

# Resilient Distributed Vector Consensus Using Centerpoint <sup>★</sup>

Waseem Abbas <sup>a</sup>, Mudassir Shabbir <sup>b</sup>, Jiani Li <sup>b</sup>, Xenofon Koutsoukos <sup>b</sup>

<sup>a</sup>*Department of Systems Engineering, University of Texas at Dallas, Richardson, TX, USA.*

<sup>b</sup>*Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA.*

---

## Abstract

In this paper, we study the resilient vector consensus problem in networks with adversarial agents and improve resilience guarantees of existing algorithms. A common approach to achieving resilient vector consensus is that every non-adversarial (or normal) agent in the network updates its state by moving towards a point in the convex hull of its *normal* neighbors' states. Since an agent cannot distinguish between its normal and adversarial neighbors, computing such a point, often called as *safe point*, is a challenging task. To compute a safe point, we propose to use the notion of *centerpoint*, which is an extension of the median in higher dimensions, instead of Tverberg partition of points, which is often used for this purpose. We discuss that the notion of centerpoint provides a complete characterization of safe points in  $\mathbb{R}^d$ . In particular, we show that a safe point is essentially an interior centerpoint if the number of adversaries in the neighborhood of a normal agent  $i$  is less than  $\frac{N_i}{d+1}$ , where  $d$  is the dimension of the state vector and  $N_i$  is the total number of agents in the neighborhood of  $i$ . Consequently, we obtain necessary and sufficient conditions on the number of adversarial agents to guarantee resilient vector consensus. Further, by considering the complexity of computing centerpoints, we discuss improvements in the resilience guarantees of vector consensus algorithms and compare with the other existing approaches. Finally, we numerically evaluate the performance of our approach through experiments.

*Key words:* Resilient consensus, computational geometry, centerpoint, fault tolerant networks.

---

## 1 Introduction

Resilient consensus in a network of agents, some of which might be adversarial or faulty, has several applications in multirobot networks, distributed computing, estimation, learning and optimization (for instance, see [2,3,4,5,6,7]). The main goal of resilient consensus is to ensure that all normal agents in a network agree on a common state despite the presence of some adversarial agents, whose identities are unknown to normal agents. Resilient consensus is achieved if appropriate state update laws are designed for normal agents and the underlying network topology satisfies certain connectivity and robustness conditions. For instance, when agents' states are scalars, [8] presents a resilient distributed algorithm that guarantees convergence of normal agents to a common state  $\bar{x} \in [x_{\min} x_{\max}]$ , where  $x_{\min}$  and  $x_{\max}$  are the minimum and maximum of the initial val-

ues of normal agents, respectively. If agents' states are vectors or points in  $\mathbb{R}^d$ ,  $d \geq 2$ , then the resilient consensus objective is to ensure that normal agents converge at some point in the convex hull of their initial states. One such application is the resilient multirobot rendezvous problem, where the goal is to gather mobile robots dispersed in a region at a common point in a distributed manner [4,9]. The robots must move within their original convex region to ensure that they do not drift to an unsafe region. A simple approach could be to run  $d$  instances of scalar resilient consensus, one for each dimension. However, as a result of this approach, normal agents might converge at a point outside the convex hull of their initial states, as discussed in [10]. We also illustrate such a situation in Figure 1. As a result, we cannot rely on resilient scalar consensus algorithms to achieve resilient vector consensus.

In this paper, we study the resilient vector consensus problem and propose a *resilient aggregation rule* for normal (non-adversarial) agents to update their states by combining their neighbors' states, which might include adversarial agents. In the proposed resilient aggregation rule, each normal agent computes a *centerpoint*, which is an extension of the median in the higher dimensional

---

<sup>★</sup> Some preliminary results are presented in [1].

*Email addresses:* waseem.abbas@utdallas.edu (Waseem Abbas), mudassir.shabbir@vanderbilt.edu (Mudassir Shabbir), jiani.li@vanderbilt.edu (Jiani Li), xenofon.koutsoukos@vanderbilt.edu (Xenofon Koutsoukos).

