM [33, 34]	CWTM [44]	CWTM [35]	CWTM [7]	SDMMFD [20, 21]	SDFD [21]	CWTM, RVO, SDMMFD, SDFD (our work)
Dimension ¹	single	single	single	multi	multi	multi	multi
Convexity	convex	convex	convex	strongly convex	convex	convex	strongly convex
Gradient	bounded and Lipschitz	bounded	bounded	bounded and Lipschitz	bounded	bounded	Lipschitz
Network	nonempty source component in reduced graphs	connected dominating set	robust graph	nonempty source component in reduced graphs	robust graph	robust graph	robust graph
Step size	decreasing	decreasing	decreasing	decreasing	decreasing	decreasing	constant
Addition	-	-	-	i.i.d. training data	-	-	-
Consensus	exact, asymptotic	exact, asymptotic	exact, sublinear rate	exact, sublinear rate	exact, asymptotic	-	approximate, linear rate
Convergence	neighborhood, asymptotic	neighborhood, asymptotic	neighborhood, asymptotic	minimum, sublinear	neighborhood, asymptotic	neighborhood, asymptotic	neighborhood, linear

¹This pertains to the suitability of each algorithm in relation to the number of independent variables within local functions.

$\mathcal{B}(\mathbf{x}_0, r) := \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x} - \mathbf{x}_0\| \leq r\}$. For $N \in \mathbb{Z}_+$, a matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is (row-)stochastic if $\mathbf{W}\mathbf{1} = \mathbf{1}$ and $w_{ij} \geq 0$ for all $i, j \in [N]$. For $N \in \mathbb{Z}_+$, we use \mathbb{S}^N to denote the set of all $N \times N$ (row-)stochastic matrices.

3.1.2. Functions properties. For a differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ and a point $\mathbf{x} \in \mathbb{R}^d$, define the vector $\nabla f(\mathbf{x}) \in \mathbb{R}^d$ to be the gradient of f at point \mathbf{x} .

DEFINITION 3.1 (strongly convex function). *Given a nonnegative real number $\mu \in \mathbb{R}_{\geq 0}$ and differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, f is μ -strongly convex if for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$,*

$$(3.1) \quad f(\mathbf{x}_1) \geq f(\mathbf{x}_2) + \langle \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle + \frac{\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|^2.$$

Note that a differentiable function is convex if it is 0-strongly convex.

DEFINITION 3.2 (Lipschitz gradient). *Given a nonnegative real number $L \in \mathbb{R}_{\geq 0}$ and differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, f has an L -Lipschitz gradient if for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$,*

$$(3.2) \quad \|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \leq L \|\mathbf{x}_1 - \mathbf{x}_2\|.$$

3.1.3. Graph theory. We denote a network by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which consists of the set of N nodes $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ and the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. If $(v_i, v_j) \in \mathcal{E}$, then node v_j can receive information from node v_i . The in-neighbor and out-neighbor sets are denoted by $\mathcal{N}_i^{\text{in}} = \{v_j \in \mathcal{V} : (v_j, v_i) \in \mathcal{E}\}$ and $\mathcal{N}_i^{\text{out}} = \{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\}$, respectively. A path from node $v_i \in \mathcal{V}$ to node $v_j \in \mathcal{V}$ is a sequence of nodes $v_{k_1}, v_{k_2}, \dots, v_{k_l}$ such that $v_{k_1} = v_i$, $v_{k_l} = v_j$ and $(v_{k_r}, v_{k_{r+1}}) \in \mathcal{E}$ for $r \in [l-1]$. Throughout the paper, the terms nodes and agents will be used interchangeably. Given a set of vectors $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, where each $\mathbf{x}_i \in \mathbb{R}^d$, we use the following shorthand notation for all $\mathcal{S} \subseteq \mathcal{V}$: $\{\mathbf{x}_i\}_{\mathcal{S}} = \{\mathbf{x}_i \in \mathbb{R}^d : v_i \in \mathcal{S}\}$.

DEFINITION 3.3. *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be rooted at $v_i \in \mathcal{V}$ if for all $v_j \in \mathcal{V} \setminus \{v_i\}$, there is a path from v_i to v_j . A graph is said to be rooted if it is rooted at some $v_i \in \mathcal{V}$.*

We will rely on the following definitions from [22].

DEFINITION 3.4 (r -reachable set). *For a given graph \mathcal{G} and a positive integer $r \in \mathbb{Z}_+$, a subset of nodes $\mathcal{S} \subseteq \mathcal{V}$ is said to be r -reachable if there exists a node $v_i \in \mathcal{S}$ such that $|\mathcal{N}_i^{\text{in}} \setminus \mathcal{S}| \geq r$.*

DEFINITION 3.5 (r -robust graphs). *For a positive integer $r \in \mathbb{Z}_+$, a graph \mathcal{G} is said to be r -robust if for all pairs of disjoint nonempty subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$, at least one of \mathcal{S}_1 or \mathcal{S}_2 is r -reachable.*

The above definitions capture the idea that sets of nodes should contain individual nodes that have a sufficient number of neighbors outside that set. This will be important for the *local* decisions made by each node in resilient distributed algorithms, and will allow information from the rest of the network to penetrate into different sets of nodes.

Next, following [3], we define the composition of two graphs and conditions on a sequence of graphs which will be useful for stating a mild condition for achieving approximate consensus guarantees later as follows.

DEFINITION 3.6 (composition). *The composition of a directed graph $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E}_1)$ with a directed graph $\mathcal{G}_2 = (\mathcal{V}, \mathcal{E}_2)$ written as $\mathcal{G}_2 \circ \mathcal{G}_1$, is the directed graph $(\mathcal{V}, \mathcal{E})$ with $(v_i, v_j) \in \mathcal{E}$ if there is a $v_k \in \mathcal{V}$ such that $(v_i, v_k) \in \mathcal{E}_1$ and $(v_k, v_j) \in \mathcal{E}_2$.*

DEFINITION 3.7 (jointly rooted). *A finite sequence of directed graphs $\{\mathcal{G}_k\}_{k \in [K]}$ is jointly rooted if the composition $\mathcal{G}_K \circ \mathcal{G}_{K-1} \circ \cdots \circ \mathcal{G}_1$ is rooted.*

DEFINITION 3.8 (repeatedly jointly rooted). *An infinite sequence of graphs $\{\mathcal{G}_k\}_{k \in \mathbb{Z}_+}$ is repeatedly jointly rooted if there is a positive integer $q \in \mathbb{Z}_+$ for which each finite sequence $\mathcal{G}_{q(k-1)+1}, \dots, \mathcal{G}_{qk}$ is jointly rooted for all $k \in \mathbb{Z}_+$.*

For a stochastic matrix $S \in \mathbb{S}^N$, let $\mathbb{G}(S)$ denote the graph \mathcal{G} whose adjacency matrix is the transpose of the matrix obtained by replacing all of S 's nonzero entries with 1's.

3.1.4. Adversarial behavior.

DEFINITION 3.9. *A node $v_i \in \mathcal{V}$ is said to be Byzantine if during each iteration of the prescribed algorithm, it is capable of sending arbitrary values to different neighbors. It is also allowed to update its local information arbitrarily at each iteration of any prescribed algorithm.*

The set of Byzantine agents is denoted by $\mathcal{V}_B \subset \mathcal{V}$. The set of regular agents is denoted by $\mathcal{V}_R = \mathcal{V} \setminus \mathcal{V}_B$. The identities of the Byzantine agents are unknown to the regular agents in advance. Furthermore, we allow the Byzantine agents to know the entire topology of the network, functions equipped by the regular nodes, and the deployed algorithm. In addition, Byzantine agents are allowed to coordinate with other Byzantine agents and access the current and previous information contained by the nodes in the network (e.g., current and previous states of all nodes). Such extreme behavior is typical in the field of distributed computing [23] and in adversarial distributed optimization [33, 35, 41, 39, 8]. In exchange for allowing such extreme behavior, we will consider a limitation on the number of such adversaries in the neighborhood of each regular node, as follows.

DEFINITION 3.10 (F -local model). *For a positive integer $F \in \mathbb{Z}_+$, we say that the set of adversaries \mathcal{V}_B is an F -local set if $|\mathcal{N}_i^{\text{in}} \cap \mathcal{V}_B| \leq F$ for all $v_i \in \mathcal{V}_R$.*

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

3.2. Problem formulation. Consider a group of N agents \mathcal{V} interconnected over a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each agent $v_i \in \mathcal{V}$ has a local cost function $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$. Since Byzantine nodes are allowed to send arbitrary values to their neighbors at each iteration of any algorithm, it is not possible to minimize the quantity $\frac{1}{N} \sum_{v_i \in \mathcal{V}} f_i(\mathbf{x})$ that is typically sought in distributed optimization (since one is not guaranteed to infer any information about the true functions of the Byzantine agents) [35, 33]. Thus, we restrict the summation only to the regular agents' functions, i.e., the objective is to solve the following optimization problem,

$$(3.3) \quad \min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}), \quad \text{where} \quad f(\mathbf{x}) := \frac{1}{|\mathcal{V}_R|} \sum_{v_i \in \mathcal{V}_R} f_i(\mathbf{x}),$$

where \mathcal{V}_R represents the set of regular agents, and $f_i(\mathbf{x})$ denotes the objective function associated with agent v_i .

A key challenge in solving the above problem is that no regular agent is aware of the identities or actions of the Byzantine agents. In particular, solving (3.3) exactly is not possible under Byzantine behavior, since the identities and local functions of the Byzantine nodes are not known (the Byzantine agents can simply change their local functions and pretend to be a regular agent in the algorithms and these can never be

detected). Therefore, one must settle for computing an approximate solution to (3.3) (see [35, 33] for a more detailed discussion of this fundamental limitation).

Establishing the convergence (especially obtaining the rate of convergence) for resilient distributed optimization algorithms under general assumptions on the local functions (i.e., not assuming i.i.d. or redundancy) is nontrivial as evidenced by the lack of such results in the literature. We close this gap by introducing a *proper* intermediate step which is showing the states contraction property (Definition 5.4) before proceeding to show the convergence (Proposition 5.7 and Theorem 5.8). Importantly, the contraction property not only captures some of state-of-the-art resilient distributed optimization algorithms in the literature (Theorem 5.15) but also facilitates the (geometric) convergence analysis.

4. Resilient distributed optimization algorithms.

4.1. Our framework. In this subsection, we introduce a class of resilient distributed optimization algorithms represented by the *resilient distributed gradient-descent algorithmic framework* (REDGRAF) shown in Algorithmic Framework 4.1. At each time step $k \in \mathbb{N}$, each regular agent² $v_i \in \mathcal{V}_{\mathcal{R}}$ maintains and updates a state vector $\mathbf{x}_i[k] \in \mathbb{R}^d$, which is its estimate of the solution to problem (3.3), and optionally an auxiliary vector $\mathbf{y}_i[k] \in \mathbb{R}^{d'}$ where the dimension $d' \in \mathbb{N}$ depends on the specific algorithm. In our algorithmic framework, we let $\mathbf{z}_i[k] = [\mathbf{x}_i^T[k], \mathbf{y}_i^T[k]]^T \in \mathbb{R}^{d+d'}$ and, similarly, $\tilde{\mathbf{z}}_i[k] = [\tilde{\mathbf{x}}_i^T[k], \tilde{\mathbf{y}}_i^T[k]]^T \in \mathbb{R}^{d+d'}$. In fact, REDGRAF is a generalization of BRIDGE proposed in [7] in the sense that our framework allows the state vector $\mathbf{z}_i[k]$ to include the auxiliary vector $\mathbf{y}_i[k]$. In Algorithmic Framework 4.1, the operation

Algorithmic Framework 4.1. Resilient Distributed Gradient-Descent Algorithmic Framework (REDGRAF).

Input: Network \mathcal{G} , functions $\{f_i\}_{v_i \in \mathcal{V}_{\mathcal{R}}}$, parameter F

- 1: **Step I:** Initialization
Each $v_i \in \mathcal{V}_{\mathcal{R}}$ sets $\mathbf{z}_i[0] \leftarrow \text{init}(f_i)$
- 2: **for** $k = 0, 1, 2, 3, \dots$ **do**
- 3: **for** $v_i \in \mathcal{V}_{\mathcal{R}}$ **do**
- 4: **Step II:** Broadcast and Receive
 v_i broadcasts $\mathbf{z}_i[k]$ to $\mathcal{N}_i^{\text{out}}$ and receives $\mathbf{z}_j[k]$ from $v_j \in \mathcal{N}_i^{\text{in}}$. Let $\mathcal{Z}_i[k] = \{\mathbf{z}_j[k] : v_j \in \mathcal{N}_i^{\text{in}} \cup \{v_i\}\}$
- 5: **Step III:** Filtering Step
 $\tilde{\mathbf{z}}_i[k] \leftarrow \text{filt}(\mathcal{Z}_i[k], F)$ \triangleright Note: $\tilde{\mathbf{z}}_i[k] = [\tilde{\mathbf{x}}_i^T[k], \tilde{\mathbf{y}}_i^T[k]]^T$
- 6: **Step IV:** Gradient Update

$$(4.1) \quad \begin{aligned} \mathbf{x}_i[k+1] &= \tilde{\mathbf{x}}_i[k] - \alpha_k \nabla f_i(\tilde{\mathbf{x}}_i[k]), \\ \mathbf{y}_i[k+1] &= \tilde{\mathbf{y}}_i[k] \quad (\text{if exists}), \end{aligned}$$

where $\alpha_k \in \mathbb{R}_{>0}$ is the step size

- 7: **end for**
 - 8: $\mathbf{z}_i[k+1] = [\mathbf{x}_i^T[k+1], \mathbf{y}_i^T[k+1]]^T$ for $v_i \in \mathcal{V}_{\mathcal{R}}$
 - 9: **end for**
-

²Byzantine agents do not necessarily need to follow the above algorithm, and can update their states, however they wish.

$\text{init}(f_i)$ initializes $\mathbf{z}_i[0] = [\mathbf{x}_i^T[0], \mathbf{y}_i^T[0]]^T$, and the operation $\text{filt}(\mathcal{Z}_i[k], F)$ performs a filtering procedure (to remove potentially adversarial states received from neighbors) and returns a vector $\tilde{\mathbf{z}}_i[k]$. These functions will vary across algorithms, and will be discussed for specific algorithms later.

4.2. Definition of some standard operations for resilient distributed optimization. To show that our framework (REDGRAF) captures several existing resilient distributed optimization algorithms as special cases, we first define some operations that are used by existing algorithms. Throughout, let $\mathcal{V}_i[k] \subseteq \mathcal{N}_i^{\text{in}} \cup \{v_i\}$, $\mathcal{X}_i[k] = \{\mathbf{x}_j[k]\}_{\mathcal{N}_i^{\text{in}} \cup \{v_i\}}$ and $\mathcal{Y}_i[k] = \{\mathbf{y}_j[k]\}_{\mathcal{N}_i^{\text{in}} \cup \{v_i\}}$.

- $\tilde{\mathcal{V}}_i[k] \leftarrow \text{dist.filt}(\mathcal{V}_i[k], \mathcal{Z}_i[k], F)$:
Regular agent $v_i \in \mathcal{V}_R$ removes F states that are far away from $\mathbf{y}_i[k]$. More specifically, an agent $v_j \in \mathcal{V}_i[k]$ is in $\tilde{\mathcal{V}}_i[k]$ if and only if

$$\|\mathbf{x}_j[k] - \mathbf{y}_i[k]\| \leq \max\{q_M, \|\mathbf{x}_i[k] - \mathbf{y}_i[k]\|\},$$

where $q_M = M_F(\{\|\mathbf{x}_s[k] - \mathbf{y}_i[k]\|\}_{v_s \in \mathcal{V}_i[k]})$.

