Robotics: Science and Systems 2020
Corvalis, Oregon, USA, July 12-16, 2020

Resilient Distributed Diffusion for Multi-Robot Systems Using Centerpoint

Jiani Li*, Waseem Abbas*, Mudassir Shabbir†, Xenofon Koutsoukos*
*Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, USA
{jiani.li, waseem.abbas, xenofon.koutsoukos}@vanderbilt.edu
†Computer Science Department, Information Technology University, Lahore, Pakistan
mudassir@rutgers.edu

***Abstract*—In this paper, we study the resilient diffusion problem in a network of robots aiming to perform a task by optimizing a global cost function in a cooperative manner. In distributed diffusion, robots combine the information collected from their local neighbors and incorporate this aggregated information to update their states. If some robots are adversarial, this cooperation can disrupt the convergence of robots to the desired state. We propose a resilient aggregation rule based on the notion of *centerpoint*, which is a generalization of the median in the higher dimensional Euclidean space. Robots exchange their d-dimensional state vectors with neighbors. We show that if a normal robot implements the centerpoint-based aggregation rule and has n neighbors, of which at most $\lceil \frac{n}{d+1} \rceil - 1$ are adversarial, then the aggregated state always lies in the convex hull of the states of the normal neighbors of the robot. Consequently, all normal robots implementing the distributed diffusion algorithm converge resiliently to the true target state. We also show that commonly used aggregation rules based on the coordinate-wise median and geometric median are, in fact, not resilient to certain attacks. We numerically evaluate our results on mobile multi-robot networks and demonstrate the cases where diffusion with the weighted average, coordinate-wise median, and geometric median-based aggregation rules fail to converge to the true target state, whereas diffusion with the centerpoint-based rule is resilient in the same scenario.**

***Index Terms*—Resilient distributed learning and optimization, resilient aggregation, centerpoint**

I. Introduction

Diffusion strategies enable adaptation and learning over networks in a distributed manner by exploiting cooperation among agents that seek to optimize some global cost function through local interactions. Distributed diffusion usually involves an *adaptation* step in which an agent learns from its own data, and a *combination* step in which agent aggregates the information collected from its neighbors. This adaptation-combination diffusion strategy has proven to be quite effective in solving distributed optimization problems in various network applications [1]. In multi-robot systems, distributed diffusion is useful for problems, such as target localization and tracking [2], distributed clustering [3], distributed sensing and estimation [4], and biologically inspired designs for mobile networks [5].

Although the cooperation among agents helps improve the overall learning performance, the aggregation step in distributed diffusion is susceptible to attacks where non-cooperative or adversarial neighbors sharing wrong information can disrupt the convergence of the algorithm. It has been shown (for both fixed and adaptive weights) that a single misbehaving agent can adversely impact the convergence of remaining agents to the desired target state [6], [7]. Therefore, it is crucial to design a resilient aggregation rule for distributed diffusion.

Recent studies regarding the resilience of the distributed learning algorithms [8]–[12] demonstrate the success of median-based aggregation rules, such as the coordinate-wise median and geometric median, to Byzantine adversaries. In this paper, we show that such aggregation rules are not resilient under certain conditions, especially when the robots' state vectors are d-dimensional, where $d \geq 2$. The main reason is the inability of these rules to guarantee that the aggregated state lies in the convex hull of the normal neighbors' states, which is crucial for the convergence of distributed diffusion and consensus algorithms [1], [13]. For instance, in the coordinate-wise median approach, resilient aggregation based on the median is applied separately for each coordinate $j \in \{1, 2, \cdots, d\}$ of the state. This guarantees that the value at the j^{th} coordinate of the aggregated state lies between the minimum and the maximum values at the j^{th} coordinate of the states of the normal neighbors. However, it does not ensure that the aggregated state (in $\mathbb{R}^d$) necessarily lies in the convex hull of the normal neighbors' states, as we discuss in Section V.

To address this issue, we propose a resilient aggregation rule based on centerpoint, which extends the notion of the median in higher dimensions [14], [15]. A normal robot having n neighbors gathers their state vectors (each of which is in $\mathbb{R}^d$) and then computes a centerpoint of these n points in $\mathbb{R}^d$. If the number of adversaries in the neighborhood of a normal robot is at most $\lceil \frac{n}{d+1} \rceil - 1$, then a centerpoint lies in the convex hull of points corresponding to the state values of normal neighbors. Consequently, this property guarantees that normal robots implementing distributed diffusion converge to the true target state even in the presence of non-cooperating and adversarial robots. Our main contributions are:

- We propose an aggregation rule based on centerpoint for distributed diffusion that guarantees the convergence of the algorithm to the true model given the number of adversarial robots in the neighborhood of a normal robot is limited to $\lceil \frac{n}{d+1} \rceil - 1$. Here, n is the size of the neighborhood, and d is the dimension of the state vector of the robots.
- We analyze the resilience and performance in terms of steady-state mean-square-deviation (MSD) of the centerpoint-based distributed diffusion. We also discuss the time complexity of computing a centerpoint.

- We numerically evaluate our results on a mobile adaptive multi-robot network and compare the proposed centerpoint-based aggregation rule with the weighted average, coordinate-wise median, and geometric median-based rules. The simulation results show that our approach is resilient to $\lceil \frac{n}{d+1} \rceil - 1$ Byzantine robots in the neighborhood, while the other approaches are not resilient in the same scenarios.

The rest of the paper is organized as follows: Section II discusses the related work. Section III introduces the distributed adapt-then-combine diffusion algorithm. Section IV formulates the resilient distributed diffusion problem. Section V discusses the resilience of coordinate-wise and geometric median-based aggregation rules. Section VI introduces the centerpoint-based aggregation rule, and analyzes the resilient distributed diffusion with centerpoint-based aggregation. Section VII gives a numerical evaluation of the results. Finally, Section VIII concludes the paper.

II. Related Work

The resilient distributed consensus problem is widely studied in the robotics and control systems community and is very relevant to the resilient distributed diffusion. The main goal in resilient distributed consensus is to ensure that all normal agents in the network, including malicious and non-cooperative agents, reach an agreement over the values of the state variable in a distributed manner. The normal agents' states throughout the process, as well as the consensus state, must lie in the convex hull of their initial states, a condition commonly referred to as the *safety* condition. For scalar states, the Weighted-Mean Subsequence Reduced (WMSR) algorithms [16] and the median-based algorithms [17] guarantees the resilient convergence in the presence of adversaries under certain robustness conditions on the underlying network graph. Different variations of W-MSR algorithm have also been proposed [18], [19]. The resilient consensus problem is more challenging when the state vector is in $\mathbb{R}^d$ where $d \geq 2$. Although variants of the W-MSR algorithm has been applied to mobile multi-robot systems for formation control [20] and flocking [21] applications, such methods cannot guarantee the resilient vector consensus in fact as we discuss in Section V. To achieve resilient vector consensus, Tverberg partition [22]–[25] and centerpoint-based [26] approaches have been proposed. The approximate Tverberg point-based approach has also been successfully applied to the resilient multi-robots rendezvous problem [13].