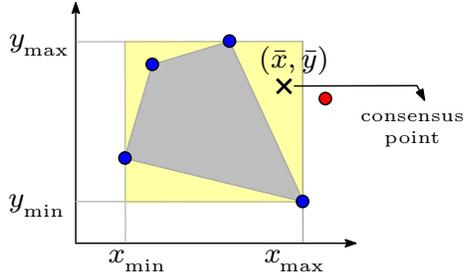


Fig. 1. Normal nodes (blue) implement resilient scalar consensus for  $x$  and  $y$  variables separately, and converge at  $(\bar{x}, \bar{y})$ . Here,  $\bar{x} \in [x_{\min}, x_{\max}]$  and  $\bar{y} \in [y_{\min}, y_{\max}]$ , as guaranteed by resilient scalar consensus. However,  $(\bar{x}, \bar{y})$  is outside the convex hull of normal nodes, which is a (gray) shaded region.

Euclidean space, of its neighbors' states. We then obtain necessary and sufficient conditions on the number of adversarial agents each normal node can have in its neighborhood, while guaranteeing resilient vector consensus. We show that the centerpoint-based aggregation improves the resilience of the known vector consensus algorithms, for instance, the *Approximate Distributed Robust Convergence (ADRC)* algorithm recently proposed in [4]. We note that centerpoint of points in  $\mathbb{R}^d$  is different from taking the coordinate-wise median of points.

Various resilient vector consensus algorithms have been proposed in the literature, for instance, see [4,10,11,9,12,13,14]. The main idea in these algorithms is that each normal agent updates its state iteratively by computing a state (point in  $\mathbb{R}^d$ ) that is guaranteed to lie in the *interior of the convex hull of its normal neighbors'* states. Since a normal agent cannot distinguish between its normal and adversarial neighbors, computing such a point, which is often referred to as the *safe point*, is the primary challenge in these consensus algorithms. To compute a safe point, previous works [4,9] have utilized the idea of *Tverberg partition* of points in  $\mathbb{R}^d$  (discussed in Section 3). We argue that instead of computing Tverberg partition, it is better to use the notion of *centerpoint* in  $\mathbb{R}^d$  to compute safe points. We show that safe point is essentially an interior centerpoint of the neighbors' states. This perspective provides a complete characterization of safe points, and hence allows us to improve the resilience bound of the vector consensus algorithms. A centerpoint has been known to the discrete geometry community for a long time, and its properties and generalizations are still an active topic of research [15,16]. We summarize our contributions below:

- For a given set of  $N_i$  points in  $\mathbb{R}^d$ , of which any  $F_i$  are adversarial, we show the equivalence between the notion of a safe point and an interior centerpoint for  $N_i > F_i(d+1)$ . Here,  $F_i$  and  $N_i$  denote the number of adversarial agents and the total number of agents in the neighborhood of a normal agent  $i$ , respectively. Using this relationship, we discuss how the resilience of the vector consensus algorithm can be improved using

centerpoint to compute a safe point.

- We generalize a sufficient condition for the existence of safe point to arbitrary  $d > 0$ , where  $d$  is the dimension of the state vector. In particular, we show that if  $F_i < \lfloor \frac{N_i}{d+1} \rfloor$ , then there always exists an *interior centerpoint*, and hence, a safe point. A similar condition for the existence of safe point was established in [4] for  $d \leq 8$  using the idea of Tverberg partition.
- Using the notion of centerpoint, we show that  $N_i > (d+1)F_i$  is not only sufficient but also a necessary condition for computing a safe point.
- We show that the centerpoint-based vector consensus algorithm is resilient to  $F_i < \lfloor \frac{N_i}{d+1} \rfloor$  adversaries in the neighborhood of a normal agent  $i$  in  $d = 2, 3$ . In higher dimensions ( $d > 3$ ), the centerpoint-based algorithm is resilient to  $F_i$  adversaries, where  $F_i < F_{max} = \Omega(\frac{N_i}{d^2})$ . This approach offers improved resilience compared to the Tverberg partition-based algorithm, where  $F_{max} = \Omega(\frac{N_i}{2^d})$ .
- We numerically evaluate and compare our results with other algorithms by simulating resilient vector consensus in multirobot networks.