- $\tilde{\mathcal{V}}_i[k] \leftarrow \text{full.mm.filt}(\mathcal{V}_i[k], \mathcal{X}_i[k], F)$:
Regular agent $v_i \in \mathcal{V}_R$ removes states that have extreme values in any of their components. For a given $k \in \mathbb{N}$ and $\ell \in [d]$, let $q_m^{(\ell)} = m_F(\{x_s^{(\ell)}[k]\}_{v_i[k]})$ and $q_M^{(\ell)} = M_F(\{x_s^{(\ell)}[k]\}_{v_i[k]})$. An agent $v_j \in \mathcal{V}_i[k]$ is in $\tilde{\mathcal{V}}_i[k]$ if and only if for all $\ell \in [d]$,

$$\min\{q_m^{(\ell)}, x_i^{(\ell)}[k]\} \leq x_j^{(\ell)}[k] \leq \max\{q_M^{(\ell)}, x_i^{(\ell)}[k]\}.$$

- $\{\tilde{\mathcal{V}}_i^{(\ell)}[k]\}_{\ell \in [d]} \leftarrow \text{cw.mm.filt}(\mathcal{V}_i[k], \mathcal{X}_i[k], F)$:
For each dimension $\ell \in [d]$, regular agent $v_i \in \mathcal{V}_R$ removes the F highest and F lowest values of the states of agents in $\mathcal{V}_i[k]$ along that dimension. More specifically, for a given $k \in \mathbb{N}$ and $\ell \in [d]$, let $q_m^{(\ell)} = m_F(\{x_s^{(\ell)}[k]\}_{v_i[k]})$ and $q_M^{(\ell)} = M_F(\{x_s^{(\ell)}[k]\}_{v_i[k]})$. An agent $v_j \in \mathcal{V}_i[k]$ is in $\tilde{\mathcal{V}}_i^{(\ell)}[k]$ if and only if

$$\min\{q_m^{(\ell)}, x_i^{(\ell)}[k]\} \leq x_j^{(\ell)}[k] \leq \max\{q_M^{(\ell)}, x_i^{(\ell)}[k]\}.$$

- $\tilde{\mathbf{x}}_i[k] \leftarrow \text{full.average}(\mathcal{V}_i[k], \mathcal{X}_i[k])$:
Regular agent $v_i \in \mathcal{V}_R$ computes

$$(4.2) \quad \tilde{\mathbf{x}}_i[k] = \sum_{v_j \in \mathcal{V}_i[k]} w_{ij}[k] \mathbf{x}_j[k],$$

where $\sum_{v_j \in \mathcal{V}_i[k]} w_{ij}[k] = 1$ and $w_{ij}[k] \in \mathbb{R}_{>0}$ for all $v_j \in \mathcal{V}_i[k]$.

- $\tilde{\mathbf{x}}_i[k] \leftarrow \text{cw.average}(\{\mathcal{V}_i^{(\ell)}[k]\}_{\ell \in [d]}, \mathcal{X}_i[k])$:
For each dimension $\ell \in [d]$, regular agent $v_i \in \mathcal{V}_R$ computes

$$(4.3) \quad \tilde{\mathbf{x}}_i^{(\ell)}[k] = \sum_{v_j \in \mathcal{V}_i^{(\ell)}[k]} w_{ij}^{(\ell)}[k] x_j^{(\ell)}[k],$$

where $w_{ij}^{(\ell)}[k] \in \mathbb{R}_{>0}$ for all $v_j \in \mathcal{V}_i^{(\ell)}[k]$ and $\sum_{v_j \in \mathcal{V}_i^{(\ell)}[k]} w_{ij}^{(\ell)}[k] = 1$.

- $\tilde{\mathbf{x}}_i[k] \leftarrow \text{safe.point}(\mathcal{V}_i[k], \mathcal{X}_i[k], F)$:
Regular agent $v_i \in \mathcal{V}_R$ returns a state $\tilde{\mathbf{x}}_i[k]$ which can be written as

$$(4.4) \quad \tilde{\mathbf{x}}_i[k] = \sum_{v_j \in \mathcal{V}_i[k] \cap \mathcal{V}_R} w_{ij}[k] \mathbf{x}_j[k],$$

where $w_{ij}[k] \in \mathbb{R}_{>0}$ for all $v_j \in \mathcal{V}_i[k] \cap \mathcal{V}_R$ and $\sum_{v_j \in \mathcal{V}_i[k] \cap \mathcal{V}_R} w_{ij}[k] = 1$. The works [28, 1] discuss methods used to compute $\tilde{\mathbf{x}}_i[k]$.

4.3. Mapping existing algorithms into REDGRAF. Using the operations defined above, we now discuss some algorithms in the literature that fall into our algorithmic framework.

Simultaneous distance-minmax filtering dynamics (SDMMFD) [20, 21] and *simultaneous distance filtering dynamics (SDFD)* [21]: these two algorithms are captured in our framework by defining $\mathbf{z}_i[k] = [\mathbf{x}_i^T[k], \mathbf{y}_i^T[k]]^T$ where $\mathbf{y}_i[k] \in \mathbb{R}^d$. In the initialization step $\mathbf{z}_i[0] \leftarrow \text{init}(f_i)$ (line 1) of both algorithms, each regular agent $v_i \in \mathcal{V}_{\mathcal{R}}$ computes an approximate minimizer $\hat{\mathbf{x}}_i^* \in \mathbb{R}^d$ of its local function f_i (using any appropriate optimization algorithm) and then sets $\mathbf{x}_i[0] \in \mathbb{R}^d$ arbitrarily and $\mathbf{y}_i[0] = \hat{\mathbf{x}}_i^*$. In the filtering step $\tilde{\mathbf{z}}_i[k] \leftarrow \text{filt}(\mathcal{Z}_i[k], F)$ (line 5), SDMMFD executes the following sequence of methods:

1. $\mathcal{V}_i^{\text{dist}}[k] \leftarrow \text{dist_filt}(\mathcal{N}_i^{\text{in}} \cup \{v_i\}, \mathcal{Z}_i[k], F)$,
2. $\mathcal{V}_i^{\text{x,mm}}[k] \leftarrow \text{full_mm_filt}(\mathcal{V}_i^{\text{dist}}[k], \mathcal{X}_i[k], F)$,
3. $\tilde{\mathbf{x}}_i[k] \leftarrow \text{full_average}(\mathcal{V}_i^{\text{x,mm}}[k], \mathcal{X}_i[k])$,
4. $\{\tilde{\mathbf{y}}_i^{(\ell)}[k]\}_{\ell \in [d]} \leftarrow \text{cw_mm_filt}(\mathcal{N}_i^{\text{in}} \cup \{v_i\}, \mathcal{Y}_i[k], F)$,
5. $\tilde{\mathbf{y}}_i[k] \leftarrow \text{cw_average}(\{\tilde{\mathbf{y}}_i^{(\ell)}[k]\}_{\ell \in [d]}, \mathcal{Y}_i[k])$.

The first three steps compute the intermediate main state $\tilde{\mathbf{x}}_i[k]$ while the last two steps compute the intermediate auxiliary vector $\tilde{\mathbf{y}}_i[k]$. On the other hand, SDFD executes the same sequence of methods except that step (ii) is removed and $\mathcal{V}_i^{\text{x,mm}}[k]$ in step (iii) is replaced by $\mathcal{V}_i^{\text{dist}}[k]$. Then, for both algorithms, we set $\tilde{\mathbf{z}}_i[k] = [\tilde{\mathbf{x}}_i^T[k], \tilde{\mathbf{y}}_i^T[k]]^T$.

Coordinatewise trimmed mean (CWTM) [35, 33, 34, 9, 44, 7] and *resilient vector optimization (RVO)* based on *resilient vector consensus* [28, 1]: these algorithms are captured by setting $\mathbf{z}_i[k] = \mathbf{x}_i[k]$ (i.e., $\mathbf{y}_i[k] = \emptyset$). In the initialization step $\mathbf{z}_i[0] \leftarrow \text{init}(f_i)$ (line 1) of both algorithms, the regular agents $v_i \in \mathcal{V}_{\mathcal{R}}$ arbitrarily initialize $\mathbf{x}_i[0] \in \mathbb{R}^d$. In the filtering step $\tilde{\mathbf{z}}_i[k] \leftarrow \text{filt}(\mathcal{Z}_i[k], F)$ (line 5), CWTM executes the following sequence of methods:

1. $\{\tilde{\mathbf{y}}_i^{(\ell)}[k]\}_{\ell \in [d]} \leftarrow \text{cw_mm_filt}(\mathcal{N}_i^{\text{in}} \cup \{v_i\}, \mathcal{X}_i[k], F)$,
2. $\tilde{\mathbf{x}}_i[k] \leftarrow \text{cw_average}(\{\tilde{\mathbf{y}}_i^{(\ell)}[k]\}_{\ell \in [d]}, \mathcal{X}_i[k])$,

whereas RVO executes

1. $\tilde{\mathbf{x}}_i[k] \leftarrow \text{safe_point}(\mathcal{N}_i^{\text{in}} \cup \{v_i\}, \mathcal{X}_i[k], F)$.

In fact, the algorithms proposed in [2, 29, 10, 6] also fall into our framework. However, in this work, we focus on analyzing the four algorithms above since they share some common property (stated formally in Theorem 5.15), and we will provide a discussion on the algorithms in the works [2, 29, 10, 6] in Remark 5.17.

5. Assumptions and main results. We now turn to stating assumptions and definitions in subsection 5.1 which will be used to prove convergence properties in subsection 5.2 and consensus properties in subsection 5.3. Finally, in subsection 5.4, we analyze certain properties of each algorithm mentioned in the previous section.

5.1. Assumptions and definitions.

Assumption 5.1. For all $v_i \in \mathcal{V}$, given positive numbers $\mu_i \in \mathbb{R}_{>0}$ and $L_i \in \mathbb{R}_{>0}$, the functions f_i are μ_i -strongly convex and differentiable. Furthermore, the functions f_i have L_i -Lipschitz continuous gradients.

The strongly convex and Lipschitz continuous gradient assumptions given above are common in the distributed convex optimization literature [42, 24, 38, 14, 15]. We define $\tilde{L} := \max_{v_i \in \mathcal{V}_{\mathcal{R}}} L_i$ and $\tilde{\mu} := \min_{v_i \in \mathcal{V}_{\mathcal{R}}} \mu_i$. Since $\{f_i\}_{v_i \in \mathcal{V}_{\mathcal{R}}}$ are strongly convex functions, let $\mathbf{x}_i^* \in \mathbb{R}^d$ be the minimizer of the function f_i , i.e., $f_i(\mathbf{x}_i^*) = \min_{\mathbf{x} \in \mathbb{R}^d} f_i(\mathbf{x})$. Moreover, let $\mathbf{c}^* \in \mathbb{R}^d$ and $r^* \in \mathbb{R}_{\geq 0}$ be such that $\mathbf{x}_i^* \in \mathcal{B}(\mathbf{c}^*, r^*)$ for all $v_i \in \mathcal{V}_{\mathcal{R}}$. Both

quantities \mathbf{c}^* and r^* exist due to the existence of the set $\{\mathbf{x}_i^*\}_{i \in \mathcal{V}_{\mathcal{R}}} \subset \mathbb{R}^d$. Let $\mathbf{x}^* \in \mathbb{R}^d$ be the minimizer of the function $f(\mathbf{x})$, i.e., the solution of problem (3.3). In other words, $f(\mathbf{x}^*) = \min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$. For convenience, we also denote $f^* := f(\mathbf{x}^*)$ and $\mathbf{g}_i[k] := \nabla f_i(\tilde{\mathbf{x}}_i[k])$.

Assumption 5.2. Given a positive integer $F \in \mathbb{Z}_+$, the Byzantine agents form an F -local set.

Assumption 5.3. There exists a positive number $\omega \in \mathbb{R}_{>0}$ such that for all $k \in \mathbb{N}$ and $\ell \in [d]$, the nonzero weights $w_{ij}[k]$ in `full_average` and `safe_point`, and $w_{ij}^{(\ell)}[k]$ in `cw_average` (all defined in subsection 4.2) are lower bounded by ω .

Now, we introduce certain properties related to algorithms within our algorithmic framework (Algorithmic Framework 4.1). These definitions are crucial for proving the convergence results in subsection 5.2 and the consensus results in subsection 5.3. Specifically, the following definition captures a certain behavior of the aggregation and filtering mechanism in Step III of the framework.

DEFINITION 5.4. For a vector $\mathbf{x}_c \in \mathbb{R}^d$, constant $\gamma \in \mathbb{R}_{\geq 0}$, and sequence $\{c[k]\}_{k \in \mathbb{N}} \subset \mathbb{R}$, a resilient distributed optimization algorithm A in REDGRAF is said to satisfy the $(\mathbf{x}_c, \gamma, \{c[k]\})$ -states contraction property if it holds that $\lim_{k \rightarrow \infty} c[k] = 0$ and for all $k \in \mathbb{N}$ and $v_i \in \mathcal{V}_{\mathcal{R}}$,

$$(5.1) \quad \|\tilde{\mathbf{x}}_i[k] - \mathbf{x}_c\| \leq \sqrt{\gamma} \max_{v_j \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_j[k] - \mathbf{x}_c\| + c[k].$$

In the above definition, the vector \mathbf{x}_c is called the *contraction center* and the constant γ is called the *contraction factor*. In general, we want the contraction factor γ to be small so that the intermediate state $\tilde{\mathbf{x}}_i[k]$ remains close to the center \mathbf{x}_c . The sequence $\{c[k]\}_{k \in \mathbb{N}}$ captures a perturbation to the contraction term in each time step. For a given vector $\mathbf{x}_c \in \mathbb{R}^d$, we define

$$(5.2) \quad r_c := \max_{v_i \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_c - \mathbf{x}_i^*\|.$$

Next, we introduce a property, concerning algorithms in Algorithmic Framework 4.1, that builds on the states contraction property by further specifying a range for some of the parameters.

DEFINITION 5.5. Suppose Assumption 5.1 holds, and a resilient distributed optimization algorithm A in REDGRAF satisfies the $(\mathbf{x}_c, \gamma, \{c[k]\})$ -states contraction property (for some $\mathbf{x}_c \in \mathbb{R}^d$, $\gamma \in \mathbb{R}_{\geq 0}$, and $\{c[k]\}_{k \in \mathbb{N}} \subset \mathbb{R}$). Algorithm A is said to satisfy the reduction property of

- Type-I if $\gamma \in [0, 1)$ and $\alpha_k = \alpha \in (0, \frac{1}{L}]$;
- Type-II if $\gamma \in [1, \frac{1}{1-\frac{\mu}{L}})$ and $\alpha_k = \alpha \in (\frac{1}{\mu}(1 - \frac{1}{\gamma}), \frac{1}{L}]$.

This reduction property, in particular, will be used to specify the conditions under which an algorithm converges.

Let $\mathcal{V}_{\mathcal{R}} = \{v_{i_1}, v_{i_2}, \dots, v_{i_{|\mathcal{V}_{\mathcal{R}}|}}\}$ denote the set of all regular agents. For a set of vectors $\{\mathbf{u}_i\}_{i \in \mathcal{V}} \subset \mathbb{R}^d$ and $\ell \in [d]$, we denote $\mathbf{u}^{(\ell)} = [u_{i_1}^{(\ell)}, u_{i_2}^{(\ell)}, \dots, u_{i_{|\mathcal{V}_{\mathcal{R}}|}}^{(\ell)}]^T \in \mathbb{R}^{|\mathcal{V}_{\mathcal{R}}|}$, the vector containing the ℓ th dimension of each vector \mathbf{u}_i corresponding to the regular agents' indices. The following definition characterizes the dynamics of all the regular agents in the network which will be a crucial ingredient in showing the approximate consensus result in subsection 5.3.