Another line of related work is the Byzantine resilient aggregation problems for distributed learning and optimization. To achieve resilient optimization, one approach is to discard cooperation with possible Byzantine neighbors. In [7], a resilient diffusion algorithm has been proposed in which normal agents discard information from a certain number of neighbors, which might include Byzantine agents, in the aggregation step. However, the performance of the algorithm depends highly on the accurate estimation of the number of adversarial agents, which is usually unknown. A similar screening algorithm called Zeno has been proposed in [27] that ranks the scores of the aggregated gradients as the measurements of their trustworthiness and discards b largest scores to achieve resilient aggregation. Moreover, various majority-based aggregation rules have been used that preclude states far away from the cluster of the normal agents' states to achieve resilient distributed optimization. Well-known majority-based aggregation rules include coordinate-wise median [10], geometric median [9], [11], coordinate-wise trimmed mean [10] and Krum [6]. However, studies have already reported some of these rules are not resilient to attacks under certain conditions [28]–[30].

One can also use computation redundancy [31]–[33] to achieve resilient convergence for distributed learning, which typically involves coding theory and algorithmic redundancy. Yet this method is limited to the master-slave network where one parameter server sends a global model to each node, which then computes the gradient based on the available data, and the parameter server then aggregates gradients from all nodes to update the global model. An example of such a framework is DRACO [31] in which the parameter server uses redundant gradients received from nodes to eliminate the effects of adversarial updates. Another algorithm proposed recently is the RSA [34] that introduces an ℓ_p-norm regularization term into the objective function for resilience purpose. It eliminates the effect caused by the magnitudes of malicious messages sent by the Byzantine nodes. As a result, only the number of Byzantine nodes influence the model update, thus making it robust to arbitrary attacks from Byzantine nodes. However, it is not straightforward to apply this approach to a fully distributed network.

III. Preliminaries – distributed adapt-then-combine diffusion algorithm

In this paper, we use boldface notation to represent random variables and normal font to represent deterministic quantities. Notation $[N] := \{1, 2, \dots, N\}$, and $\|\cdot\|$ represents ℓ_2 norm. The symbol $*$ denotes complex conjugation for scalars and complex-conjugate transposition for matrices. $\mathbb{1}$ denotes the $N \times 1$ vector with all entries equal to one.

Consider a network of agents[1] modeled by a *directed graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents agents and $\mathcal{E}$ represents interactions between agents. An edge (l, k) means that agent $k \in \mathcal{V}$ can exchange information with j. Each agent should has its own information, such that $(k, k) \in \mathcal{E}, \forall k \in \mathcal{V}$. The *neighborhood* of k is the set of nodes $\mathcal{N}_k = \{l \in \mathcal{V} | (l, k) \in \mathcal{E}\}$. At each iteration i, agent k has access to a scalar random measurement $\boldsymbol{d}_k(i)$ and an i.i.d regression vector $\boldsymbol{u}_{k,i}$ of size d with zero-mean and uniform covariance matrix $R_{u,k} \triangleq \mathbb{E}\{\boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i}\} > 0$, which are related via a linear model of the following form:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i). \quad (1)$$

Here, $\boldsymbol{v}_k(i)$ represents a zero-mean i.i.d. additive noise with variance $\sigma_{v,k}^2$ and w^o denotes the unknown d-dimensional model vector that agent k attempts to estimate.

[1] We use the term *agent* and *robot* interchangeably.

The objective of each agent is to estimate w^o that minimizes a global objective function of the following form:

$$\min_{w} \left\{ J^{glob}(w) := \frac{1}{N} \sum_{k=1}^{N} J_k(w) \right\}, \tag{2}$$

where

$$J_k(w) := \mathbb{E}\left\{ \|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w\|^2 \right\}. \tag{3}$$

Stochastic gradient descent (SGD) can be used to solve the objective function (2), where each agent k computes successive estimates of w^o as follows:

$$\begin{aligned} \boldsymbol{w}_{k,i} &= \boldsymbol{w}_{k,i-1} - \mu \nabla_{w_k} J_k(\boldsymbol{w}_{k,i-1}) \\ &= \boldsymbol{w}_{k,i-1} + \mu \boldsymbol{u}_{k,i}^{*} [\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{k,i-1}], \end{aligned}$$

where $\mu > 0$ is the step size.

Instead, the minimization function (2) can also be solved in a distributed and cooperative manner using diffusion strategies. The diffusion strategies introduce an aggregation step that incorporates information gathered from the neighboring agents into the optimization procedure. One powerful diffusion scheme is adapt-then-combine (ATC) [1] which optimizes the solution using the following update:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu \boldsymbol{u}_{k,i}^{*} [\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{k,i-1}] \quad \text{(adaptation)} \tag{4}$$

$$\boldsymbol{w}_{k,i} = \sum_{l \in \mathcal{N}_k} a_{lk} \boldsymbol{\psi}_{l,i}, \quad \text{(combination)} \tag{5}$$

where a_{lk} represents the weight assigned to agent l from agent k that is used to scale the data it receives from l. The weights satisfy the following constraints:

$$a_{lk} \geq 0, \qquad \sum_{l \in \mathcal{N}_k} a_{lk} = 1, \qquad a_{lk} = 0 \text{ if } l \notin \mathcal{N}_k. \tag{6}$$

Well-known weighted average-based aggregation rules include the uniform ($a_{lk} = \frac{1}{|\mathcal{N}_k|}$), maximum-degree ($a_{lk} = \frac{1}{N}$), relative-variance ($a_{lk} = \frac{\sigma_{v,l}^{-2}}{\sum_{m \in \mathcal{N}_k} \sigma_{v,m}^{-2}}$) among others [1], [35].

Weighted average-based diffusion algorithm outperforms the non-cooperative SGD as measured by the steady-state mean-square-deviation (MSD) performance [1]. For sufficiently small step-size, the network performance of non-cooperative SGD is defined as the averaged steady-state MSD among agents and can be approximated by

$$\text{MSD}_{\text{ncop}} \triangleq \lim_{i \to \infty} \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \approx \frac{\mu d}{2} \cdot \left(\frac{1}{N} \sum_{k=1}^{N} \sigma_{v,k}^2\right), \tag{7}$$

where $\tilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i}$. And for distributed diffusion, the steady-state MSD performance of each normal agent is approximately equal to the network MSD and for sufficiently small step-size is given by

$$\text{MSD}_{\text{diff},k} \approx \text{MSD}_{\text{diff,net}} = \frac{\mu d}{2} \cdot \left(\sum_{k \in \mathcal{N}_k} p_k^2 \sigma_{v,k}^2 \right), \tag{8}$$

where $A = \{a_{lk}, k = 1, 2, \ldots, N\}$ is the $N \times N$ left-stochastic combination matrix and $p = \{p_k, k = 1, 2, \ldots, N\}$ denotes the right eigenvector of A that is associated with the eigenvalue at one and satisfies $Ap = p, p^\top \mathbb{1} = 1, 0 < p_k < 1$. Consider the case where A is doubly stochastic for a standard diffusion network such that $A^\top \mathbb{1} = A\mathbb{1} = \mathbb{1}$. Then, the right eigenvector $p_k = \frac{1}{N}$ and $p_k^2 = \frac{1}{N^2}$. This means the effect of diffusion cooperation is to scale the noise variances by the factors $\frac{1}{N^2}$, whereas the non-cooperative SGD has the effect of scaling the noise variances by the factor $\frac{1}{N}$, which demonstrates an N-fold improvement of MSD performance.