In the case of resilient *scalar* consensus, the *trimming* method has been an effective approach to aggregate neighbors' states. The basic idea in this scheme is that a normal agent collects its neighbors' states (scalar), sorts them, and then disregards *extreme* values, that is,  $F$  largest and smallest values, where  $F$  is an upper bound on the maximum number of adversarial agents in the neighborhood of a normal agent. Depending on the problem setup and threat model,  $F$  can also be an upper bound on the total number of adversarial agents in the network. The normal agent then updates its state by aggregating states of the remaining neighbors, for instance, by computing their average or median. This approach has been applied to achieve resilient consensus in a wide variety of setups, attack scenarios, and network conditions in various application domains including distributed computing [17,18,19], multirobot systems [20,21,22,23], network control [2,8,24,25,26,27,28], and distributed optimization [3,29,30,31,32,33]. The trimming approach, however, is not directly applicable to the resilient vector consensus as the appropriate notion of vector states that are 'farthest' from a normal agent's own state is difficult to define. Sorting states by simply computing the Euclidean distance between them in  $\mathbb{R}^d$  does not work and more advanced ways need to be defined for this purpose [34,35]. At the same time, implementing resilient scalar consensus using the trimming approach in each dimension separately does not always solve the resilient vector consensus problem as we discuss earlier (Figure 1). Consequently, in the resilient vector consensus case, the notion of *safe point*, which is a point obtained by a convex combination of node's normal neighbors' states, is quite useful. Using the idea of safe point, various algorithms and resilience bounds have been proposed in the literature for the resilient

vector consensus and distributed optimization problems [7,10,11,9,12,13,14,36]. Here, we focus on computational aspects and geometric characterization of safe points.

The rest of the paper is organized as follows: Section 2 introduces notations and preliminaries. Section 3 provides an overview of the resilient vector consensus algorithms, in particular, the ADRC algorithm and its resilience bounds. Section 4 discusses the notion of centripoint for resilient vector consensus and presents main results in the paper. Section 5 presents a numerical evaluation of our results, and Section 6 concludes the paper.

## 2 Notations and 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. Each agent  $i \in \mathcal{V}$  has a  $d$ -dimensional state vector whose value is updated over time. The state of agent  $i$  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 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  $X$ . 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, denoted by  $\mathcal{V}_n \subseteq \mathcal{V}$ , are the ones that interact with their neighbors synchronously and always update their states according to a pre-defined state update rule, which is the consensus algorithm. *Adversarial* agents, denoted by  $\mathcal{V}_f \subset \mathcal{V}$ , are the ones that can change their states arbitrarily and do not follow the pre-defined state update rule. Moreover, an adversarial node can transmit different values to its different neighbors, which is referred to as the *Byzantine* model. The number of adversarial nodes in the neighborhood of a normal node  $i$  is denoted by  $F_i$ . For a normal node  $i$ , all nodes in its neighborhood are indistinguishable, that is,  $i$  cannot identify which of its neighbors are adversarial.

*Resilient Vector Consensus* – The goal of the resilient vector consensus is to ensure the following conditions:

- *Safety* – Let  $X(0) = \{x_1(0), x_2(0), \dots, x_n(0)\} \subset \mathbb{R}^d$  be the set of initial states of normal nodes, then at each time step  $t$ , and for any normal node  $i$ , the state value of  $i$ , denoted by  $x_i(t)$  should be in the  $\text{conv}(X(0))$ .
- *Agreement* – For every  $\epsilon > 0$ , there exists some  $t_\epsilon$ , such that for any normal node pair  $i, j$ ,  $\|x_i(t) - x_j(t)\| < \epsilon$ ,  $\forall t > t_\epsilon$ .

## 3 Background and Resilient Vector Consensus Algorithm

In this section, we provide an overview of resilient vector consensus algorithms in which the main idea is that each normal agent updates its state by moving towards a point in the convex hull of its normal neighbors. In particular, we discuss the resilient vector consensus algorithm, known as the *approximate distributed robust convergence (ADRC)*, recently proposed in [4]. Then, we discuss improvement in the resilience guarantees of the algorithm by reconsidering its computational aspects. We note here that our results are general and mainly provide an improved way for normal agents to aggregate their neighbors' states in the presence of adversarial agents. Consequently, these results can be applied to a broader class of algorithms that require each normal agent in the network to aggregate its neighbors' states, such that the aggregated state is guaranteed to lie in the convex hull of its normal neighbors, for instance, as in resilient distributed diffusion [36,37].

The ADRC algorithm is a synchronous iterative algorithm that guarantees the consensus of normal agents 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 3.1** (*F-safe point*) *Given a set of  $N$  points in  $\mathbb{R}^d$ , of which at most  $F$  are adversarial, then a point  $s$  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  $F_i(t)$ -safe point in each iteration  $t$  by every normal agent  $i$  having  $N_i(t)$  agents in its neighborhood, of which at most  $F_i(t)$  are adversaries.<sup>1</sup> Each normal agent  $i$  updates its state as follows [4]:

- In the iteration  $t$ , a normal agent  $i$  gathers the state values of its neighbors, and then computes an  $F_i(t)$ -safe point, denoted by  $s_i$ .
- Agent  $i$  then updates its 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 [4].

