Interplay Between Resilience and Accuracy in Resilient Vector Consensus in Multi-Agent Networks

Waseem Abbas, Mudassir Shabbir, Jiani Li, and Xenofon Koutsoukos

Abstract— In this paper, we study the relationship between resilience and accuracy in the resilient distributed multi-dimensional consensus problem. We consider a network of agents, each of which has a state in $\mathbb{R}^d$. Some agents in the network are adversarial and can change their states arbitrarily. The normal (non-adversarial) agents interact locally and update their states to achieve consensus at some point in the convex hull $\mathcal{C}$ of their initial states. This objective is achievable if the number of adversaries in the neighborhood of normal agents is less than a specific value, which is a function of the local connectivity and the state dimension d. However, to be resilient against adversaries, especially in the case of large d, the desired local connectivity is large. We discuss that resilience against adversarial agents can be improved if normal agents are allowed to converge in a bounded region $\mathcal{B} \supseteq \mathcal{C}$, which means normal agents converge at some point close to but not necessarily inside $\mathcal{C}$ in the worst case. The accuracy of resilient consensus can be measured by the Hausdorff distance between $\mathcal{B}$ and $\mathcal{C}$. As a result, resilience can be improved at the cost of accuracy. We propose a resilient bounded consensus algorithm that exploits the trade-off between resilience and accuracy by projecting d-dimensional states into lower dimensions and then solving instances of resilient consensus in lower dimensions. We analyze the algorithm, present various resilience and accuracy bounds, and also numerically evaluate our results.

I. INTRODUCTION

Consider a network of agents in which each agent maintains a d-dimensional state vector and updates it by interacting with a subset of other agents. Some of the network agents may be adversarial (or faulty) and therefore send incorrect states to their neighbors. Moreover, non-adversarial, or commonly referred to as the *normal* agents, are unaware of these adversarial agents' identities. Resilient distributed multi-dimensional consensus problem requires that in the presence of adversarial agents, normal agents update their states to converge to a common state in the convex hull of normal agents' initial states. Resilient multi-dimensional or vector consensus has several applications, such as in multi-robot networks [1], [2], distributed computing [3], [4], [5], distributed optimization [6], [7] and fault-tolerant multiagent networks [8], [9].

There are distributed algorithms achieving resilient consensus in networks under certain conditions, which include bounding the maximum number of adversaries in the neighborhood of each normal agent, for instance, [10], [11], [12], [13], [14], [15]. A recently proposed approximate distributed robust convergence (ADRC) algorithm [1] guarantees convergence if each normal agent i has at most $\lceil N_i/2^d \rceil - 1$ adversaries in its neighborhood, where N_i is the size of the neighborhood of i, and d is the dimension of the state vector. We observe that the underlying network graph needs

to be dense and highly connected to ensure resilient vector consensus. Resilience incurs significant overhead in the form of many interactions between agents, which explodes with an increase in the dimension d.

In this paper, we study the interplay between resilience and accuracy in the distributed vector consensus algorithms. We analyze algorithms' performance when the number of adversaries exceeds the allowed limit, which is a function of the state dimension d and the size of the neighborhood of normal agents in the underlying network. We discuss that if conditions in resilient vector consensus algorithms are not satisfied, then the adversary can drive normal agents arbitrarily far away from the convex hull of their initial positions. However, the local connectivity requirements of normal agents within the network can be significantly relaxed if we desire normal agents to converge at some point close to but not necessarily inside the convex hull of their initial positions. In other words, we can improve the resilience of vector consensus at the cost of accuracy. We adopt a simple approach of partitioning a d-dimensional state into multiple lower-dimensional states to explore this resilience-accuracy trade-off. Instead of solving a single instance of d-dimensional resilient consensus, we solve multiple instances of lower-dimensional resilient consensus problems. Since a network exhibits improved resilience in low-dimensional states, the overall resilience is improved, albeit with reduced accuracy. Our main contributions are:

- We discuss the notion of accuracy in the resilient vector consensus problem and propose a framework to study the relationship between accuracy and resilience against adversarial agents.
- We formulate the resilient bounded consensus problem to analyze the interplay between resilience and accuracy in higher d and propose an algorithm to solve it.
- We analyze the algorithm and present various resilience and accuracy bounds that demonstrate how the resilience against adversarial agents improves at the cost of accuracy. We also numerically evaluate our results.

The rest of the paper is organized as follows: Section II presents preliminaries. Section III provides an overview of the resilient multi-dimensional consensus problem. Section IV formulates the resilient bounded consensus problem to study the trade-off between resilience and accuracy and also presents an algorithm. Section V analyzes the proposed algorithm. Section VI illustrates a numerical example, and Section VII concludes the paper.