IV. Problem Formulation

In the aggregation (combination) step of the diffusion algorithm, an agent gathers and combines the estimates of its neighbors, some of which could possibly be malicious. We consider the resilient aggregation problem in distributed diffusion in the presence of Byzantine agents. We assume two types of agents in the network, normal and Byzantine. *Normal* agents are the ones that interact with their neighbors synchronously and always update their states (estimates) according to a prescribed update rule, that is the diffusion algorithm. *Byzantine* agents are the ones that can change their states arbitrarily and do not follow the prescribed update rule. Moreover, a Byzantine agent can transmit different values to its different neighbors. For a normal agent k, all agents in its neighborhood are indistinguishable, that is, k cannot identify which of its neighbors are Byzantine. The goal of each normal agent is to achieve *resilient convergence* formally stated below.

Definition 1. (Resilient convergence) *Distributed diffusion is said to be resilient if*

$$\lim_{i \to \infty} \boldsymbol{w}_{k,i} = w^o \tag{9}$$

for every normal agent k in the network, thereby ensuring that all normal agents converge to the true model.

For aggregation, a simple and most widely used strategy in diffusion algorithms is to compute a *weighted average* of the neighbors' states. However, a single Byzantine agent can drive the output of the weighted average-based aggregation to an arbitrary value as shown in [6, Lemma 1], and hence, prevent the normal agent from converging to the target state. Moreover, in case of adaptive weights, time-dependent Byzantine attack proposed in [7] can disrupt the convergence of normal nodes.

In [1], it is shown that if there is no Byzantine agent in the network, and each normal agent computes a convex combination of its neighbors' states in the aggregation step, then all normal agents implementing diffusion algorithm ((4) and (5)) eventually converge to the true target state. In other words, in the aggregation step if each agent computes a state that is in the *convex hull*[2] of the states of its (normal) neighbors, then convergence is guaranteed.

[2] The convex hull of a set of points $\mathcal{P} = \{w_1, w_2, \ldots, w_n\}$ in $\mathbb{R}^d$ is the smallest convex set containing $\mathcal{P}$. Any point w_{in} inside the convex hull of $\mathcal{P}$ has the property that $w_{in} = \sum_{k=1}^{n} \lambda_k w_k$, where $0 \leq \lambda_k \leq 1$ and $\sum_{k=1}^{n} \lambda_k = 1$. And no point outside of the convex hull has such representation.

Since normal agents cannot easily detect Byzantine agents, the objective is to

- design an aggregation rule for a normal agent, which cannot distinguish between normal and Byzantine neighbors, such that the aggregated result by the agent is inside the convex hull of its normal neighbors' states, and
- show that the diffusion algorithm using this aggregation rule achieves resilient convergence as defined above.

V. Coordinate-wise Median and Geometric Median based Aggregation Rules

Resilient or robust aggregation has been a hot topic in the field of distributed learning in recent years [8]–[12]. In this section, we discuss two widely used majority-based aggregation rules, that is, coordinate-wise median (also known as marginal median) [10], [36], and geometric median [9], [11]. We illustrate these rules are not resilient to Byzantine attacks in certain scenarios as they cannot guarantee the aggregation result to be inside the convex hull of normal states. First, we define these notions.

Definition 2. (Coordinate-wise Median (CM)) *Let* $\text{med}(\cdot)$ *to be the one-dimensional median, then the coordinate-wise median* $\text{Median}(\cdot)$ *of vectors* $x_k \in \mathbb{R}^d, k \in [n]$ *is defined to be* $x^{\text{Med}} := \text{Median}\{x_k : k \in [n]\}$ *with the* j*-th coordinate to be* $(x^{\text{Med}})^j := \text{med}\{x_k^j : k \in [n]\}$ *for each* $j \in [d]$.

Definition 3. (Geometric median (GM)) *The geometric median* $\text{GM}(\cdot)$ *of vectors* $x_k \in \mathbb{R}^d, k \in [n]$ *is defined to be* $\text{GM}\{x_k, k \in [n]\} := \arg\min_{x\in\mathbb{R}^d} \sum_{k=1}^n \|x - x_k\|$.

In one dimension, the median has a robustness property, that is, if more than $\lceil \frac{n}{2} \rceil$ points are in $[-r, r]$ for some $r \in \mathbb{R}$, then the median must also be in $[-r, r]$ irrespective of the location of other points. This guarantees that as long as majority of the points are benign, the median is sure to be within the range of benign points. However, in multiple dimensions, CM of points in $\mathbb{R}^d$ only guarantees that the value at the j^{th} coordinate of CM is between the minimum and maximum values at the j^{th} coordinate of benign points, which does not ensure that CM is necessarily in the convex hull of benign points. Similarly, GM is not guaranteed to be inside the convex hull of benign points even if they are in the majority. We illustrate this through an example in Figure 1, where we have a total of 10 points (in a plane), of which 3 are Byzantine and 7 are normal. Byzantine points lie far from the normal points. We then compute the CM and GM of all the 10 points, and observe that both CM and GM fall outside the convex hull of normal points.

Thus, for CM and GM based diffusion, Byzantine agents can send values much smaller (or greater) than the normal values in each coordinate, and in the worst case, the aggregation result may fall outside the convex hull of normal points even when normal agents are in the majority. When this happens, at the next iteration of the diffusion (step (4)), the estimated model will be adapted on the basis of this wrong aggregation result. As time accumulates, this may disturb the convergence of normal agents and lead them to converge to a different state.

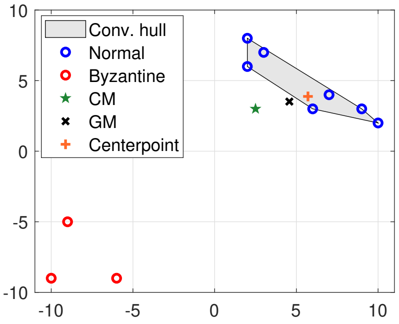


Fig. 1: Aggregating 10 points using different aggregation rules.

In Section VII, we will further show cases where distributed diffusion with such aggregation rules fail to converge to the true state.

VI. Resilient Diffusion by Centerpoint-based Aggregation

In this section, we explain the notion of a *centerpoint* and analyze the resilient diffusion using the centerpoint-based aggregation rule.