DEFINITION 5.6. For a set of sequences of matrices $\{\mathbf{W}^{(\ell)}[k]\}_{k \in \mathbb{N}, \ell \in [d]} \subset \mathbb{S}^{|\mathcal{V}_{\mathcal{R}}|}$ and constant $G \in \mathbb{R}_{\geq 0}$, a resilient distributed optimization algorithm A in REDGRAF is said to possess $(\{\mathbf{W}^{(\ell)}[k]\}, G)$ -mixing dynamics if the state dynamics can be written as

$$(5.3) \quad \mathbf{x}^{(\ell)}[k+1] = \mathbf{W}^{(\ell)}[k]\mathbf{x}^{(\ell)}[k] - \alpha_k \mathbf{g}^{(\ell)}[k]$$

for all $k \in \mathbb{N}$ and $\ell \in [d]$, the sequences of graphs $\{\mathbb{G}(\mathbf{W}^{(\ell)}[k])\}_{k \in \mathbb{N}}$ are repeatedly jointly rooted for all $\ell \in [d]$, and $\limsup_k \|\mathbf{g}_i[k]\|_{\infty} \leq G$ for all $v_i \in \mathcal{V}_{\mathcal{R}}$.

The matrix $\mathbf{W}^{(\ell)}[k]$ is called a *mixing matrix* which directly affects the ability of the nodes to reach consensus [3] while the constant G quantifies an upper bound on the perturbation (i.e., the scaled gradient $\alpha_k \mathbf{g}^{(\ell)}[k]$) to the consensus process.

Later in subsection 5.4, particularly in Theorem 5.15, we will formally discuss the $(\mathbf{x}_c, \gamma, \{c[k]\})$ -state contraction and the $(\{\mathbf{W}^{(\ell)}[k]\}, G)$ -mixing dynamics properties, along with the parameter values for each algorithm considered in subsection 4.3.

5.2. The region to which the states converge. In this subsection, we derive a convergence result for some particular algorithms in REDGRAF (Theorem 5.8). We start by establishing convergence to a neighborhood around the center point \mathbf{x}_c (Proposition 5.7). Following this, we present the main convergence theorem (Theorem 5.8), leveraging the fact that the minimizer \mathbf{x}^* , the solution to problem (3.3), resides within this neighborhood (Lemma A.5 in Appendix A.4).

For convenience, if Assumption 5.1 holds and the step size $\alpha_k = \alpha$ for all $k \in \mathbb{N}$, we define

$$(5.4) \quad \beta := \sqrt{1 - \alpha \tilde{\mu}}.$$

We now present an intermediate result, demonstrating that the states of all the regular agents will converge to a ball for all algorithms in REDGRAF that satisfy the reduction property (Definition 5.5). The proof is provided in Appendix A.3.

PROPOSITION 5.7. Suppose Assumption 5.1 holds. If an algorithm A satisfies the reduction property of Type-I or Type-II, then for all $v_i \in \mathcal{V}_{\mathcal{R}}$, it holds that

$$(5.5) \quad \limsup_k \|\mathbf{x}_i[k] - \mathbf{x}_c\| \leq \frac{r_c \sqrt{\alpha \tilde{L}}}{1 - \beta \sqrt{\gamma}} := R^*,$$

where \mathbf{x}_c , r_c and β are defined in Definition 5.4, (5.2), and (5.4), respectively. Furthermore, if $c[k] = \mathcal{O}(\xi^k)$ and $\xi \in (0, 1) \setminus \{\beta \sqrt{\gamma}\}$, then

$$(5.6) \quad \|\mathbf{x}_i[k] - \mathbf{x}_c\| \leq R^* + \mathcal{O}((\max\{\beta \sqrt{\gamma}, \xi\})^k) \quad \text{for all } v_i \in \mathcal{V}_{\mathcal{R}}.$$

We refer to R^* in (5.5) as the *convergence radius*, and we denote $\mathcal{B}(\mathbf{x}_c, R^*)$ as the *convergence region*. Additionally, the term $\beta \sqrt{\gamma}$ in (5.6) is referred to as the *convergence rate*.³ In particular, the convergence region is the ball which has the center at \mathbf{x}_c and the radius R^* depending on the functions' parameters μ_i and L_i , the contraction factor γ , the constant step size α , and the constant capturing the position of the contraction center r_c (defined in (5.2)). We emphasize that the convergence region does not depend on the contraction perturbation sequence $\{c[k]\}_{k \in \mathbb{N}}$ as long as the sequence converges to zero.

³In this context, a smaller convergence rate indicates faster convergence.

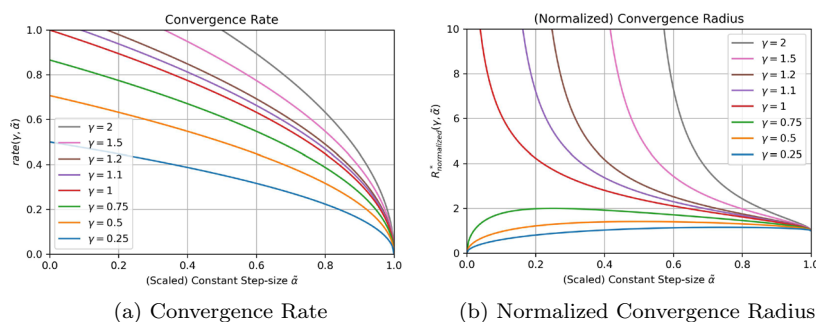


FIG. 1. The convergence rate and the normalized convergence radius for different values of the contraction factor γ and for legitimate values of the scaled constant step size $\tilde{\alpha}$.

To analyze the expression of R^* and the convergence rate $\beta\sqrt{\gamma}$, we simplify the expression as follows. Let $\kappa = \frac{\bar{L}}{\mu}$, $\tilde{\alpha} = \alpha\tilde{\mu}$, and $\text{dom}_{R^*} = \{(\gamma, \tilde{\alpha}) \in \mathbb{R}^2 : \gamma \in [0, \infty) \text{ and } \tilde{\alpha} \in (\max\{0, 1 - \frac{1}{\gamma}\}, 1]\}$. Then for R^* , we normalize the expression by $r_c\sqrt{\kappa}$. As a result, we have the convergence rate, $\text{rate} : \text{dom}_{R^*} \rightarrow [0, 1)$, and (normalized) convergence radius, $R^*_{\text{normalized}} : \text{dom}_{R^*} \rightarrow \mathbb{R}_{\geq 0}$, as

$$(5.7) \quad \text{rate}(\gamma, \tilde{\alpha}) = \sqrt{\gamma}\sqrt{1 - \tilde{\alpha}} \quad \text{and} \quad R^*_{\text{normalized}}(\gamma, \tilde{\alpha}) = \frac{\sqrt{\tilde{\alpha}}}{1 - \sqrt{\gamma}\sqrt{1 - \tilde{\alpha}}},$$

respectively. Note that the variable κ can be interpreted as an upper bound on the condition numbers of $\{f_i\}_{i \in \mathcal{V}_R}$ [12]. The plots regarding the convergence rate (rate) and normalized convergence radius ($R^*_{\text{normalized}}$) with respect to the scaled constant step size ($\tilde{\alpha}$) for some values of γ are given in Figures 1(a) and 1(b), respectively.

From (5.6), we can conclude that the states of the regular agents converge *geometrically* to the convergence region $\mathcal{B}(\mathbf{x}_c, R^*)$. Furthermore, it is evident from the inequality (5.6) and Figure 1(a) that as the constant step size α increases, the convergence rate decreases (i.e., the states of regular agents converge faster).

Considering the expression of the (normalized) convergence radius $R^*_{\text{normalized}}$ in (5.7), it should be noted that R^* is strictly increasing with respect to γ . In addition, applying [18, Lemma A.4] and noting that $R^*_{\text{normalized}}$ is a continuous function with respect to $\tilde{\alpha}$, we can conclude as follows.

- For $\gamma \in [0, 1)$, a small constant step size α would yield a small convergence radius R^* (since $R^*|_{\tilde{\alpha}=0} = 0$ and $R^*|_{\tilde{\alpha}=1} = r_c\sqrt{\kappa}$). Furthermore, we have $R^* \leq R^*|_{\tilde{\alpha}=1-\gamma} = \frac{r_c\sqrt{\kappa}}{\sqrt{1-\gamma}}$ for all valid values of α .
- For $\gamma \in [1, \infty)$, the optimal convergence radius is obtained by choosing $\alpha = \frac{1}{L}$ (due to the condition on α in Proposition 5.7) and the corresponding radius is $R^*|_{\tilde{\alpha}=\frac{1}{\kappa}} = \frac{r_c}{1 - \sqrt{\gamma}\sqrt{1 - \frac{1}{\kappa}}}$.

Additionally, the explicit characterization of R^* in (5.5) also allows us to analyze its behavior with respect to the (scaled) constant step size $\tilde{\alpha}$ when $\tilde{\alpha}$ is closed to the respective lower bound.

- When $\gamma \in [0, 1)$ and $\tilde{\alpha}$ approaches 0^+ , we have that $R^* \approx \frac{r_c\sqrt{\kappa}}{1 - \sqrt{\gamma}} \cdot \tilde{\alpha}^{\frac{1}{2}}$.
- When $\gamma = 1$ and $\tilde{\alpha}$ approaches 0^+ , we have that $R^* \approx 2r_c\sqrt{\kappa} \cdot \tilde{\alpha}^{-\frac{1}{2}}$.
- When $\gamma \in (1, \infty)$ and $\tilde{\alpha}$ approaches $(1 - \frac{1}{\gamma})^+$, we have that $R^* \approx \frac{2r_c\sqrt{\kappa}}{\gamma} \cdot \sqrt{1 - \frac{1}{\gamma}} \cdot \hat{\alpha}^{-1}$, where $\hat{\alpha} = \tilde{\alpha} - (1 - \frac{1}{\gamma})$.

Next, recall that $\mathbf{x}^* \in \mathbb{R}^d$ is the minimizer of the function $\frac{1}{|\mathcal{V}_{\mathcal{R}}|} \sum_{v_i \in \mathcal{V}_{\mathcal{R}}} f_i(\mathbf{x})$, which is our objective function (problem (3.3)). Lemma A.5, which is formally stated and proved in Appendix A.4, informs us that the true minimizer \mathbf{x}^* is within the convergence region $\mathcal{B}(\mathbf{x}_c, R^*)$, provided a certain condition on γ and α holds. This condition, in fact, aligns with the reduction property of Type-II (see Definition 5.5). Consequently, the geometric convergence of regular agents to a neighborhood of the true minimizer \mathbf{x}^* , as shown in Theorem 5.8, follows directly from applying Lemma A.5 to Proposition 5.7. However, it is important to note that determining the true minimizer \mathbf{x}^* exactly is impossible in the presence of Byzantine agents.

THEOREM 5.8 (convergence). *Suppose Assumption 5.1 holds and an algorithm A satisfies the reduction property of Type-II. If the perturbation sequence $c[k] = \mathcal{O}(\xi^k)$, where $\xi \in (0, 1) \setminus \{\beta\sqrt{\gamma}\}$, then it holds that*

$$\|\mathbf{x}_i[k] - \mathbf{x}^*\| \leq 2R^* + \mathcal{O}((\max\{\beta\sqrt{\gamma}, \xi\})^k) \quad \text{for all } v_i \in \mathcal{V}_{\mathcal{R}},$$

where the convergence radius R^* is defined in (5.5).

Before comparing our obtained convergence rate with those in the literature, let us first assume that the f_i are μ -strongly convex and have an L -Lipschitz continuous gradient for all $v_i \in \mathcal{V}_{\mathcal{R}}$, i.e., $\tilde{\mu} = \mu$ and $\tilde{L} = L$. Since our work is the first to achieve a linear convergence rate for Byzantine-resilient algorithms under these assumptions, we compare our results with nonresilient algorithms in the literature that consider the same assumptions.

To begin, recall that the convergence rate derived from our approach (see Theorem 5.8) is $\sqrt{\gamma}\sqrt{1 - \alpha\mu}$. For DGD with a constant step size α [42], it has been shown that the convergence rate is $\sqrt{1 - \frac{\alpha}{2} \cdot \frac{\mu L}{\mu + L}}$. Since it is the case that $\mu \leq L$, the best rate for DGD guaranteed by this work is $\sqrt{1 - \frac{1}{2}\alpha\mu}$ (which is the case when $\mu \ll L$). For DIGing [24], representing an algorithm from distributed optimization algorithms with the gradient tracking technique, it has been shown that the convergence rate is $\sqrt{1 - \frac{2}{3}\alpha\mu}$. Since a smaller rate yields faster convergence, our rate of convergence is superior to traditional DGD and DIGing (even though they all have the same order of $1 - \mathcal{O}(\alpha\mu)$). These convergence rates are summarized in Table 2.

Remark 5.9. Crucially, even in the absence of Byzantine agents, it is important to highlight that the states of regular agents within our framework converge to a neighborhood of \mathbf{x}^* , as formally affirmed in Theorem 5.8. Notably, this convergence to a neighborhood stands as a fundamental trait of Byzantine distributed optimization problems, regardless of the specific algorithms employed [33, 35]. This is in contrast to algorithms in traditional distributed optimization settings, where convergence to a neighborhood is an inherent property of the algorithm itself and can be avoided by changing the algorithm. Consequently, comparing the convergence radius between Byzantine and non-Byzantine settings may not be suitable.

TABLE 2
Convergence Rates of Algorithms Under Strong Convexity and Lipschitz Gradient Assumptions.

	DGD [42]	DIGing [24]	Our work (Theorem 5.8)
Convergence rate (lower is better)	$\sqrt{1 - \frac{1}{2}\alpha\mu}$	$\sqrt{1 - \frac{2}{3}\alpha\mu}$	$\sqrt{1 - \alpha\mu}$

Our result from Theorem 5.8 offers a different approach to convergence proofs than those typically found in the literature, which are often designed for specific algorithms. By focusing on proving the states contraction property (Definition 5.4), rather than the details of the functions involved, one can save a considerable amount of time and effort. However, it is worth noting that this approach only provides a sufficient condition for convergence. There may be resilient algorithms that do not satisfy the property but still converge geometrically. In fact, finding general necessary conditions for convergence in resilient distributed optimization remains an open question in the literature.

Remark 5.10. The work [36] introduces a contraction property which seems to be similar to Definition 5.4. However, there is a subtle difference in that their contraction center is time varying (since it is a function of neighbors' states) while it is a constant (but depends on algorithms) in our case. However, it is unclear whether their notion of contraction allows for the proof of geometric convergence, as demonstrated in Proposition 5.7 and Theorem 5.8.

Having established convergence of all regular agent's values to a ball that contains the true minimizer, we now turn our attention to characterizing the distance between the regular agents' values within that ball. Given the reduction property in Definition 5.5 (either Type-I or Type-II), we can use (5.5) to derive a bound on the distance between the values held by different nodes: $\limsup_k \|\mathbf{x}_i[k] - \mathbf{x}_j[k]\| \leq 2R^*$ for all $v_i, v_j \in \mathcal{V}_R$. However, this bound is not particularly useful since the right-hand side quantity can be large. In the next subsection, we will demonstrate that the mixing dynamics (Definition 5.6) and a constant step size are sufficient to obtain a more meaningful bound on the approximate consensus.

5.3. Convergence to approximate consensus of states. The following proposition characterizes the approximate consensus among the regular agents in the network under the mixing dynamics (Definition 5.6) and a constant step size (proved in Appendix B.2).

PROPOSITION 5.11. *If an algorithm A in REDGRAF satisfies the $(\{\mathbf{W}^{(\ell)}[k]\}, G)$ -mixing dynamics property (for some $\{\mathbf{W}^{(\ell)}[k]\}_{k \in \mathbb{N}, \ell \in [d]} \subset \mathbb{S}^{|\mathcal{V}_R|}$ and $G \in \mathbb{R}_{\geq 0}$) and $\alpha_k = \alpha$ for all $k \in \mathbb{N}$, then there exist $\rho \in \mathbb{R}_{\geq 0}$ and $\lambda \in (0, 1)$ such that*

$$(5.8) \quad \limsup_k \|\mathbf{x}_i[k] - \mathbf{x}_j[k]\| \leq \frac{\alpha \rho G \sqrt{d}}{1 - \lambda} \quad \text{for all } v_i, v_j \in \mathcal{V}_R.$$

From the consensus theorem above, we note that $\max_{v_i, v_j \in \mathcal{V}_R} \|\mathbf{x}_i[k] - \mathbf{x}_j[k]\| = \mathcal{O}(\alpha \sqrt{d})$ if G does not depend on the constant step size α and the dimension d .