II. Preliminaries

We consider a network of agents modeled by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents agents and $\mathcal{E}$ represents interactions between agents. The state of each agent $i \in \mathcal{V}$ at time t is represented by a point $x_i(t) \in \mathbb{R}^d$. An edge (j, i) means that i can observe the state value of j. The *neighborhood* of i is the set of nodes $\mathcal{N}_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\} \cup \{i\}$. For a given set of points $X \subset \mathbb{R}^d$, we denote its *convex hull* by $\text{conv}(X)$. 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. A point $x \in \mathbb{R}^d$ is an *interior point* of a set $X \subset \mathbb{R}^d$ if there exists an open ball centered at x which is completely contained in X. Let $X_1 \subset \mathbb{R}^{d_1}$ and $X_2 \subset \mathbb{R}^{d_2}$, then the *Cartesian product* of their convex hulls, denoted by $\text{conv}(X_1) \times \text{conv}(X_2)$ is $\{(x_1 \ \ x_2) | \ x_1 \in \text{conv}(X_1) \text{ and } x_2 \in \text{conv}(X_2)\}$. We use terms agents and nodes interchangeably, and similarly use terms points and states interchangeably.

Normal and Adversarial Agents: There are two types of agents in the network, normal and adversarial. *Normal* agents synchronously interact with their neighbors and update their states according to a pre-defined state update rule, which is the consensus algorithm. *Adversarial* agents can change their states arbitrarily and do not follow the pre-defined update rule. Moreover, an adversarial agent can transmit different values to nodes in its neighborhood, referred to as the *Byzantine* model. F_i denotes the number of adversarial agents in the neighborhood of agent i. A normal agent cannot distinguish between its normal and adversarial neighbors.

Resilient Vector Consensus: The goal of the resilient vector consensus is to ensure the following two conditions:
(1) *Safety* – Let $X(0) = \{x_1(0), x_2(0), \cdots, x_n(0)\} \subset \mathbb{R}^d$ be the set of initial states of normal nodes, then at each time step t, and for any normal node i, $x_i(t) \in \text{conv}(X(0))$.
(2) *Agreement* – For every $\epsilon > 0$, there exists some t_ϵ, such that $||x_i(t) - x_j(t)|| < \epsilon$ for all $t > t_\epsilon$, and for all normal node pairs i, j.

III. Resilient Distributed Consensus in $\mathbb{R}^d$

In this section, first, we briefly discuss a resilient distributed vector consensus algorithm, known as the *Approximate Distributed Robust Convergence (ADRC)*, proposed by Park and Hutchinson [1]. Second, we discuss a computational improvement in the algorithm discussed in [16].

A. Approximate Distributed Robust Convergence (ADRC)

The ADRC algorithm guarantees the consensus of normal agents in $\mathbb{R}^d$ if the number of adversarial agents in the neighborhood of each normal agent is bounded by a certain value that depends on d. The notion of *F-safe point* is crucial to understanding the algorithm.

Definition (*F-safe point*) Given a set of N points in $\mathbb{R}^d$, of which at most F are adversarial, then a point p that is guaranteed to lie in the interior of the convex hull of $(N - F)$ normal points is an F-safe point.

The ADRC algorithm relies on the computation of an F_i-safe point by every normal agent i having N_i agents in its neighborhood, of which at most F_i are adversaries. The ADRC is a synchronous iterative algorithm in which each normal agent i updates its state as follows [1]:

- In the iteration t, a normal agent i gathers the state values of its neighbors $\mathcal{N}_i(t)$.
- Then, it computes an F_i-safe point, denoted by $s_i(t)$, of points corresponding to its neighbors' states.
- Agent i then updates it's state as below.

$$x_i(t+1) = \alpha_i(t)s_i(t) + (1 - \alpha_i(t))x_i(t), \quad (1)$$

where $\alpha_i(t)$ is a dynamically chosen parameter in the range (0 1), whose value depends on the application [1].

If all normal agents follow the above routine and their underlying network graph satisfies certain connectivity conditions, they are guaranteed to converge at some point in the convex hull of their initial states [1]. The biggest challenge here is to ensure that each normal agent can compute a safe point. For this, [1] utilized ideas from Discrete Geometry and presented the following result.

Proposition 3.1: *(Theoretical bound)* Given a set of N points in general positions in $\mathbb{R}^d$, where $d \in \{2, 3, \cdots, 8\}$, and at most F points are adversarial, then it is possible to find an F-safe point if

$$N \geq (F + 1)(d + 1). \quad (2)$$

In particular, [1] used the notion of *Tverberg partition* to compute an F-safe point. The main idea is to partition a set of N points in $\mathbb{R}^d$ into $(F + 1)$ parts such that the convex hull of points in one part has a non-empty intersection with the convex hull of points in any other part. An F-safe point is an interior point in the intersection of these $F + 1$ convex hulls. In general, the computation of Tverberg partition of points is an NP-hard problem. The best known approximation algorithm runs in $d^{O(1)N}$ time and computes a Tverberg partition of N points into $F + 1$ parts if $F \leq \lceil \frac{N}{2^d} \rceil - 1$. In other words, we can state the following:

Proposition 3.2: *(Practical bound)* Given a set of N points in general positions in $\mathbb{R}^d$, of which at most F are adversarial, then it is possible to compute an F-safe point (using Tverberg partition) if

$$F \leq \left\lceil \frac{N}{2^d} \right\rceil - 1. \quad (3)$$

Thus, (2) and (3) provide theoretical and practical resilience guarantees of the ADRC algorithm, respectively. Consequently, in a network $\mathcal{G}$, if a normal agent i has N_i neighbors and at most F_i of them are adversarial, then resilient consensus is guaranteed by the ADRC algorithm, if $F_i \leq \lceil \frac{N_i}{2^d} \rceil - 1$, for every normal agent i.