A. Centerpoint and Its Resilient Property

Definition 4. (Centerpoint) *Given a set* S *of* n *points in* $\mathbb{R}^d$ *in general positions*[3] *where* $n \geq d+1$, *a centerpoint* p *is a point, not necessarily from* S, *such that any closed half-space*[4] *of* $\mathbb{R}^d$ *that contains* p *also contains at least* $\lceil \frac{n}{d+1} \rceil$ *points from* S.

Intuitively, a centerpoint lies in the "center region" of the point cloud, in the sense that there are enough points of S on each side of a centerpoint. A centerpoint extends the notion of median to higher dimensions, and is an active topic of study in discrete geometry [15], [37]. For any given set of points S, the existence of centerpoint is guaranteed by the famous Centerpoint Theorem (see [14], [38]).

Theorem 1. (Centerpoint Theorem) *For any given point set in general positions in an arbitrary dimension, a centerpoint always exists.*

A centerpoint is not unique, in fact, there can be infinitely many centerpoints. The set of all centerpoints constitutes the *centerpoint region* or simply the *center region*, which is known to be closed and convex. We observe that if a normal agent has at most $\lceil \frac{n}{d+1} \rceil - 1$ Byzantine agents in its neighborhood, and the agent aggregates its neighbors' states by computing their centerpoint, then the aggregated state essentially lies in the convex hull of the states of normal neighbors (See Figure 1 for an example). We formalize this in Lemma 1.

Lemma 1. *Given a set of* n *agents with* d*-dimensional state vectors, of which any of the* F *agents can be Byzantine, then*

[3] A set of points in $\mathbb{R}^d$ is said to be in *general positions* if no hyperplane of dimension $d-1$ or less contains more than d points.

[4] Recall that closed half-space in $\mathbb{R}^d$ is a set of the form $\{x \in \mathbb{R}^d : a^T x \geq b\}$ for some $a \in \mathbb{R}^d \setminus \{0\}$.

a centerpoint of the n state vectors always lies in the convex hull of the state vectors of the normal agents if and only if $F \leq \lceil \frac{n}{d+1} \rceil - 1$.

Proof. For succinctness, we use the term *Byzantine points* to denote the state vectors of Byzantine agents, and *normal points* to denote the state vectors of normal agents. Let $\mathcal{C}_F$ be the convex hull of the F Byzantine points, and $\mathcal{C}_N$ be the convex hull of the normal points.

($\Rightarrow$) Assume that a centerpoint c is not in $\mathcal{C}_N$, then there exists a half-space containing c and all the points in $\mathcal{C}_N$, and hence containing all the normal points. Since there are at least $\lceil \frac{n}{d+1} \rceil + 1$ normal points, the other half-space contains less than $\lceil \frac{n}{d+1} \rceil$ points, which is not possible by the definition of centerpoint. Therefore, c must lie in $\mathcal{C}_N$.

($\Leftarrow$) Assume $F > \lceil \frac{n}{d+1} \rceil - 1$, then there exists a set of F points outside $\mathcal{C}_N$ such that a centerpoint $c \notin \mathcal{C}_N$. To see this, consider a set of $d+1$ points arranged in a d-simplex at a unit distance from each other. Now equally place the remaining $n-d-1$ points in $d+1$ tiny balls of radius ϵ near each vertex of the simplex such that each ball contains $\frac{n}{d+1}$ points.[5] For sufficiently small value ϵ, the interior of this simplex is the centerpoint region because every hyperplane that passes through the interior of simplex contains at least one vertex (and all the points placed there) on either side. If all $\frac{n}{d+1}$ points near any one of the simplex vertices are Byzantine, then $\mathcal{C}_N$ does not contain any point from the interior of the simplex. Thus, $\mathcal{C}_N$ doesn't contain any centerpoint in this case, and the claim follows.

□

B. Diffusion with Centerpoint-based Aggregation

Here, we apply the centerpoint-based aggregation into the diffusion algorithm by replacing the weighted average-based aggregation step (equation (5)) of distributed diffusion by

$$\boldsymbol{w}_{k,i} = \mathcal{CP}\{\boldsymbol{\psi}_{l,i} : l \in [\mathcal{N}_k]\},$$

where $\mathcal{CP}$ computes a centerpoint of a set of vectors $\boldsymbol{\psi}_{l,i} : l \in [\mathcal{N}_k]$. The centerpoint-based distributed diffusion algorithm is summarized in **Algorithm 1**.

```
Algorithm 1: Centerpoint-based distributed diffusion
Input: $\mu$, $\boldsymbol{w}_{k,-1}$
for $i > 0$ and for every normal robot $k$ do
    $\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \nabla_{w_k} J_k(\boldsymbol{w}_{k,i-1})$
    send $\boldsymbol{\psi}_{k,i}$ to $l \in \mathcal{N}_k$ and receive $\boldsymbol{\psi}_{l,i}$ from $l \in \mathcal{N}_k$
    $\boldsymbol{w}_{k,i} = \mathcal{CP}\{\boldsymbol{\psi}_{l,i} : l \in [\mathcal{N}_k]\}$
```

Next, we analyze the resilient convergence and the steady-state estimation performance of the proposed algorithm.

C. Convergence Analysis

In order to analyze the convergence of the proposed algorithm, we assume the cost function $J_k : \mathbb{R}^d \to \mathbb{R}$ is a differentiable convex function and has unique global minimum $w^o \in \mathbb{R}^d$. We also assume J_k to have an L-Lipschitz continuous gradient formally defined below.

Definition 5. (L-Lipschitz continuous gradient) *A differentiable convex function f is said to have an Lipschitz continuous gradient, if there exists a constant $L > 0$, such that*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \forall x, y.$$

If f has an L-Lipschitz continuous gradient, then it has the following property which is referred to as "co-coercivity":

$$(\nabla f(x) - \nabla f(y))^\top (x - y) \geq \frac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2, \forall x, y.$$

Given that the number of Byzantine neighbors is less than $\lceil \frac{|\mathcal{N}_k|}{d+1} \rceil$ for each agent k, where $|\mathcal{N}_k|$ is the size of $\mathcal{N}_k$, we know by Lemma 1 that the result of the centerpoint-based aggregation for each agent will be in the convex hull of points corresponding to the normal robots' states. Based on this fact as well as the above assumptions about the cost function, the centerpoint-based distributed diffusion algorithm will converge to the actual model, thus achieving the resilient convergence objective in (9). We formalize this below.