Remark 5.12. According to [3], the quantity $\lambda \in (0, 1)$ depends only on the network topology (for each time step) induced by the sequence of graphs $\{\mathbb{G}(\mathbf{W}^{(\ell)}[k])\}$ while the quantity $\rho \in \mathbb{R}_{\geq 0}$ depends on the number of regular agents $|\mathcal{V}_R|$ and the quantity λ .

In fact, the states contraction property (Definition 5.4) implies a bound on the gradient $\|\mathbf{g}_i[k]\|_\infty$ (the formal statement is provided in Appendix B.1) which is one of the requirements of the mixing dynamics. Thus, we can achieve a similar approximate consensus result as Proposition 5.11 given that an algorithm satisfies the reduction property (Definition 5.5) and the associated sequence of graphs for each dimension is repeatedly jointly rooted as shown in the following theorem whose proof is provided in Appendix B.3.

THEOREM 5.13 (consensus). *Suppose Assumption 5.1 holds and an algorithm A satisfies the reduction property of Type-I or Type-II. If the dynamics of the regular*

states can be written as (5.3), where $\{\mathbb{G}(\mathbf{W}^{(\ell)}[k])\}_{k \in \mathbb{N}}$ is repeatedly jointly rooted for all $\ell \in [d]$, then there exist $\rho \in \mathbb{R}_{\geq 0}$ and $\lambda \in (0, 1)$ such that

$$(5.9) \quad \limsup_k \|\mathbf{x}_i[k] - \mathbf{x}_j[k]\| \leq \frac{\alpha \rho r_c \tilde{L} \sqrt{d}}{1 - \lambda} \left(1 + \frac{\sqrt{\alpha \gamma \tilde{L}}}{1 - \beta \sqrt{\gamma}} \right) := D^* \text{ for all } v_i, v_j \in \mathcal{V}_{\mathcal{R}},$$

where r_c and β are defined in (5.2) and (5.4), respectively. Furthermore, if $c[k] = \mathcal{O}(\xi^k)$, where $\xi \in (0, 1) \setminus \{\beta \sqrt{\gamma}\}$, then there exist $\rho \in \mathbb{R}_{\geq 0}$ and $\lambda \in (0, 1)$ such that

$$(5.10) \quad \|\mathbf{x}_i[k] - \mathbf{x}_j[k]\| \leq D^* + \mathcal{O}((\max\{\beta \sqrt{\gamma}, \xi, \lambda\})^k) \text{ for all } v_i, v_j \in \mathcal{V}_{\mathcal{R}}.$$

We refer to D^* in (5.9) as the *approximate consensus diameter*. From (5.10), we can conclude that the distance between any two regular agents' states converge *geometrically* to the approximate consensus diameter D^* . Furthermore, as suggested by (5.10), for the case that $\beta \sqrt{\gamma} > \max\{\xi, \lambda\}$, the distance converges faster as the constant step size α increases.

To analyze the expression of D^* , we simplify the expression as follows. Let $\text{dom}_{D^*} = \{(\kappa, \gamma, \tilde{\alpha}) \in \mathbb{R}^3 : \kappa \in [1, \infty), \gamma \in (0, 1) \text{ and } \tilde{\alpha} \in (0, \frac{1}{\kappa}], \text{ or } \kappa \in [1, \infty), \gamma \in [1, \frac{\kappa}{\kappa-1}] \text{ and } \tilde{\alpha} \in (1 - \frac{1}{\gamma}, \frac{1}{\kappa}]\}$. Using changes of variables $\kappa = \frac{\tilde{\mu}}{\mu}$ and $\tilde{\alpha} = \alpha \tilde{\mu}$ (as in subsection 5.2) and then normalizing the expression by $\frac{\rho r_c \sqrt{d}}{1 - \lambda}$, we have the (normalized) approximate consensus diameter $D_{\text{normalized}}^* : \text{dom}_{D^*} \rightarrow \mathbb{R}_{\geq 0}$ defined as

$$(5.11) \quad D_{\text{normalized}}^*(\kappa, \gamma, \tilde{\alpha}) = \kappa \tilde{\alpha} \left(1 + \frac{\sqrt{\kappa \gamma \tilde{\alpha}}}{1 - \sqrt{\gamma} \cdot \sqrt{1 - \tilde{\alpha}}} \right) = \kappa \tilde{\alpha} (1 + \sqrt{\kappa \gamma} \cdot R_{\text{normalized}}^*),$$

where $R_{\text{normalized}}^*$ is the (normalized) convergence radius defined in (5.7). It can be noted that $D_{\text{normalized}}^*$ is strictly increasing with both κ and γ . However, $D_{\text{normalized}}^*$ is neither an increasing nor decreasing function with respect to $\tilde{\alpha}$. The plots between the (normalized) approximate consensus diameter $D_{\text{normalized}}^*$ and the (scaled) constant step size $\tilde{\alpha}$ for some values of κ and γ are given in Figures 2(a) to 2(c). The plots suggest that for $\gamma \leq 1$, small constant step sizes α provide small approximate consensus diameters D^* while large constant step sizes α may be preferable in the case that $\gamma > 1$.

Additionally, by applying the approximation of R^* from subsection 5.2 to (5.11), we obtain insights regarding the dependence of the approximate consensus diameter D^* on the (scaled) constant step size $\tilde{\alpha}$ as shown below.

- When $\gamma \in (0, 1)$ and $\tilde{\alpha}$ approaches 0^+ , we have that $D^* \approx \frac{\rho r_c \kappa \sqrt{d}}{1 - \lambda} \cdot \tilde{\alpha}$.
- When $\gamma = 1$ and $\tilde{\alpha}$ approaches 0^+ , we have that $D^* \approx \frac{2 \rho r_c \kappa^{\frac{3}{2}} \sqrt{d}}{1 - \lambda} \cdot \tilde{\alpha}^{\frac{1}{2}}$.

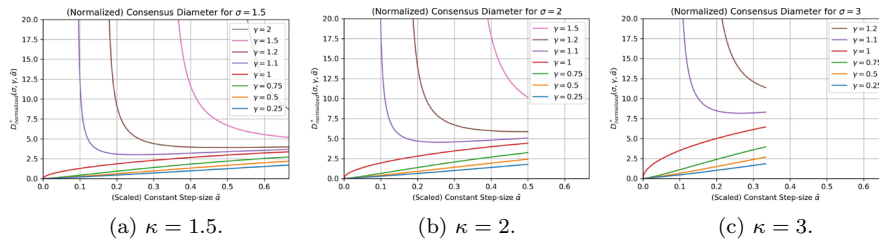


FIG. 2. The (normalized) approximate consensus diameter $D_{\text{normalized}}^*$ for different values of the contraction factor γ and for legitimate values of the (scaled) constant step size $\tilde{\alpha}$.

- When $\gamma \in (1, \infty)$ and $\tilde{\alpha}$ approaches $(1 - \frac{1}{\gamma})^+$, we have that $D^* \approx \frac{2\rho r_c}{1-\lambda} \sqrt{\frac{d}{\gamma}}$.
 $(\kappa(1 - \frac{1}{\gamma}))^{\frac{3}{2}} \cdot \hat{\alpha}^{-1}$, where $\hat{\alpha} = \tilde{\alpha} - (1 - \frac{1}{\gamma})$.

Remark 5.14. In fact, the states contraction property (Definition 5.4) implicitly relates to network connectivity, where connectivity assures the states' contraction property for nonfaulty cases. In such cases, network connectivity conditions suffice to achieve both convergence and consensus, as evidenced in [42] and [24]. However, in Byzantine scenarios, the states' contraction depends not only on connectivity but also on filter properties specific to each resilient algorithm. Our work introduces the states contraction property (Definition 5.4) to capture the desired characteristics of resilient algorithm filters, while the mixing dynamics property (Definition 5.6) aims to capture network connectivity (associated with the equivalent dynamics of regular agents). However, Theorem 5.13 presents a set of assumptions that lead to both the convergence and consensus results (even though the convergence result does not require the repeatedly jointly rooted assumption).

5.4. Implications for existing resilient distributed optimization algorithms. We now describe the implication of our above results (for our general framework) for the specific existing algorithms discussed in subsection 4.3: SDMMFD [20, 21], SDFD [21], CWTM [35, 33, 34, 9, 44, 7], and RVO [28, 1]. In particular, we show that the algorithms satisfy the states contraction (Definition 5.4) and the mixing dynamics (Definition 5.6) properties with different quantities which are determined in the following theorem.

Before stating the theorem, recall that $d \in \mathbb{Z}_+$ is the number of dimensions of the optimization variable \mathbf{x} in (3.3). For SDMMFD and SDFD, let $\mathbf{y}[\infty] \in \mathbb{R}^d$ be the point such that $\lim_{k \rightarrow \infty} \mathbf{y}_i[k] = \mathbf{y}[\infty]$ for all $v_i \in \mathcal{V}_{\mathcal{R}}$. Such a point exists due to [21, Proposition 1].

Additionally, recall that $F \in \mathbb{Z}_+$ is the parameter in the F -local model (Assumption 5.2). Since the step in RVO depends on a specific implemented algorithm, we assume that there exists a function $p : \mathbb{Z}_+ \times \mathbb{Z}_+ \rightarrow \mathbb{Z}_+$ such that if the graph \mathcal{G} is $p(d, F)$ -robust then the step in RVO returns a nonempty set of states.

THEOREM 5.15. *Suppose Assumptions 5.1–5.3 hold, $\alpha_k = \alpha$ for all $k \in \mathbb{N}$, and r_c and β are defined in (5.2) and (5.4), respectively. Let $\{0[k]\} = \{0\}_{k \in \mathbb{N}}$.*

- *If \mathcal{G} is $((2d + 1)F + 1)$ -robust then there exists $c_1, c_2 \in \mathbb{R}_{\geq 0}$ such that the SDMMFD from [20, 21] satisfies the $(\mathbf{y}[\infty], 1, \{2c_1 e^{-c_2 k}\})$ -states contraction property and there exists $\{\mathbf{W}^{(\ell)}[k]\}_{k \in \mathbb{N}, \ell \in [d]} \subset \mathbb{S}^{|\mathcal{V}_{\mathcal{R}}|}$ such that the algorithm satisfies the $(\{\mathbf{W}^{(\ell)}[k]\}, G)$ -mixing dynamics property with $G = r_c \tilde{L}(1 + \frac{\sqrt{\alpha \tilde{L}}}{1-\beta})$.*
- *If \mathcal{G} is $(2F + 1)$ -robust then there exists $c_1, c_2 \in \mathbb{R}_{\geq 0}$ such that the SDFD from [21] satisfies the $(\mathbf{y}[\infty], 1, \{2c_1 e^{-c_2 k}\})$ -states contraction property.*
- *If \mathcal{G} is $(2F + 1)$ -robust then the CWTM from [35, 33, 34, 9, 44, 7] satisfies the $(\mathbf{c}^*, d, \{0[k]\})$ -states contraction property and there exists $\{\mathbf{W}^{(\ell)}[k]\}_{k \in \mathbb{N}, \ell \in [d]} \subset \mathbb{S}^{|\mathcal{V}_{\mathcal{R}}|}$ such that the algorithm satisfies the $(\{\mathbf{W}^{(\ell)}[k]\}, G)$ -mixing dynamics property with $G = r_c \tilde{L}(1 + \frac{\sqrt{\alpha \tilde{L}}}{1-\beta \sqrt{d}})$.*
- *If \mathcal{G} is $p(d, F)$ -robust then the RVO from [28, 1] satisfies the $(\mathbf{c}^*, 1, \{0[k]\})$ -states contraction property and there exists $\{\mathbf{W}^{(\ell)}[k]\}_{k \in \mathbb{N}, \ell \in [d]} \subset \mathbb{S}^{|\mathcal{V}_{\mathcal{R}}|}$ such that the algorithm satisfies the $(\{\mathbf{W}^{(\ell)}[k]\}, G)$ -mixing dynamics property with $G = r_c \tilde{L}(1 + \frac{\sqrt{\alpha \tilde{L}}}{1-\beta})$.*

We provide an outline of the proof of Theorem 5.15 here, omitting the full details which can be found in [18]. To establish the states contraction property for each algorithm, we proceed as follows:

- For SDMMFD and SDFD, we combine the results from [21, Proposition 1] and [21, Lemma 2].
- For CWTM, we apply the results of [35, Proposition 5.1].
- For RVO, we manipulate the equation (4.4) to derive the required result.

To demonstrate the mixing dynamics property for each algorithm (specifically for SDMMFD, CWTM, and RVO), we proceed as follows:

- For both SDMMFD and CWTM, we rewrite the dynamics of the regular agents as in (5.3), utilizing [35, Theorem 6.1].
- For RVO, we similarly rewrite the dynamics of the regular agents using (4.4).

Next, we employ [35, Lemma 2.3], given the robustness conditions, to establish repeated jointly rootedness for the corresponding graph sequences. Finally, we determine the constant G in Definition 5.6 for each case by substituting the corresponding contraction factor γ into (B.2) (from Lemma B.1).

Now, we consider bounding the distance r_c , as defined in (5.2). It is worth noting that the constant r_c appears in two important quantities: the convergence radius R^* and approximate consensus diameter D^* defined in (5.5) and (5.9), respectively. In fact, r_c can be upper bounded by a quantity depending on the diameter of the minimizers of the regular agents' functions r^* defined in subsection 5.1. The formal statement is provided below while the proof is provided in Appendix C.1.

LEMMA 5.16. *Suppose Assumption 5.1 holds and for the initialization step of SDMMFD and SDFD, there exists $\epsilon^* \in \mathbb{R}_{\geq 0}$ such that $\|\hat{\mathbf{x}}_i^* - \mathbf{x}_i^*\|_\infty \leq \epsilon^*$ for all $v_i \in \mathcal{V}_{\mathcal{R}}$.*

- *For SDMMFD and SDFD, we have $r_c \leq \sqrt{d}(r^* + \epsilon^*) + r^*$.*
- *For CWTM and RVO, we have $r_c \leq r^*$.*

Applying the lemma to (5.5), we can conclude that the convergence radius R^* is $\mathcal{O}(\sqrt{d}r^*)$ for SDMMFD and SDFD, and $\mathcal{O}(r^*)$ for CWTM and RVO. Even though [35, 21] consider different set of assumptions, they also obtain the linear dependency on r^* . To achieve a convergence radius of $o(r^*)$, additional assumptions are required, as evident from [39, 7, 11]. We conjecture that $\mathcal{O}(r^*)$ may be an optimal characteristic, applicable to both convex cases (under bounded gradient assumptions) and strongly convex cases (under Lipschitz gradient assumptions). This assertion arises from the inherent ambiguity in determining the minimizer of the collective sum of all local functions, as discussed in [16, 17, 19, 43]. Still, the question regarding a tight lower bound on the convergence radius for the general case (whether how it depends on r^* , $\tilde{\mu}$, and \tilde{L}) remains an open problem.

Remark 5.17. It is worth noting that the algorithms proposed in [2] and [29] do not satisfy the states contraction property (Definition 5.4). However, in fact, they satisfy inequality (5.1) with the perturbation term being bounded by a constant, and thus it is not difficult to use our techniques to show that they geometrically converge to a region with the contraction center \mathbf{x}_c but the region has the radius greater than R^* given in (5.5). On the other hand, the algorithms in [10, 6] do not satisfy the contraction property and may require other techniques to establish convergence (if possible).

Having proved the states contraction and mixing dynamics properties of the algorithms from [35, 33, 34, 9, 44, 7, 20, 21, 28, 1], from Theorem 5.8, we can deduce that under certain conditions on the graph robustness and step size α_k , the states of the regular agents geometrically converge to the convergence region with \mathbf{x}_c and γ

determined by Theorem 5.15. On the other hand, from Theorem 5.13, we can deduce that the states of the regular agents geometrically converge together at least until the diameter reaches the approximate consensus diameter D^* (where the parameters r_c and γ depend on the implemented algorithm).