B. Resilient Vector Consensus Using Centerpoint

Recently, [16] proposed to utilize the notion of *centerpoint* instead of Tverberg partition to compute a safe point. The centerpoint can be viewed as an extension of the median in the higher dimension Euclidean space and is defined below.

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

It is shown in [16] that for a given set of N points in $\mathbb{R}^d$, an F-safe point is essentially an interior centerpoint for $F = \lceil \frac{N}{d+1} \rceil - 1$. Using centerpoint to compute safe point improves the practical resilience guarantees of the ADRC algorithm. In particular, we have the following result [16].

Proposition 3.3: Given a set of N points in general positions in $\mathbb{R}^d$, of which at most F are adversarial, then an F-safe point can be computed (using centerpoint) if

$$
\begin{aligned}
F &\leq \left\lceil \frac{N}{d+1} \right\rceil - 1, \quad \text{for } d = 2, 3, \text{ and} \\
F &= \Omega\left(\frac{N_i}{d^2} \right) \qquad\qquad \text{for } d > 3,
\end{aligned}
\tag{4}
$$

where $r > 1$ is some positive integer. Moreover, such an F-safe point can be computed in $O(N)$ and $O(N^2)$ times in $\mathbb{R}^2$ and $\mathbb{R}^3$ respectively, and in $O\left(N^{c \log d}(2d)^d\right)$ in $\mathbb{R}^d$ for $d > 3$, where c is some constant.

For algorithmic details of computing a centerpoint, we refer readers to [17], [18], [19].

IV. RESILIENCE-ACCURACY TRADE-OFF AND RESILIENT BOUNDED CONSENSUS

If the number of adversaries F_i in the neighborhood of a normal agent i satisfies (2), then all normal agents are guaranteed to converge in the convex hull of their initial points. Here, we are interested in analyzing the interplay between between resilience and accuracy of the algorithm. In other words, *what are the implications if the number of adversaries is greater than the one in* (2)? Can the normal agents still converge? If they do, how far could the agreement point be from the convex hull of initial points?

First, we note that if $F \geq \lceil N/(d+1) \rceil$, then an F-safe point does not exist [16]. Consequently, $F \leq \lceil \frac{N}{d+1} \rceil - 1$ is not only a sufficient but also a necessary condition for the existence of an F-safe point. Second, we note that if $F \geq \lceil \frac{N}{d+1} \rceil$, then the centerpoint of a given set of N points can be arbitrarily far away from the convex hull of normal points.[2] Thus, in the ADRC algorithm, if a normal agent i has N_i neighbors, of which $F_i \geq \lceil \frac{N_i}{d+1} \rceil$ are adversarial, and i updates its state based on a centerpoint of its neighbors' states, then $x_i(t)$ can be arbitrarily far away from the convex hull of normal agents' initial states.

Next, we ask *if it is possible to guarantee the convergence of normal agents in some bounded region $\mathcal{B}$ if each normal agent i satisfies $\lceil \frac{N_i}{d+1} \rceil \leq F_i \leq F_i'$ for some F_i'?* Here, we can expect $\mathcal{C} \subseteq \mathcal{B} \subset \mathbb{R}^d$, where $\mathcal{C} = \text{conv}(X(0))$ is the convex hull of the points corresponding to normal agents'

---

[1]A 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 $a \in \mathbb{R}^d \setminus \{0\}$.

[2]By the Centerpoint Theorem, every finite set of points in $\mathbb{R}^d$ has some centerpoint [20].

initial states. In other words, is it possible to improve the resilience of the consensus algorithm in d dimensions at the cost of accuracy, where accuracy measures how far away from $\mathcal{C}$ do normal agents converge? To formalize this, we define the *Resilient Bounded Consensus* problem below.

Definition *(Resilient Bounded Consensus)* Consider a network of agents $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ in which each normal agent i has $N_i(t)$ neighbors, of which at most $F_i(t)$ are adversarial. Normal agents in $\mathcal{V}$ update their states such that at each time step t and for every normal agent i, the state $x_i(t)$ is in a bounded convex region $\mathcal{B} \subset \mathbb{R}^d$ (irrespective of the states of adversarial agents). Moreover, for every $\epsilon > 0$, there exists some t_ϵ such that $||x_i(t) - x_j(t)|| < \epsilon$ for all $t > t_\epsilon$ and for all normal node pairs i, j.