Proposition 1. *All the normal robots converge to the true model w^o by the centerpoint-based distributed diffusion in **Algorithm 1** with step size $\mu \in (0, \frac{2}{L}]$, if there are $F \leq \lceil \frac{|\mathcal{N}_k|}{d+1} \rceil - 1$ Byzantine agents in the neighborhood of a normal agent k for all k.*

Proof. Given that the number of the Byzantine neighbors $F \leq \lceil \frac{|\mathcal{N}_k|}{d+1} \rceil - 1$, by Lemma 1, a centerpoint of a set of vectors $\boldsymbol{\psi}_{l,i}$ for $l \in \mathcal{N}_k$, is guaranteed to be in the convex hull of $\boldsymbol{\psi}_{l,i}$ for $l \in \mathcal{N}_k^+$, where $\mathcal{N}_k^+$ is the set of normal neighbors and $\mathcal{N}_k^+ \subseteq \mathcal{N}_k$. Thus, the centerpoint-based aggregation result $\boldsymbol{w}_{k,i}$ and the normal points $\boldsymbol{\psi}_{l,i}$ for $l \in \mathcal{N}_k^+$ holds the following:

$$\begin{aligned} \|\boldsymbol{w}_{k,i} - w^o\| &\leq \max_{l \in \mathcal{N}_k^+} \|\boldsymbol{\psi}_{l,i} - w^o\| \\ &= \max_{l \in \mathcal{N}_k^+} \|\boldsymbol{w}_{l,i-1} - \mu \nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1}) - w^o\|. \end{aligned} \tag{10}$$

We can express $\|\boldsymbol{w}_{l,i-1} - \mu \nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1}) - w^o\|^2$ as

$$\begin{aligned} &\|\boldsymbol{w}_{l,i-1} - \mu \nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1}) - w^o\|^2 \\ =&\|\boldsymbol{w}_{l,i-1} - w^o\|^2 - 2\mu \nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1})^\top (\boldsymbol{w}_{l,i-1} - w^o) \\ &+ \mu^2 \|\nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1})\|^2. \end{aligned} \tag{11}$$

By the co-coercivity of J_l, it yields

$$\begin{aligned} &\left(\nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1}) - \nabla_{w_l} J_l(w_o)\right)^\top (\boldsymbol{w}_{l,i-1} - w^o) \\ \geq &\frac{1}{L}\|\nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1}) - \nabla_{w_l} J_l(w_o)\|^2. \end{aligned}$$

Since $\nabla_{w_l} J_l(w_o) = 0$, we obtain

$$\nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1})^\top (\boldsymbol{w}_{l,i-1} - w^o) \geq \frac{1}{L}\|\nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1})\|^2.$$

[5] For simplicity, we assume that n is divisible by $d+1$.

Put it to (11) and given $\mu \in (0, \frac{2}{L}]$, we obtain

$$\begin{aligned}
&\|\boldsymbol{w}_{l,i-1} - \mu\nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1}) - w^o\|^2 \\
\le&\|\boldsymbol{w}_{l,i-1} - w^o\|^2 - \left(\frac{2\mu}{L} - \mu^2\right)\|\nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1})\|^2 \\
\le&\|\boldsymbol{w}_{l,i-1} - w^o\|^2.
\end{aligned}$$

Put it to (10), we obtain

$$\begin{aligned}
\|\boldsymbol{w}_{k,i} - w^o\|^2 \le \max_{l\in\mathcal{N}_k^+} \|\boldsymbol{w}_{l,i-1} - \mu\nabla_{w_l} J_l(\boldsymbol{w}_{l,i-1}) - w^o\|^2 \\
\le \max_{l\in\mathcal{N}_k^+} \|\boldsymbol{w}_{l,i-1} - w^o\|^2.
\end{aligned} \tag{12}$$

Suppose for each iteration i, there exists $\bar{\boldsymbol{w}}_i$ for all k such that

$$\bar{\boldsymbol{w}}_i = \operatorname*{arg\,max}_{\boldsymbol{w}_{l,i}, l\in\mathcal{N}_k^+} \|\boldsymbol{w}_{l,i} - w^o\|^2.$$

Given (12), we have

$$\|\boldsymbol{w}_{k,i} - w^o\|^2 \le \|\bar{\boldsymbol{w}}_{i-1} - w^o\|^2. \tag{13}$$

Since k can be any normal robot, let k be the one associated with $\bar{\boldsymbol{w}}_i$, such that

$$\|\bar{\boldsymbol{w}}_i - w^o\|^2 \le \|\bar{\boldsymbol{w}}_{i-1} - w^o\|^2.$$

This means $\bar{\boldsymbol{w}}_i$ converges towards w^o as $i \to \infty$. Given (13), for any $k \in \mathcal{N}_k^+$, we conclude that $\boldsymbol{w}_{k,i}$ converges to w^o. □

D. Steady-State Performance Analysis

Compared to the non-cooperative SGD, network cooperation in general leads to the improved steady-state MSD performance without Byzantine agents. Therefore, with no Byzantine agents, centerpoint diffusion achieves better MSD performance as compared to the non-cooperative SGD. In fact, centerpoint-based diffusion performs better than the non-cooperative SGD even if there are Byzantine agents in the network. In each iteration of the algorithm, the centerpoint-based aggregation guarantees that the aggregated result is in the convex hull of normal agents' estimates, that is, the aggregated result can be expressed by a weighted sum of all the normal agents' estimates, i.e., $\boldsymbol{w}_{k,i} = \sum_{l\in\mathcal{N}_k} a_{lk}(i)\boldsymbol{\psi}_{k,i}$, where $0 \le a_{lk}(i) \le 1$ and $\sum_{l\in\mathcal{N}_k} a_{lk}(i) = 1$. Assume the noise variance is uniform across all normal agents, i.e., $\sigma_{v,k}^2 = \sigma_v^2, (k = 1, 2, \ldots, N)$, and let $\mathcal{N}_k^+$ denote the normal neighbors of agent k, then we observe the steady-state MSD performance of the centerpoint-based diffusion algorithm is better than the non-cooperative SGD by (7) and (8) as

$$\begin{aligned}
&\text{MSD}_{\mathcal{CP},\text{net}} - \text{MSD}_{\text{ncop,net}} \\
=&\frac{\mu d}{2} \cdot \left(\sum_{k\in\mathcal{N}_k^+} p_k^2\sigma_{v,k}^2 - \frac{1}{|\mathcal{N}_k^+|}\sum_{k\in\mathcal{N}_k^+} \sigma_{v,k}^2\right) \\
=&\frac{\mu d}{2} \cdot \left(\sum_{k\in\mathcal{N}_k^+} p_k^2 - 1\right) \cdot \sigma_v^2 < 0,
\end{aligned}$$

where $\sum_{k\in\mathcal{N}_k^+} p_k^2 < 1$ given $\sum_{k\in\mathcal{N}_k} p_k = 1$ and $p_k > 0$.

Based on the above discussion, the centerpoint-based aggregation rule always achieves better estimation performance than the non-cooperative SGD. Further, it also outperforms other aggregation rules which may not achieve resilient convergence in the presence of Byzantine agents as illustrated in Section V.

E. Time Complexity to Compute a Centerpoint

In two and three dimensions, the time complexity of computing a centerpoint is $O(n)$ [39] and $O(n^2)$ [40] respectively. However, for higher dimensions $d > 3$, the expected time bound is $O(n^{d-1})$ [40], which is impractical for very large d. Consequently, algorithms are proposed to compute an approximate centerpoint [41]. For instance, given a set of n points, of which at most $\left(\frac{n}{d^{r/r-1}}\right)$ are Byzantine and the remaining are normal points, then using the approximate centerpoint algorithm in [41], we can compute a point that is in the convex hull of normal points in time $O((rd)^d)$, where r is any integer greater than 1. By increasing r, the quality of approximation, and hence the bound on the number of Byzantine agents improves and approaches $\frac{n}{d}$.