*Convergence* – It is shown in [4] that if all normal agents update their states according to (1) after computing  $F_i(t)$ -safe points in each iteration  $t$  and the sequence of graphs induced by normal agents is *repeatedly reachable*

<sup>1</sup> For simplicity, when the context is clear, we use  $N_i$  and  $F_i$  instead of  $N_i(t)$  and  $F_i(t)$ , respectively.

(as defined below), then all normal agents converge at a common point and achieve resilient consensus. The notion of repeatedly reachable sequence of graphs is defined in [4]. We state it here also for completeness.

Let  $\bar{\mathcal{V}}$  be the set of *normal* agents in the network and  $\bar{\mathcal{G}}(t) = (\bar{\mathcal{V}}, \mathcal{E}(t))$  be the graph induced by normal agents at time  $t$ .

**Definition 3.2** (*Jointly Reachable Sequence of Graphs*) Let  $j$  be a non-negative integer. A finite sequence of graphs  $\bar{\mathcal{G}}(T_j), \bar{\mathcal{G}}(T_j+1), \dots, \bar{\mathcal{G}}(T_{j+1}-1)$ , where each graph in the sequence has the same vertex set  $\bar{\mathcal{V}}$  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 from  $v'$  to  $v$  in this union of graphs.

**Definition 3.3** (*Repeatedly Reachable Sequence of Graphs*) An infinite sequence of graphs  $\bar{\mathcal{G}}(0), \bar{\mathcal{G}}(1), \dots$  is called *repeatedly reachable* if there is a sequence of times  $0 = T_1 < T_2 < T_3 < \dots$  such that  $T_{j+1} - T_j < \infty$  and the subsequence  $\bar{\mathcal{G}}(T_j), \bar{\mathcal{G}}(T_j + 1), \dots, \bar{\mathcal{G}}(T_{j+1} - 1)$  is jointly reachable for all  $j$ .

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

We note that two types of conditions need to be satisfied for resilient consensus: (a) The underlying network topology should be such that the sequence of graphs induced by normal agents is repeatedly reachable. (b) Each normal agent can compute a safe point in each iteration. Our focus in this paper is on the computation of a safe point, including conditions under which it is possible.

*Computation of Safe point* – It is the key step in the ADRC algorithm. For this, [4] utilizes the notion of *Tverberg partition* [38]. The main idea is to partition a set of  $N$  points in  $\mathbb{R}^d$  into  $(F + 1)$  subsets such that the convex hull of points in one subset has a non-empty intersection with the convex hull of points in any other subset. The intersection of convex hulls of these  $(F + 1)$  subsets is a *Tverberg region* and any interior point in such a region is an  $F$ -safe point, as illustrated in Figure 2. To see this, observe that if there are at most  $F$  points corresponding to adversarial agents, then there is at least one subset in the partition that consists of points corresponding to normal agents only. The convex hull of points in that subset and therefore, the Tverberg region is guaranteed to lie in the convex hull of points corresponding to normal agents, which means that an interior point in the Tverberg region is an  $F$ -safe point.

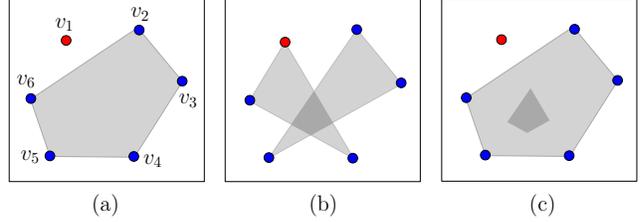


Fig. 2. (a) Five normal (blue) and a single adversarial (red) node. Shaded area is the convex hull of normal nodes. (b) A Tverberg partition consisting of two subsets, of which one contains only normal nodes. Convex hulls of both subsets have a non-empty intersection, corresponding to a Tverberg region (dark shaded). (c) The intersection of Tverberg region and the convex hull of normal nodes.