Here, we consider $\mathcal{B}$ such that $\mathcal{C} \subseteq \mathcal{B}$. If $\mathcal{B} = \mathcal{C}$, we get the typical resilient vector consensus problem (Section II).

A. Accuracy and Resilience in Resilient Bounded Consensus

If $x^* \in \mathcal{B}$ is a consensus point of all normal agents in the resilient bounded consensus, then the distance between x^* and $\mathcal{C}$, denoted by $\delta(x^*, \mathcal{C})$, is defined as,

$$
\delta(x^*, \mathcal{C}) = \min_{c \in \partial\mathcal{C}} ||x^* - c||,
\tag{5}
$$

where $\partial\mathcal{C}$ is the boundary of $\mathcal{C}$, and $||x^* - c||$ is the Euclidean distance between points x^* and c. To quantify the *accuracy* of bounded consensus—how far can the agreement point in $\mathcal{B}$ be from $\mathcal{C}$—we use the notion of *Hausdorff distance*, which is often used to measure how well one convex shape approximates the other [21], [22].

Definition *(Hausdorff Distance)* Given two convex regions $\mathcal{B}, \mathcal{C} \subset \mathbb{R}^d$, the *Hausdorff distance* from $\mathcal{B}$ to $\mathcal{C}$ is

$$
\delta(\mathcal{B}, \mathcal{C}) = \max_{b \in \partial\mathcal{B}} \min_{c \in \partial\mathcal{C}} ||b - c||,
\tag{6}
$$

where $||b - c||$ is the Euclidean distance between b and c, and $\partial\mathcal{B}$, $\partial\mathcal{C}$ are the boundaries of $\mathcal{B}$ and $\mathcal{C}$, respectively.

Note that $\delta(x^*, \mathcal{C}) \leq \delta(\mathcal{B}, \mathcal{C})$. Typically, we state the accuracy of resilient bounded consensus relative to the *diameter* of $\mathcal{C}$, which is denoted by $\mu(\mathcal{C})$, and defined as,

$$
\mu(C) = \max_{c_1, c_2 \in \mathcal{C}} ||c_1 - c_2||.
\tag{7}
$$

We are interested in the ratio $\delta(\mathcal{B}, \mathcal{C})/\mu(\mathcal{C})$ to examine the *accuracy* of the resilient bounded consensus.

The *resilience* of the resilient bounded consensus algorithm is measured by the maximum number of adversarial agents F_i in the neighborhood of a normal agent i, such that despite the presence of these adversarial agents, all normal agents achieve resilient bounded consensus inside the convex region $\mathcal{B}$. If $\mathcal{B} = \mathcal{C}$, the resilience bound is $F_i \leq \lceil \frac{N_i}{d+1} \rceil - 1$. As $\mathcal{B}$ grows ($\mathcal{B} \supset \mathcal{C}$), the accuracy deteriorates as $\delta(\mathcal{B}, \mathcal{C})$ increases. At the same time, the resilience bound may improve. We are interested in a resilient bounded consensus algorithm that exploits this *resilience-accuracy* trade-off.

B. Resilient Bounded Consensus Algorithm

Our approach to achieving resilient bounded consensus is to partition the d-dimensional state into parts, implement the centerpoint based resilient consensus algorithm (in lower dimensions) on each part, and then combine the results to get the updated d-dimensional state. The performance, in terms of resilience and accuracy, will depend on the partition of state to lower dimensions.

Next, we present *Resilient Bounded Consensus (RBC)* algorithm for each normal agent i in Algorithm 1. First, we introduce some notations:

I	$\{1, 2, \ldots, d\}$;
P	$\{I_1, I_2, \cdots, I_k\}$ (partition of I into k subsets);
$x_i(t)$	$[x_{i,j}(t)]_{j \in I} \in \mathbb{R}^d$ (state vector);
$x_i^\ell(t)$	$[x_{i,j}(t)]_{j \in I_\ell} \in \mathbb{R}^{|I_\ell|}$ (vector consisting of the values of $x_i(t)$ at the coordinates indexed in I_ℓ);
$F_i^\ell(t)$	$\frac{N_i(t)}{|I_\ell|+1} - 1$, where $N_i(t) = |\mathcal{N}_i(t)|$;
$\bar{\mathcal{V}} \subseteq \mathcal{V}$	set of normal agents.