VII. Evaluation

In this section, we evaluate the proposed centerpoint-based aggregation rule for the diffusion algorithm on a mobile adaptive network in which agents attempt to solve a target localization problem cooperatively. We show in our experiments that diffusion with average ($a_{lk} = \frac{1}{|\mathcal{N}_k|}$ for $l \in \mathcal{N}_k$), coordinate-wise median (CM), and geometric median (GM) based aggregation rules fail to converge to the true target w^o in the presence of Byzantine agents, but diffusion with centerpoint-based aggregation succeeds in the same scenario. Moreover, the centerpoint-based diffusion achieves better steady-state MSD performance than non-cooperative SGD with and without the presence of Byzantine agents.

A. Network Setup

We consider a mobile adaptive network [5] of N agents that move collectively in pursuit of a target located at $w^o \in \mathbb{R}^d$. Suppose the location of agent k at time i is denoted by $x_{k,i} \in \mathbb{R}^d$. The distance between agent k and target at time i can be expressed as

$$d_k^o(i) = u_{k,i}^o(w^o - x_{k,i}), \tag{14}$$

where $u_{k,i}^o$ denotes the unit direction vector pointing from $x_{k,i}$ to w^o. Suppose agents have only noisy observations $\{d_k(i), u_{k,i}\}$ of the distance and the unit direction vector, i.e.,

$$\begin{aligned}
d_k(i) &= d_k^o(i) + n_k^d(i), \\
u_{k,i} &= u_{k,i}^o + n_{k,i}^u,
\end{aligned} \tag{15}$$

where $n_{k,i}^u$ and $n_k^d(i)$ denote noise terms. Here, $d_k(i) \in \mathbb{R}$ and $u_{k,i} \in \mathbb{R}^d$. From (14) and (15), we have

$$d_k(i) = u_{k,i}(w^o - x_{k,i}) + n_k(i),$$

where $n_k(i) \triangleq -n_{k,i}^u(w^o - x_{k,i}) + n_k^d(i)$. Let $\hat{d}_k(i) \triangleq d_k(i) + u_{k,i}x_{k,i}$, then the goal is to derive a linear model for variables

$\{\hat{d}_k(i), u_{k,i}\}$ as in equation (1). As a result, agents can rely on diffusion for the target localization problem.

Two agents are neighbors if they exchange information with each other. At each iteration i, agent k knows its location $x_{k,i} \in \mathbb{R}^d$ and velocity $v_{k,i} \in \mathbb{R}^d$, and it can observe its neighbors' location $x_{l,i}$ for $l \in \mathcal{N}_k(i)$. Since agents want to achieve harmonious motion and collision avoidance [5], they update the velocity according to the following update rule:

$$v_{k,i+1} = \lambda \cdot h(w_{k,i} - x_{k,i}) + \beta v^g_{k,i} + \gamma \delta_{k,i}, \qquad (16)$$

where $w_{k,i}$ is the estimate of the target location by k at time i, $v^g_{k,i}$ is the velocity of the center of mass of the network, $\lambda, \beta, \gamma, r$ are non-negative parameters, and

$$h(w_{k,i} - x_{k,i}) = \begin{cases} w_{k,i} - x_{k,i}, & \text{if } \|w_{k,i} - x_{k,i}\| \le s \\ s \cdot \frac{w_{k,i} - x_{k,i}}{\|w_{k,i} - x_{k,i}\|}, & \text{otherwise} \end{cases}$$

for some positive scaling factor s used to bound the speed in pursuing the target. Moreover, $\delta_{k,i}$ is given by

$$\delta_{k,i} = \sum_{l \in \mathcal{N}_k \setminus \{k\}} (\|x_{l,i} - x_{k,i}\| - r) \frac{x_{l,i} - x_{k,i}}{\|x_{l,i} - x_{k,i}\|},$$

where r is a non-negative value.

The first term in (16) relates to the objective of having the network move towards the unknown target, and the other two terms suggest that agents should adjust their velocities to be consistent with the average displacement vector in the neighborhood while maintaining a distance from their neighbors. Agents then update their location according to

$$x_{k,i+1} = x_{k,i} + \Delta t \cdot v_{k,i+1},$$

where Δt represents the time step.

To obtain the velocity, agents need to know the estimation of the target location $w_{k,i}$ and the velocity of the center of mass of the network $v^g_{k,i}$, which should be the unique minimizers of the following cost functions:

$$J^{glob}(w) = \sum_{k \in \mathcal{N}^+} \mathbb{E}\|\hat{d}_k(i) - u_{k,i} w\|^2,$$

$$J^{glob}(v^g) = \sum_{k \in \mathcal{N}^+} \|v_{k,i} - v^g\|^2,$$

where $\mathcal{N}^+$ denotes the set of normal agents. The normal agents use the ATC diffusion algorithm to optimize the above cost functions. The adaptation steps take the following form:

$$\psi_{k,i} = w_{k,i} + \mu u^*_{k,i}(\hat{d}_k(i) - u_{k,i} w_{k,i-1}),$$

$$\phi_{k,i} = v^g_{k,i-1} + \nu(v_{k,i} - v^g_{k,i-1}),$$

where μ and ν are step sizes. The aggregation steps can be expressed as follows:

$$w_{k,i} = \mathrm{Aggr}^w(\psi_{1,i}, \psi_{2,i}, \ldots, \psi_{|\mathcal{N}_k|,i}),$$

$$v^g_{k,i} = \mathrm{Aggr}^{v^g}(\phi_{1,i}, \phi_{2,i}, \ldots, \phi_{|\mathcal{N}_k|,i}),$$

where Aggr^w and Aggr^{v^g} represent aggregation rules and $|\mathcal{N}_k|$ denotes the size of $\mathcal{N}_k$.

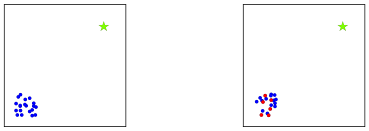
B. Numerical simulation

In our simulation[6], we consider a fully connected (complete) network with 20 agents. We consider a 2-dimensional example and select $w^o = (4, 4)$. The regression vector $u_{k,i}$ has uniform covariance matrix $R_{u,k} = \sigma^2_{u,k} I_2$, $\sigma^2_{u,k} \in [0, 1.0]$ where I_2 is the identity matrix of size 2. The noise variance of distance $\sigma^2_{d,k} = 5.0, \forall k$. When the distance to the target is less than 2.0, both $\sigma^2_{d,k}$ and $\sigma^2_{u,k}$ start to decrease linearly as the distance to the target decreases. The step sizes for updating location and velocity estimates are $\mu = \nu = 0.05$. Further, we select $\lambda = 0.5$, $\beta = 0.5$, $\gamma = 0.01$, $s = 1$, $r = 1$ and $\Delta t = 0.5s$. The sensing range of agents is 4.0. When the distance between two agents is larger than the sensing range, the two agents lose connection.