To the best of our knowledge, our work is the first to show the geometric convergence results and characterize the convergence region for the resilient algorithms mentioned above. Thus, our framework, defined properties, and proof techniques provide a general approach for analyzing the convergence region and rate for a wide class of resilient optimization algorithms.

6. Numerical experiments. We now present numerical experiments to illustrate the behavior of the algorithms discussed in subsection 4.3. Using synthetic quadratic functions, we investigate the geometric convergence properties of these algorithms and analyze how step size affects both convergence rates and final outcomes.⁴

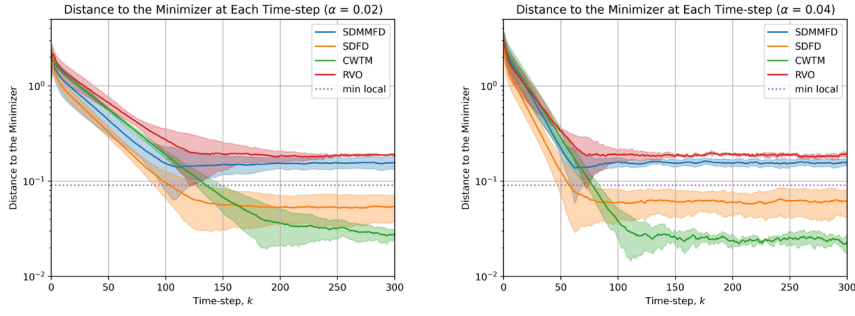
In particular, we focus on quadratic functions with two independent variables as the local cost functions, primarily due to the computational bottleneck posed by RVO. The network comprises 40 agents and is modeled as an 11-robust graph. We implement the F -local adversary model, setting $F = 2$. Each Byzantine agent transmits a random vector to its regular neighbors, designed to closely resemble other received vectors, increasing the likelihood of bypassing the filter applied by regular agents. For all algorithms (SDMMFD, SDFD, CWTM, and RVO), a constant step size of either $\alpha = 0.02$ or $\alpha = 0.04$ is used.

Keeping the network structure, local functions, and Byzantine agent identity consistent, we perform four independent runs of the experiment for each algorithm to account for the stochastic nature of adversary behavior and variability in the initialization of state and auxiliary vectors. The results, reported as the mean and standard deviation of key metrics, are averaged over all runs.

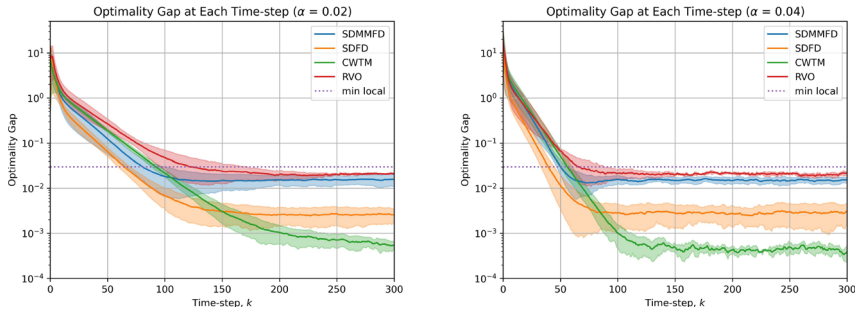
Let $\bar{\mathbf{x}}[k] = \frac{1}{|\mathcal{V}_R|} \sum_{v_i \in \mathcal{V}_R} \mathbf{x}_i[k]$ be the average of the states over regular agents at time step k . In Figure 3(a), each solid curve represents the Euclidean distance from the average state to the true minimizer, i.e., $\|\bar{\mathbf{x}}[k] - \mathbf{x}^*\|$, while the dotted line labeled `min_local` denotes the minimum Euclidean distance from the minimizers of the local functions to the true minimizer across all regular agents, i.e., $\min_{v_i \in \mathcal{V}_R} \|\mathbf{x}_i^* - \mathbf{x}^*\|$. In Figure 3(b), each solid curve corresponds to the optimality gap computed at the average state, i.e., $f(\bar{\mathbf{x}}[k]) - f^*$, and the dotted line labeled `min_local` indicates the minimum optimality gap computed at the local function minimizers among all regular agents, i.e., $\min_{v_i \in \mathcal{V}_R} f(\mathbf{x}_i^*) - f^*$. In Figure 3(c), each solid curve represents the maximum Euclidean distance between the states of any two regular agents (regular agents' diameter), i.e., $\max_{v_i, v_j \in \mathcal{V}_R} \|\mathbf{x}_i[k] - \mathbf{x}_j[k]\|$. In Figures 3(a) to 3(c), the solid curves are the means over all experiment rounds, and the shaded regions represent a ± 1 standard deviation from the means.

As we can see from Figure 3(a), for both cases, $\alpha = 0.02$ and $\alpha = 0.04$, the distances to the true minimizer drop at geometric rates in the early time steps. However, in the later time steps, these distances remain relatively stable, albeit with slight oscillations of small magnitudes. This behavior corresponds to the regular nodes' states entering the convergence region. Furthermore, it is notable that the convergence rates of all algorithms decrease, indicating faster convergence as the constant step size α transitions from 0.02 to 0.04, which aligns with the predictions from Theorem 5.8. Still, for each algorithm, the distances to the minimizer at later time steps are similar for

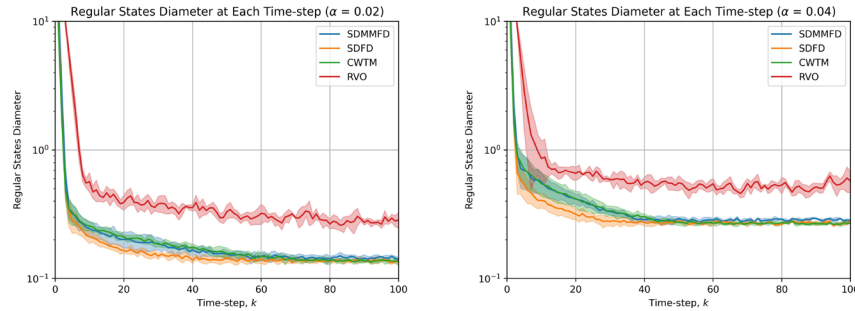
⁴Our code is available at <https://github.com/kkuwaran/resilient-distributed-optimization>.



(a) The Euclidean distance from the average of the regular agents' states to the true minimizer $\|\bar{\mathbf{x}} - \mathbf{x}^*\|$ for each algorithm.



(b) The optimality gap evaluated at the average of the regular agents' states $f(\bar{\mathbf{x}}) - f^*$ for each algorithm.



(c) The maximum Euclidean distance between two regular agents' states (regular states' diameter) $\max_{v_i, v_j \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_i - \mathbf{x}_j\|$ for each algorithm.

FIG. 3. The plots show the results obtained from SDMMFD (blue), SDFD (orange), CWTM (green), and RVO (red) for given constant step sizes $\alpha = 0.02$ (left) and $\alpha = 0.04$ (right).

both constant step sizes. Analogous patterns can be observed in the optimality gaps, mirroring the trends seen in the distance to the minimizer, as illustrated in Figure 3(b).

In Figure 3(c), representing the diameters of the regular states, it is important to note the shorter time horizon for the time step k . The diameters initially drop at a geometric rate during the early time steps, followed by relatively stable values. A slightly faster convergence can be observed for higher (constant) step sizes, with the values ceasing to drop at around $k = 60$ for $\alpha = 0.02$ and $k = 40$ for $\alpha = 0.04$. Additionally, the diameter at later time steps is evidently higher for larger step sizes compared to smaller ones. These results align well with the discussion in subsection 5.3 regarding approximate consensus, particularly Theorem 5.13 and Figure 2.

It is noteworthy that, even though Theorem 5.15 implies a contraction factor of $\gamma = d$ for CWTM (with $d = 2$ in this case), and $\gamma = 1$ for SDMMFD, SDFD, and RVO, for most of the time steps in this experiment, the states contraction property (5.1) for all algorithms holds with γ approximately less than 1. The insensitivity of distances to the minimizer at later time steps with respect to the step size α is attributed to the flat region observed in Figure 1(b), corresponding to $\gamma < 1$. In conclusion, the observed behaviors of all considered algorithms concerning the step size are comprehensively explained by the theories presented in section 5.

From Figures 3(a) and 3(b), we observe that although some algorithms exhibit worse distances to the minimizer compared to the local minimizers (i.e., comparing the solid curves to the dotted line), all algorithms are likely to achieve better optimality gaps than the local minimizers. Notably, all local minimizers fall within the convergence region $\mathcal{B}(\mathbf{x}_c, R^*)$. Additionally, when comparing the algorithms, SDFD shows higher uncertainty regarding distances to the minimizer and optimality gaps due to its vulnerability to hostile state vectors. Interestingly, RVO might have a higher effective contraction factor γ for this experiment, resulting in slower convergence and worse final values compared to the others across all metrics, as shown in Figure 3.

7. Conclusions. In this work, we considered the (peer-to-peer) distributed optimization problem in the presence of Byzantine agents. We introduced a general resilient (peer-to-peer) DGD framework called REDGRAF which includes some state-of-the-art resilient algorithms such as SDMMFD [20, 21], SDFD [21], CWTM [35, 33, 34, 9, 44, 7], and RVO [28, 1] as special cases. We analyzed the convergence of algorithms captured by our framework, assuming they satisfy a certain states contraction property. In particular, we derived a geometric rate of convergence of all regular agents to the convergence region under the strong convexity of the local functions and constant step-size regime. As we have shown, the convergence region, in fact, contains the true minimizer (the minimizer of the sum of the regular agents' functions). In addition, given a mixing dynamics property, we also derived a geometric rate of convergence of all regular agents to approximate consensus with a certain diameter under similar conditions. Considering each resilient algorithm, we analyzed the states contraction and mixing dynamics properties which, in turn, dictate the convergence rates, the size of the convergence region, and the approximate consensus diameter.

Future work includes developing resilient algorithms satisfying both the states contraction and mixing dynamics properties which give fast rates of convergence as well as a small convergence region and small approximate consensus diameter, identifying other properties for resilient algorithms to achieve good performance, analyzing the convergence property of other existing algorithms from the literature, considering nonconvex functions with certain properties, and establishing a tight lower bound for the convergence region.

Appendix A. Proof of convergence results in subsection 5.2.

A.1. Convex functions. From [45], an equivalent definition of a μ -strongly convex differentiable function f is as follows: for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$,

$$(A.1) \quad \langle \nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle \geq \mu \|\mathbf{x}_1 - \mathbf{x}_2\|^2.$$

We will use the following useful result from [45] regarding the convexity and Lipschitz gradient of a function.

LEMMA A.1. *If f is convex and has L -Lipschitz gradient then for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$,*

$$(A.2) \quad f(\mathbf{x}_1) \geq f(\mathbf{x}_2) + \langle \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle + \frac{1}{2L} \|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\|^2$$

and

$$(A.3) \quad f(\mathbf{x}_1) \leq f(\mathbf{x}_2) + \langle \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle + \frac{L}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|^2.$$

A.2. The reduction property implication. We first introduce the following lemma which is useful for deriving the convergence result (Proposition 5.7).

LEMMA A.2. *Suppose Assumption 5.1 holds. If an algorithm A in REDGRAF satisfies the reduction property of Type-I or Type-II, then $\beta\sqrt{\gamma} < 1$.*

Proof. In the first case, where $\gamma \in [0, 1)$ and $\alpha_k = \alpha \in (0, \frac{1}{L}]$, we have $\beta\sqrt{\gamma} = \sqrt{\gamma} \cdot \sqrt{1 - \alpha\tilde{\mu}} < 1$. In the second case, since $\gamma \in [1, \frac{1}{1 - \frac{L}{\mu}})$, we have that $0 \leq \frac{1}{\mu}(1 - \frac{1}{\gamma}) < \frac{1}{L}$ which indicates that setting the step size $\alpha_k = \alpha \in (\frac{1}{\mu}(1 - \frac{1}{\gamma}), \frac{1}{L}]$ is valid. Since $\alpha > \frac{1}{\mu}(1 - \frac{1}{\gamma})$, we also obtain that $\beta\sqrt{\gamma} = \sqrt{\gamma} \cdot \sqrt{1 - \alpha\tilde{\mu}} < 1$. For both cases, we have that $\beta\sqrt{\gamma} < 1$. \square

A.3. Proof of Proposition 5.7. We refactor Proposition 5.7 into Lemmas A.3 and A.4 where Lemma A.3 mainly captures the final convergence radius and Lemma A.4 captures the convergence rate.

LEMMA A.3. *Suppose Assumption 5.1 holds. If algorithm A in REDGRAF satisfies the $(\mathbf{x}_c, \gamma, \{c[k]\})$ -states contraction property (for some $\mathbf{x}_c \in \mathbb{R}^d$, $\gamma \in \mathbb{R}_{\geq 0}$, and $\{c[k]\}_{k \in \mathbb{N}} \subset \mathbb{R}$) and $\alpha_k = \alpha \in (0, \frac{1}{L}]$, then for all $k \in \mathbb{N}$ and $v_i \in \mathcal{V}_{\mathcal{R}}$,*

$$(A.4) \quad \begin{aligned} \|\mathbf{x}_i[k] - \mathbf{x}_c\| &\leq (\beta\sqrt{\gamma})^k \max_{v_s \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_s[0] - \mathbf{x}_c\| + \beta \sum_{s=0}^{k-1} (\beta\sqrt{\gamma})^s c[k-s-1] \\ &\quad + r_c \sqrt{\alpha\tilde{L}} \sum_{s=0}^{k-1} (\beta\sqrt{\gamma})^s, \end{aligned}$$

where r_c and β are defined in (5.2) and (5.4), respectively. Furthermore, if A satisfies the reduction property of Type-I or Type-II, then it holds that

$$(A.5) \quad \limsup_k \|\mathbf{x}_i[k] - \mathbf{x}_c\| \leq \frac{r_c \sqrt{\alpha\tilde{L}}}{1 - \beta\sqrt{\gamma}} \quad \text{for all } v_i \in \mathcal{V}_{\mathcal{R}}.$$

Proof. Consider a regular agent $v_i \in \mathcal{V}_{\mathcal{R}}$. Since $\mathbf{x}_i[k+1] = \tilde{\mathbf{x}}_i[k] - \alpha_k \mathbf{g}_i[k]$ from (4.1), we can write

$$(A.6) \quad \|\mathbf{x}_i[k+1] - \mathbf{x}_c\|^2 = \|\tilde{\mathbf{x}}_i[k] - \mathbf{x}_c\|^2 - 2\langle \tilde{\mathbf{x}}_i[k] - \mathbf{x}_c, \alpha_k \mathbf{g}_i[k] \rangle + \alpha_k^2 \|\mathbf{g}_i[k]\|^2.$$

Since f_i is μ_i -strongly convex (from Assumption 5.1), from (3.1) we have that $-\langle \tilde{\mathbf{x}}_i[k] - \mathbf{x}_c, \mathbf{g}_i[k] \rangle \leq (f_i(\mathbf{x}_c) - f_i(\tilde{\mathbf{x}}_i[k])) - \frac{\mu_i}{2} \|\tilde{\mathbf{x}}_i[k] - \mathbf{x}_c\|^2$. Substituting this inequality into (A.6) we get

$$(A.7) \quad \begin{aligned} \|\mathbf{x}_i[k+1] - \mathbf{x}_c\|^2 &\leq (1 - \alpha_k \mu_i) \|\tilde{\mathbf{x}}_i[k] - \mathbf{x}_c\|^2 + \alpha_k^2 \|\mathbf{g}_i[k]\|^2 \\ &\quad + 2\alpha_k (f_i(\mathbf{x}_c) - f_i(\tilde{\mathbf{x}}_i[k])). \end{aligned}$$