Algorithm 1 RBC for a Normal Agent $i \in \bar{\mathcal{V}}$

1: **Given** Partition P of $I = \{1, 2, \cdots, d\}$.
2: **for** each iteration t **do**
3:     **for** each $I_\ell \in P$ **do**
4:         Compute $x_j^\ell(t), \forall j \in \mathcal{N}_i(t)$.
5:         Compute an $F_i^\ell(t)$-safe point, say $s_i^\ell(t)$, by computing a centerpoint of $\{x_j^\ell(t)\}, j \in \mathcal{N}_i(t)$.
6:         Update $x_i^\ell(t)$ by the following rule:
        $x_i^\ell(t+1) = \alpha_i(t) s_i^\ell(t) + (1 - \alpha_i(t)) x_i^\ell(t)$.
7:     **end for**
8:     Combine $x_i^\ell(t+1), \forall \ell \in \{1, \cdots, k\}$ to get the updated state $x_i(t+1) \in \mathbb{R}^d$.
9: **end for**

In line 6, $\alpha_i(t)$ satisfies $0 < \alpha_i(t) < 1$, and is chosen depending on the specific application [1]. We illustrate the algorithm using an example in Section VI.

V. ANALYSIS OF THE RESILIENT BOUNDED CONSENSUS ALGORITHM

In this section, we analyze the accuracy and resilience of Algorithm 1. First, we define the notions of *jointly reachable* and *repeatedly reachable* sequence of graphs [1] needed to state the convergence of RBC algorithm. Let $\bar{\mathcal{G}}(t) = (\bar{\mathcal{V}}, \bar{\mathcal{E}}(t))$ be a graph representing normal nodes $\bar{\mathcal{V}} \subseteq \mathcal{V}$ and edges between them at time t.

Definition (*Repeatedly reachable graph sequence*) Let j be a non-negative integer. A finite sequence of graphs $\bar{\mathcal{G}}(T_j), \bar{\mathcal{G}}(T_j + 1) \cdots, \bar{\mathcal{G}}(T_{j+1} - 1)$, where each graph in the sequence has the same vertex set $\mathcal{V}$ is called jointly reachable if the union of graphs defined as

$$\bigcup_{t=T_j}^{T_{j+1}-1} \bar{\mathcal{G}}(t) = \left(\bar{\mathcal{V}}, \bigcup_{t=T_j}^{T_{j+1}-1} \bar{\mathcal{E}}(t) \right)$$

contains a vertex $v \in \bar{\mathcal{V}}$ such that for every $v' \neq v$ there exists a path form v' to v in this union of graphs.

Definition (*Jointly reachable graph sequence*) An infinite sequence of graphs $\bar{\mathcal{G}}(0), \bar{\mathcal{G}}(1) \cdots$ is called repeatedly reachable if there is a sequence of times $0 = T_1 < T_2 < T_3 \cdots$ such that $T_{j+1} - T_j < \infty$ and the subsequence $\bar{\mathcal{G}}(T_j), \bar{\mathcal{G}}(T_j + 1), \cdots \bar{\mathcal{G}}(T_{j+1} - 1)$ is jointly reachable $\forall j$.

Basically, an infinite sequence $\bar{\mathcal{G}}(0), \bar{\mathcal{G}}(1) \cdots$ is repeatedly reachable if it can be partitioned into contiguous finite length subsequences that are themselves jointly reachable.

Moreover, we define $X^\ell(0)$ to be the set of initial positions of normal agents at indices in I_ℓ, that is,

$$X^\ell(0) := \{x_i^\ell(0)\}_{i \in \bar{\mathcal{V}}}, \tag{8}$$

where $\bar{\mathcal{V}}$ is the set of normal agents. Similarly, let $\mathcal{C}^\ell$ be the convex hull of points in $X^\ell(0)$, that is,

$$\mathcal{C}^\ell := \text{Conv}(X^\ell(0)). \tag{9}$$

A consequence of [1, Theorem V.1] is Theorem 5.1 below

Theorem 5.1: Let $\mathcal{G}(t) = (\bar{\mathcal{V}} \cup \mathcal{A}, \mathcal{E}(t))$ be a network of normal $\bar{\mathcal{V}}$ and $\mathcal{A}$ adversarial agents, where each $i \in (\bar{\mathcal{V}} \cup \mathcal{A})$ has a state $x_i(t) \in \mathbb{R}^d$. Let $P = \{I_1, I_2, \cdots, I_k\}$ be a partition of $I = \{1, 2, \cdots, d\}$ into k subsets. Each $i \in \bar{\mathcal{V}}$ implements Algorithm 1, and has at most $F_i(t)$ adversaries in its neighborhood at time t. If

$$F_i(t) \leq \left\lceil \frac{N_i(t)}{\max_\ell |I_\ell| + 1} \right\rceil - 1, \tag{10}$$

and the sequence of connectivity graphs of normal agents $\bar{\mathcal{G}}(0), \bar{\mathcal{G}}(1), \cdots$ is repeatedly reachable, then all normal agents converge to a common point in $\mathcal{B}$, which is a Cartesian product $\mathcal{C}^1 \times \mathcal{C}^2 \times \cdots \times \mathcal{C}^k$.