Centerpoint-based aggregation is resilient up to $\lceil \frac{20}{3} \rceil - 1 = 6$ Byzantine agents. Therefore, we consider the worst case where six Byzantine agents are present in the network. As discussed in Section V, Byzantine agents can send false estimates to make the aggregation results of coordinate-wise median (CM) and geometric median (GM) based aggregation rules to be outside the convex hull of the normal agents' estimates. In our experiments, Byzantine agents continuously send $\psi_{l,i} = \phi_{l,i} = (0, 0)$ to all the normal agents as their current estimates. We run the non-cooperative SGD and diffusion algorithm with average/CM/GM/centerpoint-based aggregation rules to estimate the state $w_{k,i}$ and the velocity $v^g_{k,i}$. Figure 2 shows the initial deployment of agents with and without Byzantine attacks, agents are located in $[0, 1] \times [0, 1]$. Figure 3 shows the final deployment of agents after running diffusion algorithm with different aggregation rules and non-cooperative SGD with and without Byzantine attacks. In these figures, the green star denotes the target location $w^o = (4, 4)$, the red nodes are the Byzantine agents, and the blue nodes denote the normal agents. Byzantine agents do not change their location throughout the simulation.

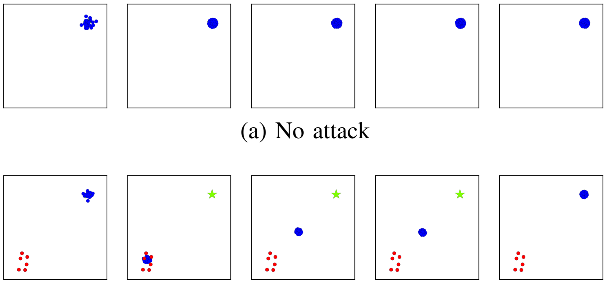
In the case of no attack, we find all the four aggregation rules—average, CM, GM and centerpoint-based—converge to the target as shown in Figure 3a. Yet in the presence of Byzantine agents, only the centerpoint-based diffusion converges to the target as shown in Figure 3b. Figure 4 illustrates the state estimates as a function of time (number of iterations). We observe that if there are no Byzantine agents, all the four diffusion algorithms achieve a better estimation accuracy than the non-cooperative SGD. However, in the presence of six Byzantine agents, only the centerpoint-based diffusion algorithm converges to the target state, whereas the diffusion algorithm with other aggregation rules fails to converge to the target. The steady-state MSD performances are illustrated in Figure 5. We observe that diffusion with all the four aggregation rules achieve better steady-state MSD than the non-cooperative SGD under no attack, whereas only the centerpoint-based diffusion achieves a better steady-state MSD than the non-cooperative SGD under attack.

[6] Simulation code can be found in https://github.com/JianiLi/Centerpoint_resilient_diffusion and video can be found in https://youtu.be/Y9sdOKLKs24.



(a) No attack (b) With 6 Byzantine agents

Fig. 2: Mobile network's initial deployment (green star: target, blue nodes: normal agents, red nodes: Byzantine agents).



(a) No attack

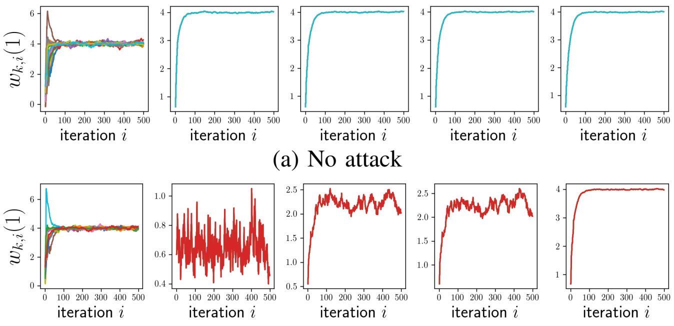
(b) With 6 Byzantine agents

Fig. 3: Mobile network's final deployment (from left to right: noncooperative SGD, average/CM/GM/centerpoint-based diffusion).

C. Simulation on Robotarium testbed

We have also carried out simulations on Robotarium [42], a multirobot testbed developed at the Georgia Institute of Technology, to verify our results. The robots are 11 cm wide, 10 cm long, and operate on a 3m x 2m area as shown in Figure 6. We consider a network of 20 robots that remain fully connected throughout the simulation. In the case of attack, six of them are selected to be Byzantine. The target point is set to be $(2.7, 1.7)$m. The control parameters are the same as in Section VII-B.

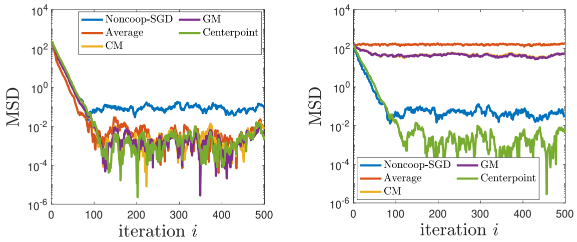
We evaluated the diffusion algorithm with three different aggregation rules, including CM, GM, and centerpoint-based aggregation. Figure 6 shows the initial and final network deployments using CM/GM/centerpoint-based diffusion. The normal and Byzantine robots are indicated by yellow and red colors respectively, and the target location is denoted by the green star. Byzantine robots remain static and send $(0, 0)$ estimates of target location and velocity to normal robots throughout the experiment. We find that without the attack, robots adopting diffusion with CM/GM/centerpoint-based aggregation all converge to the target. However, in the presence of Byzantine agents, only robots adopting the centerpoint-based diffusion converge to the target, whereas robots implementing CM or GM based diffusion converge to somewhere in the middle of the arena.



(a) No attack

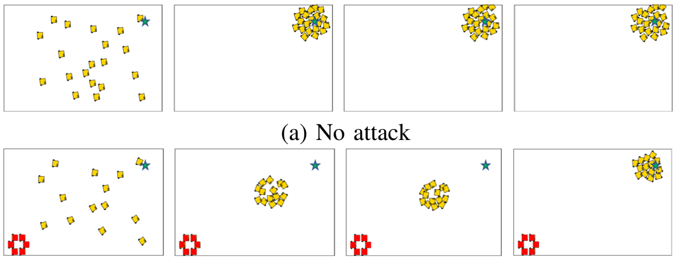
(b) With 6 Byzantine agents

Fig. 4: $w_{k,i}$ (1^{st} dimension) (each line represents the estimates of a normal agent k). From left to right: noncooperative SGD, average/CM/GM/centerpoint-based diffusion.



(a) No attack (b) With 6 Byzantine agents

Fig. 5: Network MSD for different aggregation rules.



(a) No attack

(b) With 6 Byzantine agents

Fig. 6: Network deployment on Robotarium simulator, from left to right: initial deployment, final deployment for CM/GM/centerpoint based-diffusion.