Since f_i has an L_i -Lipschitz gradient (from Assumption 5.1), from (A.2) we have that $\|\mathbf{g}_i[k]\|^2 \leq 2L_i(f_i(\tilde{\mathbf{x}}_i[k]) - f_i(\mathbf{x}_i^*))$. Substituting this inequality into (A.7) yields

$$(A.8) \quad \begin{aligned} \|\mathbf{x}_i[k+1] - \mathbf{x}_c\|^2 &\leq (1 - \alpha_k \mu_i) \|\tilde{\mathbf{x}}_i[k] - \mathbf{x}_c\|^2 - 2\alpha_k(1 - \alpha_k L_i) f_i(\tilde{\mathbf{x}}_i[k]) \\ &\quad - 2\alpha_k^2 L_i f_i(\mathbf{x}_i^*) + 2\alpha_k f_i(\mathbf{x}_c). \end{aligned}$$

Since f_i has an L_i -Lipschitz gradient (from Assumption 5.1), from (A.3) we have that $f_i(\mathbf{x}_c) \leq f_i(\mathbf{x}_i^*) + \frac{L_i}{2} \|\mathbf{x}_c - \mathbf{x}_i^*\|^2$. Substituting this inequality into (A.8), we obtain

$$\begin{aligned} \|\mathbf{x}_i[k+1] - \mathbf{x}_c\|^2 &\leq (1 - \alpha_k \mu_i) \|\tilde{\mathbf{x}}_i[k] - \mathbf{x}_c\|^2 \\ &\quad - 2\alpha_k(1 - \alpha_k L_i) (f_i(\tilde{\mathbf{x}}_i[k]) - f_i(\mathbf{x}_i^*)) + \alpha_k L_i \|\mathbf{x}_c - \mathbf{x}_i^*\|^2. \end{aligned}$$

Since $\alpha_k = \alpha \in (0, \frac{1}{L}]$, $L_i \leq \tilde{L}$, and $\mu_i \geq \tilde{\mu}$, the above inequality implies that

$$(A.9) \quad \|\mathbf{x}_i[k+1] - \mathbf{x}_c\|^2 \leq (1 - \alpha \tilde{\mu}) \|\tilde{\mathbf{x}}_i[k] - \mathbf{x}_c\|^2 + \alpha \tilde{L} \|\mathbf{x}_c - \mathbf{x}_i^*\|^2.$$

Since the algorithm A satisfies the $(\mathbf{x}_c, \gamma, \{c[k]\})$ -states contraction property given in (5.1), (A.9) becomes

$$\|\mathbf{x}_i[k+1] - \mathbf{x}_c\|^2 \leq (1 - \alpha \tilde{\mu}) \left(\sqrt{\gamma} \max_{v_j \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_j[k] - \mathbf{x}_c\| + c[k] \right)^2 + \alpha \tilde{L} \max_{v_j \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_c - \mathbf{x}_j^*\|^2,$$

which implies that

$$\begin{aligned} \|\mathbf{x}_i[k+1] - \mathbf{x}_c\| &\leq \sqrt{\gamma} \cdot \sqrt{1 - \alpha \tilde{\mu}} \max_{v_j \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_j[k] - \mathbf{x}_c\| \\ &\quad + \sqrt{1 - \alpha \tilde{\mu}} c[k] + \sqrt{\alpha \tilde{L}} \max_{v_j \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_c - \mathbf{x}_j^*\|. \end{aligned}$$

Recall the definition of r_c and β from (5.2) and (5.4), respectively. Since the above inequality holds for all $v_i \in \mathcal{V}_{\mathcal{R}}$, taking the maximum over $v_i \in \mathcal{V}_{\mathcal{R}}$ yields

$$\max_{v_i \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_i[k+1] - \mathbf{x}_c\| \leq \beta \sqrt{\gamma} \max_{v_i \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_i[k] - \mathbf{x}_c\| + \beta c[k] + r_c \sqrt{\alpha \tilde{L}}.$$

Unfolding the recursive inequality above, we obtain that

$$\begin{aligned} \max_{v_i \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_i[k] - \mathbf{x}_c\| &\leq (\beta \sqrt{\gamma})^k \max_{v_i \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_i[0] - \mathbf{x}_c\| \\ &\quad + \beta \sum_{s=0}^{k-1} (\beta \sqrt{\gamma})^s c[k-s-1] + r_c \sqrt{\alpha \tilde{L}} \sum_{s=0}^{k-1} (\beta \sqrt{\gamma})^s, \end{aligned}$$

which completes the first part of the proof.

Consider the second part of the lemma. Since the algorithm A satisfies the reduction property of Type-I or Type-II, from Lemma A.2 we have that $\beta \sqrt{\gamma} < 1$. Considering the right-hand side (RHS) of (A.4), since $\lim_{k \rightarrow \infty} \sum_{s=0}^k (\beta \sqrt{\gamma})^s$ is finite, using [18, Corollary A.3], we have

$$\begin{aligned} \lim_{k \rightarrow \infty} \left[r_c \sqrt{\alpha \tilde{L}} \sum_{s=0}^{k-1} (\beta \sqrt{\gamma})^s + \beta \sum_{s=0}^{k-1} (\beta \sqrt{\gamma})^s c[k-s-1] \right. \\ \left. + (\beta \sqrt{\gamma})^k \max_{v_s \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_s[0] - \mathbf{x}_c\| \right] = \frac{r_c \sqrt{\alpha \tilde{L}}}{1 - \beta \sqrt{\gamma}}. \end{aligned}$$

The result (A.5) follows from taking \limsup_k on both sides of (A.4) and then applying the above equation. \square

LEMMA A.4. Suppose Assumption 5.1 holds and an algorithm A in REDGRAF satisfies the reduction property of Type-I or Type-II. If the perturbation sequence $c[k] = \mathcal{O}(\xi^k)$ with $\xi \in (0, 1) \setminus \{\beta\sqrt{\gamma}\}$, then

$$(A.10) \quad \|\mathbf{x}_i[k] - \mathbf{x}_c\| \leq R^* + \mathcal{O}((\max\{\beta\sqrt{\gamma}, \xi\})^k) \quad \text{for all } v_i \in \mathcal{V}_{\mathcal{R}}.$$

Proof. Consider the second term on the RHS of (A.4). Since $c[k] = \mathcal{O}(\xi^k)$, we obtain that

$$\begin{aligned} \beta \sum_{s=0}^{k-1} (\beta\sqrt{\gamma})^s c[k-s-1] &= \mathcal{O}\left(\xi^{k-1} \sum_{s=0}^{k-1} \left(\frac{\beta\sqrt{\gamma}}{\xi}\right)^s\right) \\ &= \mathcal{O}\left(\frac{(\beta\sqrt{\gamma})^k - \xi^k}{\beta\sqrt{\gamma} - \xi}\right) = \mathcal{O}((\max\{\beta\sqrt{\gamma}, \xi\})^k). \end{aligned}$$

Using the above equation and the fact that $r_c \sqrt{\alpha \tilde{L}} \cdot \sum_{s=0}^{k-1} (\beta\sqrt{\gamma})^s \leq R^*$, we have that (A.4) implies (A.10). \square

A.4. Convergence region containment of the true minimizer.

LEMMA A.5. Suppose Assumption 5.1 holds, and let \mathbf{x}^* be the minimizer of the function (3.3). For a given vector $\mathbf{x}_c \in \mathbb{R}^d$, if $r_c = \max_{v_i \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_c - \mathbf{x}_i^*\|$, then $\mathbf{x}^* \in \mathcal{B}(\mathbf{x}_c, \frac{\tilde{L}}{\mu} r_c)$. Furthermore, if $\gamma \in [1, \frac{1}{1-\frac{\tilde{L}}{\mu}})$ and $\alpha \in (\frac{1}{\mu}(1 - \frac{1}{\gamma}), \frac{1}{\tilde{L}}]$, then $\mathbf{x}^* \in \mathcal{B}(\mathbf{x}_c, R^*)$.

Proof. Suppose \mathbf{x} is a point in \mathbb{R}^d such that $\|\mathbf{x} - \mathbf{x}_c\| > \frac{\tilde{L}}{\mu} r_c$. In order to conclude that $\mathbf{x}^* \in \mathcal{B}(\mathbf{x}_c, \frac{\tilde{L}}{\mu} r_c)$, we will show that $\sum_{v_i \in \mathcal{V}_{\mathcal{R}}} \nabla f_i(\mathbf{x}) \neq \mathbf{0}$.

In the first step, we will show that $\cos \angle(\nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_c) > 0$ for all $v_i \in \mathcal{V}_{\mathcal{R}}$. For a regular agent $v_i \in \mathcal{V}_{\mathcal{R}}$, consider the angle between the vectors $\mathbf{x} - \mathbf{x}_c$ and $\mathbf{x} - \mathbf{x}_i^*$. Suppose $r_c > 0$; otherwise, we have that $\angle(\mathbf{x} - \mathbf{x}_c, \mathbf{x} - \mathbf{x}_i^*) = 0$. Using Lemma A.1 from the arXiv version of [21], we can bound the angle as follows:

$$(A.11) \quad \begin{aligned} \angle(\mathbf{x} - \mathbf{x}_c, \mathbf{x} - \mathbf{x}_i^*) &\leq \max_{\mathbf{x}_0 \in \mathcal{B}(\mathbf{x}_c, r_c)} \angle(\mathbf{x} - \mathbf{x}_c, \mathbf{x} - \mathbf{x}_0) \\ &= \arcsin\left(\frac{r_c}{\|\mathbf{x} - \mathbf{x}_c\|}\right) < \arcsin\left(\frac{\tilde{\mu}}{\tilde{L}}\right). \end{aligned}$$

On the other hand, for a regular agent $v_i \in \mathcal{V}_{\mathcal{R}}$, since f_i is μ_i -strongly convex, from (A.1) we have that $\langle \nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_i^* \rangle \geq \mu_i \|\mathbf{x} - \mathbf{x}_i^*\|^2$ which is equivalent to

$$(A.12) \quad \|\nabla f_i(\mathbf{x})\| \cos \angle(\nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_i^*) \geq \mu_i \|\mathbf{x} - \mathbf{x}_i^*\|.$$

Since f_i has an L_i -Lipschitz gradient, from (3.2) we have that $\|\nabla f_i(\mathbf{x})\| \leq L_i \|\mathbf{x} - \mathbf{x}_i^*\|$. Substitute this inequality into (A.12) to obtain $\cos \angle(\nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_i^*) \geq \frac{\mu_i}{L_i} \geq \frac{\tilde{\mu}}{\tilde{L}}$ which implies that

$$(A.13) \quad \angle(\nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_i^*) \leq \arccos\left(\frac{\tilde{\mu}}{\tilde{L}}\right).$$

Using (A.11) and (A.13), we can bound the angle between the vectors $\nabla f_i(\mathbf{x})$ and $\mathbf{x} - \mathbf{x}_c$ as follows:

$$\begin{aligned} \angle(\nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_c) &\leq \angle(\nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_i^*) + \angle(\mathbf{x} - \mathbf{x}_c, \mathbf{x} - \mathbf{x}_i^*) \\ &< \arccos\left(\frac{\tilde{\mu}}{\tilde{L}}\right) + \arcsin\left(\frac{\tilde{\mu}}{\tilde{L}}\right) = \frac{\pi}{2}, \end{aligned}$$

where the first inequality is obtained from Corollary 12 in [4]. This means that $\cos \angle(\nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_c) > 0$ as desired.

In the second step, we will show that $\|\nabla f_i(\mathbf{x})\| > 0$ for all $v_i \in \mathcal{V}_{\mathcal{R}}$. For a regular agent $v_i \in \mathcal{V}_{\mathcal{R}}$, consider the lower bound of the gradient's norm $\|\nabla f_i(\mathbf{x})\|$ in (A.12) which implies that

$$\|\nabla f_i(\mathbf{x})\| \geq \mu_i \|\mathbf{x} - \mathbf{x}_i^*\| \geq \mu_i (\|\mathbf{x} - \mathbf{x}_c\| - \|\mathbf{x}_i^* - \mathbf{x}_c\|).$$

Since $\|\mathbf{x} - \mathbf{x}_c\| > \frac{\tilde{L}}{\tilde{\mu}} r_c$, the above inequality becomes

$$\|\nabla f_i(\mathbf{x})\| > \left(\frac{\tilde{L}}{\tilde{\mu}} \max_{v_j \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_j^* - \mathbf{x}_c\| - \|\mathbf{x}_i^* - \mathbf{x}_c\| \right) \geq 0,$$

where the second inequality is obtained by using $\tilde{L} \geq \tilde{\mu}$.

In the last step, we will show that $\sum_{v_i \in \mathcal{V}_{\mathcal{R}}} \nabla f_i(\mathbf{x}) \neq \mathbf{0}$. Consider the following inner product

$$\left\langle \sum_{v_i \in \mathcal{V}_{\mathcal{R}}} \nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_c \right\rangle = \|\mathbf{x} - \mathbf{x}_c\| \sum_{v_i \in \mathcal{V}_{\mathcal{R}}} \|\nabla f_i(\mathbf{x})\| \cos \angle(\nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_c).$$

Since $\|\nabla f_i(\mathbf{x})\| > 0$ and $\cos \angle(\nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_c) > 0$ for all $v_i \in \mathcal{V}_{\mathcal{R}}$, and $\|\mathbf{x} - \mathbf{x}_c\| > 0$, we have that $\langle \sum_{v_i \in \mathcal{V}_{\mathcal{R}}} \nabla f_i(\mathbf{x}), \mathbf{x} - \mathbf{x}_c \rangle > 0$. This implies that $\sum_{v_i \in \mathcal{V}_{\mathcal{R}}} \nabla f_i(\mathbf{x}) \neq \mathbf{0}$ which completes the first part of the proof.

For the second part of the lemma, in order to conclude that $\mathbf{x}^* \in \mathcal{B}(\mathbf{x}_c, R^*)$, we will show that $\frac{\tilde{L}}{\tilde{\mu}} r_c \leq R^*$, where R^* is defined in (5.5). Since $\gamma \geq 1$, we have that

$$R^* \geq \frac{r_c \sqrt{\alpha \tilde{L}}}{1 - \sqrt{1 - \alpha \tilde{\mu}}}.$$

Multiplying $1 + \sqrt{1 - \alpha \tilde{\mu}}$ to both the numerator and denominator of the RHS of the above inequality, we obtain that

$$R^* \geq r_c \sqrt{\frac{\tilde{L}}{\tilde{\mu}}} \left(\frac{1}{\sqrt{\alpha \tilde{\mu}}} + \sqrt{\frac{1}{\alpha \tilde{\mu}} - 1} \right).$$

Since $\alpha \leq \frac{1}{\tilde{L}}$ implies that $\frac{1}{\alpha \tilde{\mu}} \geq \frac{\tilde{L}}{\tilde{\mu}}$, we can bound R^* as follows:

$$R^* \geq r_c \sqrt{\frac{\tilde{L}}{\tilde{\mu}}} \left(\sqrt{\frac{\tilde{L}}{\tilde{\mu}}} + \sqrt{\frac{\tilde{L}}{\tilde{\mu}} - 1} \right) = r_c \left(\frac{\tilde{L}}{\tilde{\mu}} + \sqrt{\frac{\tilde{L}}{\tilde{\mu}}} \left(\frac{\tilde{L}}{\tilde{\mu}} - 1 \right) \right).$$

Since $\tilde{L} \geq \tilde{\mu}$, we obtain that $R^* \geq \frac{\tilde{L}}{\tilde{\mu}} r_c$ which completes the second part of the proof. \square

Appendix B. Proof of consensus results in subsection 5.3.

B.1. Bound on gradients. As we have claimed in the main text, the states contraction property (Definition 5.4) implies a bound on the gradient $\|\mathbf{g}_i[k]\|_{\infty}$. The following lemma formally illustrates this fact.