Proof: See [23]. ∎

Theorem 5.1 provides a resilience bound for the Algorithm 1 and Figure 1 illustrates it in terms of the resilience of the ADRC algorithm. We consider a network in which agents have d-dimensional states and a normal agent i, implementing ADRC algorithm, is resilient against at most F_i adversaries in its neighborhood. In Figure 1(a), we plot the resilience of RBC (in terms of F_i) as a function of the maximum dimension of state obtained after partitioning a d-dimensional state. We observe that resilience improves as the maximum dimension of state in the partition decreases. For instance, if a d-dimensional state is partitioned into two $d/2$-dimensional states, the resilience of RBC improves by a factor of $\frac{\frac{2N}{d+2}-1}{\frac{N}{d+1}-1}$ as compared to the resilience of ADRC algorithm. In Figure 1(b), we fix F_i and plot the number of agents needed in the neighborhood of a normal agent i for it to be resilient against F_i adversaries. Here, N_i is the number of agents needed in the neighborhood of i in the case of ADRC algorithm in d-dimensions. Again, we note that as the maximum dimension of state in the partition decreases, the required number of agents in the neighborhood of i decreases. In other words, same resilience can be achieved with reduced local connectivity.

3130

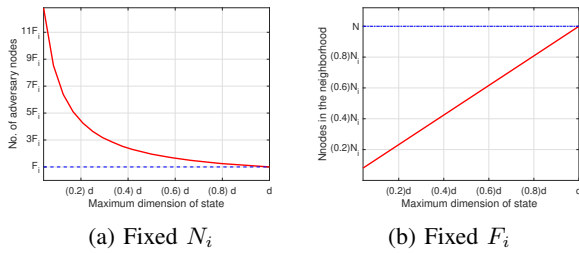(a) Fixed N_i

(b) Fixed F_i

Fig. 1: Resilience of RBC as a function of the maximum dimension of state among partitioned states.

A. Accuracy of the Resilient Bounded Consensus Algorithm

Next, we discuss the accuracy of RBC algorithm by computing the Hausdorff distance $\delta(\mathcal{B}, \mathcal{C})$, where $\mathcal{B} \supseteq \mathcal{C}$ is the convex region in which normal nodes converge as a result of RBC. The shape of region $\mathcal{B}$ and $\delta(\mathcal{B}, \mathcal{C})$ depend on the partition of dimension. If we partition the d-dimensional state into d scalars in the RBC algorithm and each normal agent i satisfies $N_i(t) \geq 2(F_i(t)+1)$, then all normal agents will converge in a hyperrectangle $\mathcal{B} = \mathcal{C}^1 \times \mathcal{C}^2 \times \cdots \times \mathcal{C}^d$, where $\mathcal{C}^\ell$ is defined in (9). Note that each $\mathcal{C}^\ell$ here is an interval in the ℓth dimension. We call such a convex region as the *axis-parallel bounding box*, and note the following.

Fact 5.2: In RBC algorithm, $\delta(\mathcal{B}, \mathcal{C})$ is maximum when $\mathcal{B}$ is the axis-parallel bounding box.

Thus, in the following our goal is to estimate the ratio $\delta(\mathcal{B}, \mathcal{C})/\mu(\mathcal{C})$, where $\mathcal{C}$ is the convex hull of a set of points in $\mathbb{R}^d$, $\mathcal{B}$ is the corresponding axis-parallel bounding box, and $\mu(\mathcal{C})$ is the diameter of $\mathcal{C}$. We start our discussion with the following conjecture:

Conjecture 5.3: For a given set of points in $\mathbb{R}^d$, let $\mathcal{C}$ be the convex hull of points, $\mu(\mathcal{C})$ be the diameter of $\mathcal{C}$ and $\mathcal{B}$ be the corresponding axis-parallel bounding box, then

$$\delta(\mathcal{B}, \mathcal{C}) \leq \sqrt{\frac{d}{2}} \, \mu(\mathcal{C}). \quad (11)$$

We prove the above statement for $d = 3$ and some other special cases. We believe that the above statement is true for any d, and would like to find a proof for the general case in the future. We begin by proving in three dimensions.

Theorem 5.4: Let $\mathcal{C}$ be the convex hull of a given set of points in $\mathbb{R}^3$ and $\mathcal{B}$ be the axis-parallel bounding box of $\mathcal{C}$, then

$$\delta(\mathcal{B}, \mathcal{C}) \leq \sqrt{\frac{3}{2}} \, \mu(\mathcal{C}). \quad (12)$$

Proof: See [23]. ■

In the following, we show that the statement of the conjecture is true for all d for certain *symmetric* pointsets. Before we state and prove the result, we make some observations.