Thus, to achieve resilient consensus using ADRC algorithm, a normal agent  $i$  computes an  $F_i$ -safe point by computing an interior point in the Tverberg region of an appropriate Tverberg partition. Now, for given  $N_i, F_i$  and  $d$ , the condition guaranteeing the existence of Tverberg partition is stated below [4,39,40,41].

**Theorem 3.4** (*Sufficient condition for  $F_i$ -safe point*) Given a set of  $N_i$  points in general positions in  $\mathbb{R}^d$  where  $d \in \{2, 3, \dots, 8\}$ , of which at most  $F_i$  points correspond to adversarial nodes, then it is possible to find an  $F_i$ -safe point if

$$F_i \leq \frac{N_i}{d+1} - 1. \quad (2)$$

In general, computing a Tverberg partition is an NP-hard problem and it could take  $d^{O(d^2)} N_i^{(d+1)^2+1}$  time to compute a Tverberg partition of  $N_i$  points in  $\mathbb{R}^d$  into  $\lceil N_i/(d+1) \rceil$  parts [42]. The best known algorithm that computes it in a reasonable run time is an approximate algorithm [43], which has a time complexity of  $d^{O(1)} N_i$ . The algorithm is approximate in a sense that to have a partition of  $N_i$  points into  $r$  subsets,  $N_i \geq 2^d r$  (compared to the theoretical bound,  $N_i \geq r(d+1)$ ). Consequently, to compute an  $F_i$ -safe point in the presence of  $F_i$  adversarial neighbors, a normal agent  $i$  needs to have at least  $N_i \geq (F_i + 1)2^d$  agents in its neighborhood, which gives the following resilience bound [4].

**Theorem 3.5** (*Practical resilience bound*) Given a set of  $N_i$  points in general positions in  $\mathbb{R}^d$ , of which at most  $F_i$  are adversarial, then it is possible to compute an  $F_i$ -safe point using Tverberg partition if

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

Note that (3) indicates *resilience* of the ADRC algorithm based on approximate Tverberg partitions to compute safe points. For instance, the algorithm guarantees resilient consensus in  $\mathbb{R}^2$  if for every normal agent, less than 25% of its neighbors are adversarial.

Here, we ask if it is possible to improve the resilience of the ADRC algorithm. In particular, we study the following problems in the rest of the paper.

- (1) ADRC algorithm based on approximate Tverberg partition is resilient against  $\Omega\left(\frac{N_i}{2^d}\right)$  adversaries in the neighborhood of a normal agent  $i$ . Can we improve the resilience by computing safe points some other way?
- (2) We have a sufficient condition for an  $F_i$ -safe point in (2). Can we provide a complete characterization of an  $F_i$ -safe point? In other words, what is a necessary and sufficient condition for a point to be an  $F_i$ -safe point?
- (3) The sufficient condition in (2) holds for  $d \leq 8$ . Can we generalize results to dimensions  $d > 8$ ?

The primary difference between our proposed approach and previous methods is the characterization and computation of safe points. There are other algorithms in the literature to compute safe points, for instance, [4,9,13,35]. We discuss in detail that the centerpoint-based approach is better for the characterization and computation of safe points and also compare this approach with the existing methods in Section 4.5.

#### 4 Resilient Vector Consensus Using Centerpoint

In this section, we show that it is possible to improve the resilience of ADRC algorithm by computing safe points differently, that is, by using the notion of *centerpoint* instead of Tverberg partition. Moreover, centerpoint provides a better characterization of necessary and sufficient conditions for computing safe points.

##### 4.1 $F$ -Safe Point and Interior Centerpoint

The notion of centerpoint can be viewed as an extension of median in higher dimensions and is defined below.

**Definition 4.1** (*Centerpoint*) Given a set  $\mathcal{S}$  of  $N$  points in  $\mathbb{R}^d$  in general positions, a centerpoint  $p$  is a point, such that any closed halfspace<sup>2</sup> of  $\mathbb{R}^d$  that contains  $p$  also contains at least  $\frac{N}{d+1}$  points from  $\mathcal{S}$ . Note that centerpoint does not need to be one of the points in  $\mathcal{S}$ .