LEMMA B.1. *Suppose Assumption 5.1 holds. If an algorithm A in REDGRAF satisfies the $(\mathbf{x}_c, \gamma, \{c[k]\})$ -states contraction property (for some $\mathbf{x}_c \in \mathbb{R}^d$, $\gamma \in \mathbb{R}_{\geq 0}$, and $\{c[k]\}_{k \in \mathbb{N}} \subset \mathbb{R}$) and $\alpha_k = \alpha \in (0, \frac{1}{\tilde{L}}]$, then for all $k \in \mathbb{N}$ and $v_i \in \mathcal{V}_{\mathcal{R}}$,*

$$(B.1) \quad \begin{aligned} \|\mathbf{g}_i[k]\|_{\infty} &\leq \tilde{L}\sqrt{\gamma} \left[(\beta\sqrt{\gamma})^k \max_{v_s \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_s[0] - \mathbf{x}_c\| \right. \\ &\quad \left. + \beta \sum_{s=0}^{k-1} (\beta\sqrt{\gamma})^s c[k-s-1] + r_c \sqrt{\alpha\tilde{L}} \sum_{s=0}^{k-1} (\beta\sqrt{\gamma})^s \right] + \tilde{L}c[k] + r_c\tilde{L}, \end{aligned}$$

where r_c and β are defined in (5.2) and (5.4), respectively. Furthermore, if A satisfies the reduction property of Type-I or Type-II, then it holds that

$$(B.2) \quad \limsup_k \|\mathbf{g}_i[k]\|_{\infty} \leq r_c \tilde{L} \left(1 + \frac{\sqrt{\alpha\gamma\tilde{L}}}{1 - \beta\sqrt{\gamma}} \right) \quad \text{for all } v_i \in \mathcal{V}_{\mathcal{R}}.$$

Proof. Consider a time step $k \in \mathbb{N}$, and a regular agent $v_i \in \mathcal{V}_{\mathcal{R}}$. We can write

$$\|\tilde{\mathbf{x}}_i[k] - \mathbf{x}_i^*\| \leq \|\tilde{\mathbf{x}}_i[k] - \mathbf{x}_c\| + \|\mathbf{x}_i^* - \mathbf{x}_c\|.$$

Since the algorithm A satisfies the $(\mathbf{x}_c, \gamma, \{c[k]\})$ -states contraction property, applying (5.1) to the above inequality yields

$$(B.3) \quad \|\tilde{\mathbf{x}}_i[k] - \mathbf{x}_i^*\| \leq \sqrt{\gamma} \max_{v_j \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_j[k] - \mathbf{x}_c\| + c[k] + r_c,$$

where r_c is defined in (5.2). Since $\mathbf{g}_i[k] = \nabla f_i(\tilde{\mathbf{x}}_i[k])$, using Assumption 5.1 we can write

$$\|\mathbf{g}_i[k]\| = \|\nabla f_i(\tilde{\mathbf{x}}_i[k]) - \nabla f_i(\mathbf{x}_i^*)\| \leq \tilde{L} \|\tilde{\mathbf{x}}_i[k] - \mathbf{x}_i^*\|.$$

Substituting (B.3) into the above inequality and using the fact that $\|\mathbf{g}_i[k]\|_{\infty} \leq \|\mathbf{g}_i[k]\|$, we obtain that

$$(B.4) \quad \|\mathbf{g}_i[k]\|_{\infty} \leq \tilde{L}\sqrt{\gamma} \max_{v_j \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_j[k] - \mathbf{x}_c\| + \tilde{L}c[k] + r_c\tilde{L}.$$

Substituting (A.4) from Lemma A.3 into the above inequality yields (B.1).

To show the second part of the lemma, taking \limsup_k to both sides of (B.4), we have that

$$\limsup_k \|\mathbf{g}_i[k]\|_{\infty} \leq \tilde{L}\sqrt{\gamma} \limsup_k \max_{v_j \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_j[k] - \mathbf{x}_c\| + \tilde{L} \lim_{k \rightarrow \infty} c[k] + r_c\tilde{L}.$$

Using (5.5) from Proposition 5.7 and $\lim_{k \rightarrow \infty} c[k] = 0$ yields the result (B.2). \square

B.2. Proof of Proposition 5.11. Here, we present a more general version of Proposition 5.11 which will be used to prove Theorem 5.13.

PROPOSITION B.2. *If an algorithm A in REDGRAF satisfies the $(\{\mathbf{W}^{(\ell)}[k]\}, G)$ -mixing dynamics property (for some $\{\mathbf{W}^{(\ell)}[k]\}_{k \in \mathbb{N}, \ell \in [d]} \subset \mathbb{S}^{|\mathcal{V}_{\mathcal{R}}|}$ and $G \in \mathbb{R}_{\geq 0}$) and $\alpha_k = \alpha$ for all $k \in \mathbb{N}$, then there exist $\rho \in \mathbb{R}_{\geq 0}$ and $\lambda \in (0, 1)$ such that for all $k \in \mathbb{N}$ and $v_i, v_j \in \mathcal{V}_{\mathcal{R}}$, it holds that*

$$(B.5) \quad \|\mathbf{x}_i[k] - \mathbf{x}_j[k]\| \leq \rho\sqrt{d} \left(\lambda^k \max_{v_r \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_r[0]\|_{\infty} + \alpha \sum_{s=0}^{k-1} \lambda^{k-s-1} \max_{v_r \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{g}_r[s]\|_{\infty} \right).$$

Furthermore, there exist $\rho \in \mathbb{R}_{\geq 0}$ and $\lambda \in (0, 1)$ such that

$$(B.6) \quad \limsup_k \|\mathbf{x}_i[k] - \mathbf{x}_j[k]\| \leq \frac{\alpha\rho G\sqrt{d}}{1 - \lambda} \quad \text{for all } v_i, v_j \in \mathcal{V}_{\mathcal{R}}.$$

Proof. Consider a time step $k \in \mathbb{Z}_+$ and a dimension $\ell \in [d]$. Since the algorithm A satisfies the $(\{\mathbf{W}^{(\ell)}[k]\}, G)$ -mixing dynamics property in (5.3), we have that

$$(B.7) \quad \mathbf{x}^{(\ell)}[k] = \mathbf{W}^{(\ell)}[k-1]\mathbf{x}^{(\ell)}[k-1] - \alpha_{k-1}\mathbf{g}^{(\ell)}[k-1].$$

For $s, t \in \mathbb{N}$, let

$$\Phi^{(\ell)}[t, s] = \begin{cases} \mathbf{W}^{(\ell)}[t]\mathbf{W}^{(\ell)}[t-1]\cdots\mathbf{W}^{(\ell)}[s] & \text{if } t \geq s, \\ \mathbf{I} & \text{if } t < s, \end{cases}$$

We can expand (B.7) as follows:

$$(B.8) \quad \mathbf{x}^{(\ell)}[k] = \Phi^{(\ell)}[k-1, 0]\mathbf{x}^{(\ell)}[0] - \sum_{s=0}^{k-1} \alpha_s \Phi^{(\ell)}[k-1, s+1]\mathbf{g}^{(\ell)}[s].$$

Let $\mathbf{q}^{(\ell)}(s) \in \mathbb{R}^{|\mathcal{V}_{\mathcal{R}}|}$ be such that $\lim_{t \rightarrow \infty} \Phi^{(\ell)}[t, s] = \mathbf{1}\mathbf{q}^{(\ell)T}[s]$, and let $\bar{\mathbf{x}}^{(\ell)}[k] = \mathbf{1}\mathbf{q}^{(\ell)T}[k]\mathbf{x}^{(\ell)}[k]$. We can write

$$\|\mathbf{x}^{(\ell)}[k] - \bar{\mathbf{x}}^{(\ell)}[k]\|_{\infty} = \left\| (\mathbf{I} - \mathbf{1}\mathbf{q}^{(\ell)T}[k])\mathbf{x}^{(\ell)}[k] \right\|_{\infty}.$$

Applying (B.8) to the above equation, we obtain that

$$(B.9) \quad \begin{aligned} \|\mathbf{x}^{(\ell)}[k] - \bar{\mathbf{x}}^{(\ell)}[k]\|_{\infty} &\leq \left\| \Phi^{(\ell)}[k-1, 0] - \mathbf{1}\mathbf{q}^{(\ell)T}[0] \right\|_{\infty} \|\mathbf{x}^{(\ell)}[0]\|_{\infty} \\ &+ \sum_{s=0}^{k-1} \left(\alpha_s \left\| \Phi^{(\ell)}[k-1, s+1] - \mathbf{1}\mathbf{q}^{(\ell)T}[s+1] \right\|_{\infty} \|\mathbf{g}^{(\ell)}[s]\|_{\infty} \right). \end{aligned}$$

From Proposition 1 in [3], we have that there exist constants $\rho' \in \mathbb{R}_{\geq 0}$ and $\lambda \in (0, 1)$ such that for all $k > s \geq 0$,

$$\left\| \Phi^{(\ell)}[k-1, s] - \mathbf{1}\mathbf{q}^{(\ell)T}[s] \right\|_{\infty} \leq \rho' \lambda^{k-s}.$$

Thus, applying the above inequality, (B.9) can be bounded as

$$(B.10) \quad \|\mathbf{x}^{(\ell)}[k] - \bar{\mathbf{x}}^{(\ell)}[k]\|_{\infty} \leq \rho' \lambda^k \|\mathbf{x}^{(\ell)}[0]\|_{\infty} + \sum_{s=0}^{k-1} \alpha_s \rho' \lambda^{k-s-1} \|\mathbf{g}^{(\ell)}[s]\|_{\infty}.$$

Since $\alpha_s = \alpha$ for all $s \in \mathbb{N}$, and for $s \in \mathbb{N}$, $\ell \in [d]$, and $\mathbf{z}^{(\ell)}[s] = \mathbf{x}^{(\ell)}[s]$ or $\mathbf{g}^{(\ell)}[s]$,

$$\|\mathbf{z}^{(\ell)}[s]\|_{\infty} \leq \max_{\ell \in [d]} \max_{v_i \in \mathcal{V}_{\mathcal{R}}} |z_i^{(\ell)}[s]| = \max_{v_i \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{z}_i[s]\|_{\infty},$$

the inequality (B.10) becomes

$$(B.11) \quad \|\mathbf{x}^{(\ell)}[k] - \bar{\mathbf{x}}^{(\ell)}[k]\|_{\infty} \leq \rho' \lambda^k \max_{v_i \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_i[0]\|_{\infty} + \alpha \rho' \sum_{s=0}^{k-1} \lambda^{k-s-1} \max_{v_i \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{g}_i[s]\|_{\infty}.$$

Let $\bar{\mathbf{x}}^{(\ell)}[k] = \mathbf{q}^{(\ell)T}[k]\mathbf{x}^{(\ell)}[k]$. For $v_i \in \mathcal{V}_{\mathcal{R}}$, we can write

$$\|\mathbf{x}_i[k] - \bar{\mathbf{x}}[k]\| = \sqrt{\sum_{\ell \in [d]} |x_i^{(\ell)}[k] - \bar{x}^{(\ell)}[k]|^2} \leq \sqrt{\sum_{\ell \in [d]} \|\mathbf{x}^{(\ell)}[k] - \bar{\mathbf{x}}^{(\ell)}[k]\|_{\infty}^2}.$$

Using the above inequality, we have that for all $v_i, v_j \in \mathcal{V}_{\mathcal{R}}$,

$$\|\mathbf{x}_i[k] - \mathbf{x}_j[k]\| \leq \|\mathbf{x}_i[k] - \bar{\mathbf{x}}[k]\| + \|\mathbf{x}_j[k] - \bar{\mathbf{x}}[k]\| \leq 2 \sqrt{\sum_{\ell \in [d]} \|\mathbf{x}^{(\ell)}[k] - \bar{\mathbf{x}}^{(\ell)}[k]\|_{\infty}^2}.$$

Substituting (B.11) into the above inequality and letting $\rho = 2\rho'$, we obtain the result (B.5). Taking \limsup_k to both sides of (B.5), we have

$$\limsup_k \|\mathbf{x}_i[k] - \mathbf{x}_j[k]\| \leq \alpha \rho \sqrt{d} \limsup_k \sum_{s=0}^{k-1} \lambda^{k-s-1} \max_{v_r \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{g}_r[s]\|_{\infty}.$$

Since for all $v_r \in \mathcal{V}_{\mathcal{R}}$, we have $\limsup_k \|\mathbf{g}_r[k]\|_{\infty} \leq G$ from Definition 5.6 and $\lim_{k \rightarrow \infty} \sum_{s=0}^k \lambda^k = \frac{1}{1-\lambda}$, using [18, Lemma A.2], the above inequality becomes (B.6). \square

B.3. Proof of Theorem 5.13.

Proof of Theorem 5.13. From the inequality (B.2) in Lemma B.1, we have that the algorithm A satisfies the $(\{\mathbf{W}^{(\ell)}[k]\}, G)$ -mixing dynamics property with

$$G = r_c \tilde{L} \left(1 + \frac{\sqrt{\alpha \gamma \tilde{L}}}{1 - \beta \sqrt{\gamma}} \right).$$

Substituting G into (5.8) in Proposition 5.11 yields the result (5.9).

To show the second part of the theorem, consider the expression in the square bracket of (B.1). Since $c[k] = \mathcal{O}(\xi^k)$ and $\xi \in (0, 1) \setminus \{\beta \sqrt{\gamma}\}$, we have that

$$\begin{aligned} & (\beta \sqrt{\gamma})^k \max_{v_s \in \mathcal{V}_{\mathcal{R}}} \|\mathbf{x}_s[0] - \mathbf{x}_c\| + \beta \sum_{s=0}^{k-1} (\beta \sqrt{\gamma})^s c[k-s-1] + r_c \sqrt{\alpha \tilde{L}} \sum_{s=0}^{k-1} (\beta \sqrt{\gamma})^s \\ & \leq R^* + \mathcal{O}((\max\{\beta \sqrt{\gamma}, \xi\})^k), \end{aligned}$$

where R^* is defined in (5.5). Substituting the above inequality into (B.1), we obtain that for all $v_i \in \mathcal{V}_{\mathcal{R}}$,

$$\begin{aligned} \|\mathbf{g}_i[k]\|_{\infty} & \leq \tilde{L} \sqrt{\gamma} \left[R^* + \mathcal{O}((\max\{\beta \sqrt{\gamma}, \xi\})^k) \right] + \mathcal{O}(\xi^k) + r_c \tilde{L} \\ & = R^* \tilde{L} \sqrt{\gamma} + r_c \tilde{L} + \mathcal{O}((\max\{\beta \sqrt{\gamma}, \xi\})^k). \end{aligned}$$

Substituting the above inequality into (B.5) in Proposition B.2, we have that there exist $\rho \in \mathbb{R}_{\geq 0}$ and $\lambda \in (0, 1)$ such that for all $v_i, v_j \in \mathcal{V}_{\mathcal{R}}$,

$$\begin{aligned} \|\mathbf{x}_i[k] - \mathbf{x}_j[k]\| & \leq \rho \sqrt{d} \left[\mathcal{O}(\lambda^k) + \frac{\alpha \tilde{L}}{1 - \lambda} (r_c + R^* \sqrt{\gamma}) \right. \\ & \quad \left. + \alpha \sum_{s=0}^{k-1} \lambda^{k-s-1} \mathcal{O}((\max\{\beta \sqrt{\gamma}, \xi\})^s) \right]. \end{aligned} \tag{B.12}$$

If $\lambda = \max\{\beta \sqrt{\gamma}, \xi\}$, we can replace λ in (B.12) with $\lambda' = \lambda + \epsilon$, where $\epsilon \in (0, 1 - \lambda)$. Thus, without loss of generality, there exist $\rho \in \mathbb{R}_{\geq 0}$ and $\lambda \in (0, 1) \setminus \{\max\{\beta \sqrt{\gamma}, \xi\}\}$ such that for all $v_i, v_j \in \mathcal{V}_{\mathcal{R}}$, the inequality (B.12) holds. Consider the last term of (B.12). Since $\lambda \neq \max\{\beta \sqrt{\gamma}, \xi\}$, we have that

$$\sum_{s=0}^{k-1} \lambda^{k-s-1} \mathcal{O}((\max\{\beta \sqrt{\gamma}, \xi\})^s) = \mathcal{O}((\max\{\beta \sqrt{\gamma}, \xi, \lambda\})^k).$$

Substituting the above equality into (B.12) yields the result (5.10). \square

Appendix C. Proof of algorithms results in subsection 5.4.