Since translation and rotation of points do not change the Hausdorff distance, we may assume that the origin is a point on $\mathcal{B}$ with the maximum distance from $\mathcal{C}$. As both $\mathcal{C}$ and $\mathcal{B}$ are convex, origin must be a corner vertex of the bounding box. For $\mathcal{B}$ to be a minimum axis-parallel bounding box, each facets adjacent to origin must contain at least one point from the given pointset P. For $1 \leq i \leq d$, let p_i be a point in

P closest to the origin for which the ith coordinate is zero. Then $p_1, p_2, \ldots, p_d$ define a hyperplane X. We observe the following:

Fact 5.5: The Hausdorff distance $\delta(\mathcal{B}, \mathcal{C})$ is at most the Euclidean distance from origin to X regardless of the positions of points $p_{d+1}, p_{d+2}, \ldots, p_n$.

Furthermore, there exists a pointset for which the Hausdorff distance $\delta(\mathcal{B}, \mathcal{C})$ is equal to the Euclidean distance from origin to X, regardless of the positions of points $p_{d+1}, p_{d+2}, \ldots, p_n$. This clearly is the case when all of the points $p_{d+1}, p_{d+2}, \ldots, p_n$ lie in the halfspace defined by X that does not contain origin. From these observations, we deduce that the upper bound on $\delta(\mathcal{B}, \mathcal{C})$ is independent of the positions of points $p_{d+1}, \ldots, p_n$. Thus, we only care about the first d points.

Theorem 5.6: Let $\{p_1, p_2, \ldots, p_d\}$ be a subset of a pointset such that the ith coordinate of the point p_i is zero and all other coordinates are set to some constant a. Then,

$$\delta(\mathcal{B}, \mathcal{C}) \leq (d-1)/d \sqrt{\frac{d}{2}} \times \mu(\mathcal{C}).$$

Proof: See [23]. ■

We note in Theorem 5.6 that as d goes to infinity, $(d-1)/d$ approaches to one and this ratio goes to $\sqrt{\frac{d}{2}} \times \mu(\mathcal{C})$. The following proposition shows that the result in the previous theorem is the best possible.

Proposition 5.7: There exist pointsets for which $\delta(\mathcal{B}, \mathcal{C}) = (d-1)/d \sqrt{\frac{d}{2}} \times \mu(\mathcal{C})$.

Proof: See [23]. ■

VI. NUMERICAL ILLUSTRATION

We illustrate resilient bounded consensus through an example. We consider a complete, undirected network of $N = 20$ agents, of which 6 are adversarial. The network graph is fixed and does not change over time. Since all agents are pairwise adjacent, each normal agent i has $F_i = 6$ adversarial agents in its neighborhood. Moreover, the state of each agent is its position in $\mathbb{R}^3$. Figure 2(a) shows the network graph and initial positions of normal and adversarial agents, which are shown in blue and red colors, respectively. In our simulations, adversarial agents remain static and do not update their positions (states).

First, we implement 3-dimensional ADRC consensus algorithm using centerpoint. Since for each normal agent i, we have $N_i = 20$ and $F_i = 6 > \lceil \frac{N_i}{d+1} \rceil - 1$, the resilience bound in (2) is not satisfied. Figure 2(b) shows the final positions of normal agents and the convex hull of their initial positions. We can see that normal agents fail to converge at a common point inside the convex hull of their positions. Second, we implement the resilient bounded consensus by partitioning the 3-dimensional state into 2-dimensional and scalar states. Since $F_i = 6 \leq \lceil \frac{20}{3} \rceil - 1$, convergence of normal agents is guaranteed in a bounded region $\mathcal{B}_1$. Figure 2(c) shows final positions of normal agents achieving consensus inside the bounded region $\mathcal{B}_1$. Next, we implement the resilient bounded consensus by partitioning 3-dimensional state into three scalar states. The normal agents achieve consensus and
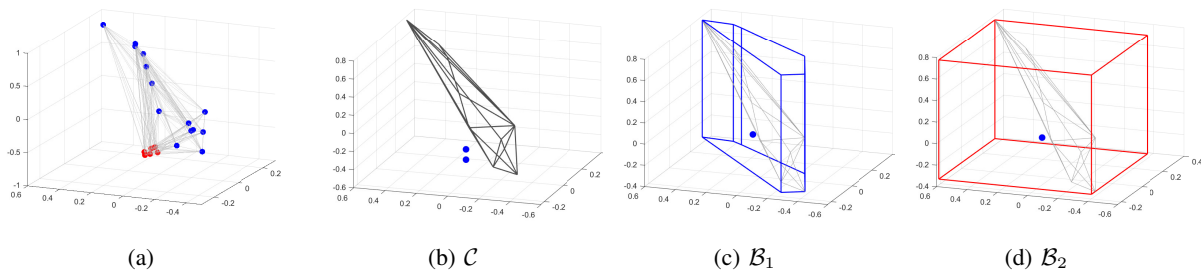
Fig. 2: (a) Initial positions of agents. Final positions of normal agents as a result of (b) 3-dimensional ADRC, (c) RBC by partitioning 3-dimensional state into 2- and 1-dimensional states, and (d) coordinate-wise RBC.

converge at a point inside a bounded region $\mathcal{B}_2$, which is a hyperrectangle illustrated in Figure 2(d).