VIII. Conclusion

In this work, we studied resilient aggregation rules for distributed diffusion. We showed that commonly used coordinate-wise median and geometric median-based aggregation do not guarantee resilient convergence for distributed diffusion. We proposed a centerpoint-based aggregation rule that generalizes the resilience property of the median into higher dimensions. The centerpoint-based aggregation rule guarantees that the diffusion algorithm converges to the true target state if the number of Byzantine agents in the neighborhood of a normal agent is less than $\lceil \frac{n}{d+1} \rceil$, where n is the number of nodes in the neighborhood, and d is the dimension of the state vector of the robots. For very large d, exact computation of a centerpoint is a computationally challenging task. In such cases, we can use approximate algorithms to compute such a point, which could reduce the resilience of the diffusion algorithm. For future work, we aim to explore how we can reduce the time complexity for resilient aggregation, probably with degraded learning performance.

References

[1] A. H. Sayed, S. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[2] R. Abdolee, S. Saur, B. Champagne, and A. H. Sayed, "Diffusion LMS localization and tracking algorithm for wireless cellular networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP)*, 2013, pp. 4598–4602.

[3] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *3rd International Workshop on Cognitive Information Processing (CIP)*, 2012, pp. 1–6.

[4] J. Plata-Chaves, N. Bogdanovic, and K. Berberidis, "Distributed diffusion-based LMS for node-specific adaptive parameter estimation," *IEEE Transactions on Signal Processing*, vol. 63, pp. 3448–3460, 2015.

[5] S. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 649–664, 2011.

[6] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, 2017, pp. 119–129.

[7] J. Li, W. Abbas, and X. Koutsoukos, "Resilient distributed diffusion in networks with adversaries," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 1–17, 2020.

[8] Z. Yang, A. Gang, and W. U. Bajwa, "Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the byzantine threat model," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 146–159, 2020.

[9] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.

[10] D. Yin, Y. Chen, K. Ramchandran, and P. L. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 5636–5645.

[11] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *arXiv:1912.13445*, 2019.

[12] E.-M. El-Mhamdi, R. Guerraoui, A. Guirguis, and S. Rouault, "SGD: Decentralized Byzantine resilience," *arXiv:1905.03853*, 2019.

[13] H. Park and S. Hutchinson, "Fault-tolerant rendezvous of multirobot systems," *IEEE Transactions on Robotics*, vol. 33, pp. 565–582, 2017.

[14] J. Matoušek, *Lectures on Discrete Geometry*. Springer, 2002.

[15] J. De Loera, X. Goaoc, F. Meunier, and N. Mustafa, "The discrete yet ubiquitous theorems of Carathéodory, Helly, Sperner, Tucker, and Tverberg," *Bulletin of the American Mathematical Society*, vol. 56, no. 3, pp. 415–511, 2019.

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

[17] H. Zhang and S. Sundaram, "A simple median-based resilient consensus algorithm," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2012, pp. 1734–1741.

[18] W. Abbas, A. Laszka, and X. Koutsoukos, "Improving network connectivity and robustness using trusted nodes with application to resilient consensus," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 4, pp. 2036–2048, 2017.

[19] F. Ghawash and W. Abbas, "Leveraging diversity for achieving resilient consensus in sparse networks," in *8th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys)*, 2019.

[20] L. Guerrero-Bonilla, A. Prorok, and V. Kumar, "Formations for resilient robot teams," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 841–848, 2017.

[21] K. Saulnier, D. Saldana, A. Prorok, G. J. Pappas, and V. Kumar, "Resilient flocking for mobile robot teams," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1039–1046, 2017.

[22] N. H. Vaidya and V. K. Garg, "Byzantine vector consensus in complete graphs," in *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing (PODC)*. ACM, 2013, pp. 65–73.

[23] N. H. Vaidya, "Iterative Byzantine vector consensus in incomplete graphs," in *International Conference on Distributed Computing and Networking (ICDCN)*. Springer, 2014, pp. 14–28.

[24] H. Mendes and M. Herlihy, "Multidimensional approximate agreement in Byzantine asynchronous systems," in *45th Annual ACM Symposium on Theory of Computing (STOC)*, 2013, pp. 391–400.

[25] X. Wang, S. Mou, and S. Sundaram, "A resilient convex combination for consensus-based distributed algorithms," *Numerical Algebra, Control & Optimization*, vol. 9, p. 269, 2019.

[26] M. Shabbir, J. Li, W. Abbas, and X. Koutsoukos, "Resilient vector consensus in multi-agent networks using centerpoints," in *Proceedings of the 2020 American Control Conference (ACC)*, July, 2020.

[27] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," in *Proceedings of the 36th International Conference on Machine Learning, (ICML)*, 2019, pp. 6893–6901.

[28] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 8632–8642.

[29] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking byzantine-tolerant SGD by inner product manipulation," in *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019, p. 83.

[30] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," *arXiv:1911.11815*, 2019.

[31] L. Chen, H. Wang, Z. B. Charles, and D. S. Papailiopoulos, "DRACO: byzantine-resilient distributed training via redundant gradients," in *Proceedings of the 35th International Conference on Machine Learning, (ICML)*, 2018, pp. 902–911.

[32] S. Rajput, H. Wang, Z. Charles, and D. Papailiopoulos, "DETOX: A redundancy-based framework for faster and more robust gradient aggregation," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 320–10 330.

[33] D. Data, L. Song, and S. N. Diggavi, "Data encoding methods for byzantine-resilient distributed optimization," in *IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 2719–2723.

[34] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *33rd AAAI Conference on Artificial Intelligence*, 2019, pp. 1544–1551.

[35] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

[36] C. Xie, O. Koyejo, and I. Gupta, "Generalized Byzantine-tolerant SGD," *arXiv:1802.10116*, 2018.

[37] N. H. Mustafa, S. Ray, and M. Shabbir, "k-centerpoints conjectures for pointsets in $\mathbb{R}^d$," *International Journal of Computational Geometry & Applications*, vol. 25, no. 03, pp. 163–185, 2015.

[38] R. Rado, "A theorem on general measure," *Journal of the London Mathematical Society*, vol. 1, no. 4, pp. 291–300, 1946.

[39] S. Jadhav and A. Mukhopadhyay, "Computing a centerpoint of a finite planar set of points in linear time," *Discrete & Computational Geometry*, vol. 12, no. 3, pp. 291–312, 1994.

[40] T. M. Chan, "An optimal randomized algorithm for maximum tukey depth," in *Proceedings of the 15th annual ACM-SIAM Symposium on Discrete Slgorithms (SODA)*. SIAM, 2004, pp. 430–436.

[41] G. L. Miller and D. R. Sheehy, "Approximate centerpoints with proofs," *Computational Geometry*, vol. 43, no. 8, pp. 647–654, 2010.

[42] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. D. Ames, E. Feron, and M. Egerstedt, "The Robotarium: A remotely accessible swarm robotics research testbed," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1699–1706.