C.1. Proof of Lemma 5.16.

Proof of Lemma 5.16. First, consider the case where the regular agents follow SDMMFD or SDFD [20, 21]. From Proposition 1 in [21], we have that for all $\ell \in [d]$,

$$(C.1) \quad y^{(\ell)}[\infty] \in \left[\min_{v_i \in \mathcal{V}_{\mathcal{R}}} y_i^{(\ell)}[0], \max_{v_i \in \mathcal{V}_{\mathcal{R}}} y_i^{(\ell)}[0] \right].$$

Since in the initialization step, we set $y_i[0] = \hat{x}_i^*$ for all $v_i \in \mathcal{V}_{\mathcal{R}}$, we can rewrite (C.1) as

$$y^{(\ell)}[\infty] \in \left[\min_{v_i \in \mathcal{V}_{\mathcal{R}}} \hat{x}_i^{*(\ell)}, \max_{v_i \in \mathcal{V}_{\mathcal{R}}} \hat{x}_i^{*(\ell)} \right].$$

Using the above expression, we can write

$$(C.2) \quad y^{(\ell)}[\infty] - c^{*(\ell)} \leq \max_{v_i \in \mathcal{V}_{\mathcal{R}}} \hat{x}_i^{*(\ell)} - c^{*(\ell)}.$$

Let $v_{i'} \in \mathcal{V}_{\mathcal{R}}$ be an agent such that $\hat{x}_{i'}^{*(\ell)} = \max_{v_i \in \mathcal{V}_{\mathcal{R}}} \hat{x}_i^{*(\ell)}$. We can rewrite inequality (C.2) as

$$y^{(\ell)}[\infty] - c^{*(\ell)} \leq (\hat{x}_{i'}^{*(\ell)} - x_{i'}^{*(\ell)}) + (x_{i'}^{*(\ell)} - c^{*(\ell)}).$$

Since $\|\hat{x}_{i'}^* - x_{i'}^*\|_{\infty} \leq \epsilon^*$ and $\|x_{i'}^* - c^*\| \leq r^*$ from the definition of c^* and r^* , the above inequality becomes

$$|y^{(\ell)}[\infty] - c^{*(\ell)}| \leq |\hat{x}_{i'}^{*(\ell)} - x_{i'}^{*(\ell)}| + |x_{i'}^{*(\ell)} - c^{*(\ell)}| \leq \epsilon^* + r^*.$$

Applying the above inequality, we have that

$$(C.3) \quad \|y[\infty] - c^*\|^2 = \sum_{\ell \in [d]} |y^{(\ell)}[\infty] - c^{*(\ell)}|^2 \leq d(r^* + \epsilon^*)^2.$$

Consider a regular agent $v_i \in \mathcal{V}_{\mathcal{R}}$. Using inequality (C.3) and the definition of c^* and r^* (defined in subsection 5.1), we obtain that

$$\|x_i^* - y[\infty]\| \leq \|x_i^* - c^*\| + \|c^* - y[\infty]\| \leq \sqrt{d}(r^* + \epsilon^*) + r^*.$$

The result follows from noting that $x_c = y[\infty]$ for SDMMFD and SDFD.

Now, consider the case where the regular agents follow CWTM [35, 33, 34, 9, 44, 7] or RVO [28, 1]. Since in this case, $x_c = c^*$, the result directly follows from the definition of c^* and r^* . \square

Acknowledgments. We would like to thank Denpoom Rangsoi for his invaluable help in revising and optimizing the code, which greatly improved performance. We also extend our gratitude to Rapeepong Reangreab for executing the code and providing useful implementation suggestions.

REFERENCES

- [1] W. ABBAS, M. SHABIR, J. LI, AND X. KOUTSOUKOS, *Resilient distributed vector consensus using centerpoint*, Automatica J. IFAC, 136 (2022), 110046, <https://doi.org/10.1016/j.automatica.2021.110046>.
- [2] W. BEN-AMEUR, P. BIANCHI, AND J. JAKUBOWICZ, *Robust distributed consensus using total variation*, IEEE Trans. Automat. Control, 61 (2016), pp. 1550–1564, <https://doi.org/10.1109/TAC.2015.2471755>.

- [3] M. CAO, A. S. MORSE, AND B. D. O. ANDERSON, *Reaching a consensus in a dynamically changing environment: A graphical approach*, SIAM J. Control Optim., 47 (2008), pp. 575–600, <https://doi.org/10.1137/060657005>.
- [4] D. CASTANO, V. E. PAKSOY, AND F. ZHANG, *Angles, triangle inequalities, correlation matrices and metric-preserving and subadditive functions*, Linear Algebra Appl., 491 (2016), pp. 15–29, <https://doi.org/10.1016/j.laa.2014.10.011>.
- [5] J. C. DUCHI, A. AGARWAL, AND M. J. WAINWRIGHT, *Dual averaging for distributed optimization: Convergence analysis and network scaling*, IEEE Trans. Automat. Control, 57 (2012), pp. 592–606, <https://doi.org/10.1109/TAC.2011.2161027>.
- [6] A. R. ELKORDY, S. PRAKASH, AND S. AVESTIMEHR, *Basil: A fast and Byzantine-resilient approach for decentralized training*, IEEE J. Sel. Area. Comm., 40 (2022), pp. 2694–2716, <https://doi.org/10.1109/JSAC.2022.3191347>.
- [7] C. FANG, Z. YANG, AND W. U. BAJWA, *BRIDGE: Byzantine-resilient decentralized gradient descent*, IEEE Trans. Signal Inform. Process. Netw., 8 (2022), pp. 610–626, <https://doi.org/10.1109/TSIPN.2022.3188456>.
- [8] S. FARHADKHANI, R. GUERRAOUI, N. GUPTA, R. PINOT, AND J. STEPHAN, *Byzantine machine learning made easy by resilient averaging of momentums*, Proc. Mach. Learn. Res. (PMLR), 162 (2022), pp. 6246–6283, <https://proceedings.mlr.press/v162/farhadkhani22a.html>.
- [9] W. FU, Q. MA, J. QIN, AND Y. KANG, *Resilient consensus-based distributed optimization under deception attacks*, Internat. J. Robust Nonlinear Control, 31 (2021), pp. 1803–1816, <https://doi.org/10.1002/rnc.5026>.
- [10] S. GUO, T. ZHANG, H. YU, X. XIE, L. MA, T. XIANG, AND Y. LIU, *Byzantine-resilient decentralized stochastic gradient descent*, IEEE Trans. Circuits. Syst. Video Technol., 32 (2022), pp. 4096–4106, <https://doi.org/10.1109/TCSVT.2021.3116976>.
- [11] N. GUPTA, T. T. DOAN, AND N. H. VAIDYA, *Byzantine fault-tolerance in decentralized optimization under 2f-redundancy*, in 2021 American Control Conference (ACC), IEEE, Piscataway, NJ, 2021, pp. 3632–3637, <https://doi.org/10.23919/ACC50511.2021.9483067>.
- [12] D. H. GUTMAN AND J. F. PENA, *The condition number of a function relative to a set*, Math. Program., 188 (2021), pp. 255–294, <https://doi.org/10.1007/s10107-020-01510-4>.
- [13] L. HE, S. P. KARIMIREDDY, AND M. JAGGI, *Byzantine-Robust Decentralized Learning via ClippedGossip*, preprint, <https://arxiv.org/abs/2202.01545>, 2023.
- [14] H. HENDRIKX, F. BACH, AND L. MASSOULIE, *Accelerated decentralized optimization with local updates for smooth and strongly convex objectives*, Proc. Mach. Learn. Res. (PMLR), 89 (2019), pp. 897–906, <https://proceedings.mlr.press/v89/hendrikx19a.html>.
- [15] D. KOVALEV, A. KOLOSKOVA, M. JAGGI, P. RICHTARIK, AND S. STICH, *A linearly convergent algorithm for decentralized optimization: Sending less bits for free!*, Proc. Mach. Learn. Res. (PMLR), 130 (2021), pp. 4087–4095, <https://proceedings.mlr.press/v130/kovalev21a.html>.
- [16] K. KUWARANANCHAROEN AND S. SUNDARAM, *On the location of the minimizer of the sum of two strongly convex functions*, in 2018 IEEE Conference on Decision and Control (CDC), IEEE, Piscataway, NJ, 2018, pp. 1769–1774, <https://doi.org/10.1109/CDC.2018.8619735>.
- [17] K. KUWARANANCHAROEN AND S. SUNDARAM, *On the set of possible minimizers of a sum of known and unknown functions*, in 2020 American Control Conference (ACC), IEEE, Piscataway, NJ, 2020, pp. 106–111, <https://doi.org/10.23919/ACC45564.2020.9147407>.
- [18] K. KUWARANANCHAROEN AND S. SUNDARAM, *On the Geometric Convergence of Byzantine-Resilient Distributed Optimization Algorithms*, preprint, <https://arxiv.org/abs/2305.10810>, 2023.
- [19] K. KUWARANANCHAROEN AND S. SUNDARAM, *The minimizer of the sum of two strongly convex functions*, Optimization, to appear, <https://doi.org/10.1080/02331934.2024.2402923>.
- [20] K. KUWARANANCHAROEN, L. XIN, AND S. SUNDARAM, *Byzantine-resilient distributed optimization of multi-dimensional functions*, in 2020 American Control Conference (ACC), IEEE, Piscataway, NJ, 2020, pp. 4399–4404, <https://doi.org/10.23919/ACC45564.2020.9147396>.
- [21] K. KUWARANANCHAROEN, L. XIN, AND S. SUNDARAM, *Scalable distributed optimization of multi-dimensional functions despite Byzantine adversaries*, IEEE Trans. Signal Inform. Process. Netw., 10 (2024), pp. 360–375, <https://doi.org/10.1109/TSIPN.2024.3379844>.
- [22] H. J. LEBLANC, H. ZHANG, X. KOUTSOUKOS, AND S. SUNDARAM, *Resilient asymptotic consensus in robust networks*, IEEE J. Sel. Area. Comm., 31 (2013), pp. 766–781, <https://doi.org/10.1109/JSAC.2013.130413>.
- [23] N. A. LYNCH, *Distributed Algorithms*, Morgan Kaufmann Publishers, San Francisco, CA, 1996, <https://dl.acm.org/doi/book/10.5555/2821576>.
- [24] A. NEDIĆ, A. OLSHEVSKY, AND W. SHI, *Achieving geometric convergence for distributed optimization over time-varying graphs*, SIAM J. Optim., 27 (2017), pp. 2597–2633, <https://doi.org/10.1137/16M1084316>.

- [25] A. NEDIĆ AND A. OZDAGLAR, *Distributed subgradient methods for multi-agent optimization*, IEEE Trans. Automat. Control, 54 (2009), pp. 48–61, <https://doi.org/10.1109/TAC.2008.2009515>.
- [26] A. NEDIĆ AND A. OLSHEVSKY, *Distributed optimization over time-varying directed graphs*, IEEE Trans. Automat. Control, 60 (2015), pp. 601–615, <https://doi.org/10.1109/TAC.2014.2364096>.
- [27] A. NEDIĆ, A. OLSHEVSKY, AND M. G. RABBAT, *Network topology and communication-computation tradeoffs in decentralized optimization*, Proc. IEEE, 106 (2018), pp. 953–976, <https://doi.org/10.1109/JPROC.2018.2817461>.
- [28] H. PARK AND S. A. HUTCHINSON, *Fault-tolerant rendezvous of multirobot systems*, IEEE Trans. Robot., 33 (2017), pp. 565–582, <https://doi.org/10.1109/TRO.2017.2658604>.
- [29] J. PENG, W. LI, AND Q. LING, *Byzantine-robust decentralized stochastic optimization over static and time-varying networks*, Signal Process., 183 (2021), 108020, <https://doi.org/10.1016/j.sigpro.2021.108020>.
- [30] S. PU, W. SHI, J. XU, AND A. NEDIĆ, *Push–pull gradient methods for distributed optimization in networks*, IEEE Trans. Automat. Control, 66 (2021), pp. 1–16, <https://doi.org/10.1109/TAC.2020.2972824>.
- [31] N. RAVI AND A. SCAGLIONE, *Detection and isolation of adversaries in decentralized optimization for non-strongly convex objectives*, IFAC-PapersOnLine, 52 (2019), pp. 381–386, <https://doi.org/10.1016/j.ifacol.2019.12.185>.
- [32] W. SHI, Q. LING, K. YUAN, G. WU, AND W. YIN, *On the linear convergence of the ADMM in decentralized consensus optimization*, IEEE Trans. Signal Process., 62 (2014), pp. 1750–1761, <https://doi.org/10.1109/TSP.2014.2304432>.
- [33] L. SU AND N. VAIDYA, *Byzantine Multi-agent Optimization: Part I*, preprint, <https://arxiv.org/abs/1506.04681>, 2015.
- [34] L. SU AND N. H. VAIDYA, *Byzantine-resilient multiagent optimization*, IEEE Trans. Automat. Control, 66 (2021), pp. 2227–2233, <https://doi.org/10.1109/TAC.2020.3008139>.
- [35] S. SUNDARAM AND B. GHARESIFARD, *Distributed optimization under adversarial nodes*, IEEE Trans. Automat. Control, 64 (2019), pp. 1063–1076, <https://doi.org/10.1109/TAC.2018.2836919>.
- [36] Z. WU, T. CHEN, AND Q. LING, *Byzantine-resilient decentralized stochastic optimization with robust aggregation rules*, IEEE Trans. Signal Process., 71 (2023), pp. 3179–3195, <https://doi.org/10.1109/TSP.2023.3300629>.
- [37] R. XIN, S. PU, A. NEDIĆ, AND U. A. KHAN, *A general framework for decentralized optimization with first-order methods*, Proc. IEEE, 108 (2020), pp. 1869–1889, <https://doi.org/10.1109/JPROC.2020.3024266>.
- [38] 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*, Annu. Rev. Control, 47 (2019), pp. 278–305, <https://doi.org/10.1016/j.arcontrol.2019.05.006>.
- [39] Z. YANG AND W. U. BAJWA, *ByRDIE: Byzantine-resilient distributed coordinate descent for decentralized learning*, IEEE Trans. Signal Inform. Process. Netw., 5 (2019), pp. 611–627, <https://doi.org/10.1109/TSIPN.2019.2928176>.
- [40] 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 Process. Mag., 37 (2020), pp. 146–159, <https://doi.org/10.1109/MSP.2020.2973345>.
- [41] D. YIN, Y. CHEN, R. KANNAN, AND P. BARTLETT, *Byzantine-robust distributed learning: Towards optimal statistical rates*, Proc. Mach. Learn. Res. (PMLR), 80 (2018), pp. 5650–5659, <https://proceedings.mlr.press/v80/yin18a.html>.
- [42] K. YUAN, Q. LING, AND W. YIN, *On the convergence of decentralized gradient descent*, SIAM J. Optim., 26 (2016), pp. 1835–1854, <https://doi.org/10.1137/130943170>.
- [43] M. ZAMANI, F. GLINEUR, AND J. M. HENDRICKX, *On the set of possible minimizers of a sum of convex functions*, IEEE Control Syst. Lett., 8 (2024), pp. 1871–1876, <https://doi.org/10.1109/LCSYS.2024.3414378>.
- [44] C. ZHAO, J. HE, AND Q.-G. WANG, *Resilient distributed optimization algorithm against adversarial attacks*, IEEE Trans. Automat. Control, 65 (2020), pp. 4308–4315, <https://doi.org/10.1109/TAC.2019.2954363>.
- [45] X. ZHOU, *On the Fenchel Duality Between Strong Convexity and Lipschitz Continuous Gradient*, preprint, <https://arxiv.org/abs/1803.06573>, 2018.