We observe that the resilient bounded consensus wherein a 3-dimensional state is partitioned into lower-dimensional states has an improved resilience ($F_i = 6$) compared to the 3-dimensional ADRC, which is resilient against at most $F_i = 4$ adversarial agents. However, this improvement comes at the cost of accuracy. The resilient bounded consensus only guarantees that normal agents converge in a bounded region $\mathcal{B}_1 = \mathcal{C}^1 \times \mathcal{C}^2$ and not necessarily inside the convex hull of initial positions $\mathcal{C}$. Let $b_1 \in \mathcal{B}_1$ (similarly $b_2 \in \mathcal{B}_2$) be the convergence point of normal agents as a result of resilient bounded consensus algorithm. Then, the Hausdorff distance-based accuracy bound in (12) is $\delta(b_1, \mathcal{C}) = 0.18 \leq \delta(\mathcal{B}_1, \mathcal{C}) = 0.731 \leq \sqrt{\frac{3}{2}}\mu(\mathcal{C}) = 1.9$. Similarly, in case of $b_2 \in \mathcal{B}_2$, we have $\delta(b_2, \mathcal{C}) = 0.16 \leq \delta(\mathcal{B}_2, \mathcal{C}) = 0.75 \leq \sqrt{\frac{3}{2}}\mu(\mathcal{C}) = 1.9$.

VII. Conclusion

There is a trade-off between resilience and accuracy in the resilient multi-dimensional consensus problem. Resilience depends on the local connectivity of normal agents and the dimension of their state vector. If each normal agent i has less than $\lceil \frac{N_i}{d+1} \rceil$ adversarial agents in its neighborhood, all normal agents can be guaranteed to converge inside the convex hull $\mathcal{C}$ of their initial states. We showed that the convergence of normal agents inside a bounded convex region $\mathcal{B} \supseteq \mathcal{C}$ can be guaranteed even if the number of adversaries in the neighborhood of a normal agent is more than $\lceil \frac{N_i}{d+1} \rceil$. The maximum possible distance between the convergence point in $\mathcal{B}$ and $\mathcal{C}$ can be measured by the Hausdorff distance from $\mathcal{B}$ to $\mathcal{C}$. We provided upper bound on the Hausdorff distance from $\mathcal{B}$ to $\mathcal{C}$ in special cases. In the future, we would like to provide accuracy bounds for more general cases.

References

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

[2] H. Park and S. Hutchinson, "Robust rendezvous for multi-robot system with random node failures: an optimization approach," *Autonomous Robots*, vol. 42, no. 8, pp. 1807–1818, 2018.

[3] L. Tseng and N. Vaidya, "Iterative approximate byzantine consensus under a generalized fault model," in *International Conference on Distributed Computing and Networking*. Springer, 2013, pp. 72–86.

[4] 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)*, 2013, pp. 65–73.

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

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

[7] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1063–1076, 2018.

[8] L. Su and N. H. Vaidya, "Fault-tolerant multi-agent optimization: optimal iterative distributed algorithms," in *ACM Symposium on Principles of Distributed Computing (PDOC)*, 2016, pp. 425–434.

[9] L. Guerrero-Bonilla, D. Saldana, and V. Kumar, "Design guarantees for resilient robot formations on lattices," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 89–96, 2018.

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

[11] 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.

[12] J. Usevitch and D. Panagou, "Resilient leader-follower consensus to arbitrary reference values in time-varying graphs," *IEEE Transactions on Automatic Control*, 2019.

[13] D. Saldana, A. Prorok, S. Sundaram, M. F. Campos, and V. Kumar, "Resilient consensus for time-varying networks of dynamic agents," in *2017 American control conference (ACC)*. IEEE, 2017, pp. 252–258.

[14] H. Mendes and M. Herlihy, "Multidimensional approximate agreement in byzantine asynchronous systems," in *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing (STOC)*, 2013.

[15] S. M. Dibaji and H. Ishii, "Consensus of second-order multi-agent systems in the presence of locally bounded faults," *Systems & Control Letters*, vol. 79, pp. 23–29, 2015.

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

[17] 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.

[18] T. M. Chan, "An optimal randomized algorithm for maximum tukey depth," in *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2004, pp. 430–436.

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

[20] J. Matousek, *Lectures on Discrete Geometry*. Springer, 2002.

[21] E. M. Bronstein, "Approximation of convex sets by polytopes," *Journal of Mathematical Sciences*, vol. 153, no. 6, pp. 727–762, 2008.

[22] M. A. Lopez and S. Reisner, "Hausdorff approximation of convex polygons," *Computational Geometry*, vol. 32, pp. 139–158, 2005.

[23] W. Abbas, M. Shabbir, j. Li, and X. Koutsoukos, "Interplay between resilience and accuracy in resilient vector consensus in multi-agent networks," 2018. [Online]. Available: https://arxiv.org/abs/2009.02417