Intuitively, a centerpoint lies in the ‘center region’ of the point cloud, in the sense that there are enough points of  $\mathcal{S}$  on each side of a centerpoint. More precisely, it is a point such that any hyperplane that goes through this point divides the set of points in  $\mathcal{S}$  into two subsets, each of which contains at least  $\frac{1}{d+1}$  fraction of points. Figure 3 illustrates centerpoint in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ . In Figure 3(a), any

<sup>2</sup> Recall that closed halfspace in  $\mathbb{R}^d$  is a set of the form  $\{x \in \mathbb{R}^d : a^T x \geq b\}$  for some  $a \in \mathbb{R}^d \setminus \{0\}$ .

line through a centerpoint divides a set of 9 points into two subsets, each of which contains at least 3 points. Similarly, any plane through a centerpoint in Figure 3(b) divides a set of 12 points into two parts, each containing at least 3 points.

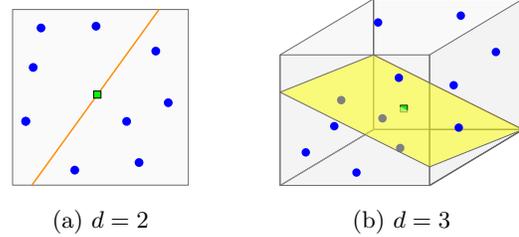


Fig. 3. Illustration of centerpoints (green squares).

A related notion of *point depth* is defined as follows:

**Definition 4.2** (*Point depth*) For a given point set, the depth of a point  $q$  is the maximum number  $\alpha$  such that every closed halfspace containing  $q$  contains at least  $\alpha$  points.

Thus, a centerpoint has depth at least  $\frac{N}{d+1}$ . A centerpoint does not need to be 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 closed and convex. The existence of a center point for any given set  $\mathcal{S}$  is guaranteed by the famous Centerpoint Theorem, which asserts that for any given point set in general positions in an arbitrary dimension, a centerpoint always exists. (see [44,45]).

We show that there is a close relationship between an  $F$ -safe point and an interior centerpoint, which can be exploited to provide a complete characterization of  $F$ -safe points. We state the following result:

**Theorem 4.3** For a given set of  $N_i$  points in  $\mathbb{R}^d$  (corresponding to  $N_i$  agents in the neighborhood of agent  $i$ ), if  $N_i > F_i(d+1)$ , then an  $F_i$ -safe point is equivalent to an interior centerpoint.

*Proof:* For the proof of this statement, we observe that every centerpoint of a given set of  $N_i$  points lies in the intersection of all convex sets that contain more than  $\frac{N_i d}{d+1}$  points [44]. To see this, imagine a convex set  $C$  that does not contain a centerpoint  $p$ . Since  $C$  is convex and  $p$  is an outside point, there exists a hyperplane  $H$  that separates  $C$  from  $p$ . The hyperplane  $H$  defines two halfspaces of  $\mathbb{R}^d$ : one halfspace contains  $p$ , and the other contains  $C$ . The halfspace that contains  $C$  contains more than  $\frac{N_i d}{d+1}$  of the given points. Therefore, the halfspace that contains  $p$  must contain less than  $\frac{N_i}{d+1}$  of the given points, which is a contradiction because  $p$  is a centerpoint. It implies that every convex set containing more than  $\frac{N_i d}{d+1}$  points must contain every centerpoint. Thus, all centerpoints lie in the intersection of such convex sets.

Now, we consider an agent  $i$  having  $N_i$  agents in its neighborhood. An  $\left(\frac{N_i-1}{d+1}\right)$ -safe point lies in the interior of convex hull of points corresponding to normal agents when there are at most  $\frac{N_i-1}{d+1}$  adversarial agents in the neighborhood of agent  $i$ . As there are more than  $\frac{N_i d}{d+1}$  normal neighbors of  $i$ , an interior centerpoint lies in the relative interior of convex hull of normal agents' points. Therefore, every interior centerpoint is an  $F_i$ -safe point. On the other hand, if there are  $\frac{N_i-1}{d+1}$  points corresponding to adversarial agents (whose identities are unknown) in the neighborhood of  $i$ ; then an  $F_i$ -safe point, where  $F_i = \frac{N_i-1}{d+1}$ , must lie in the interior of an arbitrary convex set  $X$  containing more than  $\frac{N_i d}{d+1}$  points. This is true because otherwise an adversary can choose all agents corresponding to points in  $X$  to be normal and all remaining agents to be adversarial to leave a chosen point 'unsafe'. It follows that every  $F_i$ -safe point must lie in the interior of the intersection of convex sets that contain more than  $\frac{N_i d}{d+1}$  points. Thus, every  $F_i$ -safe point must be an interior centerpoint. ■

We illustrate the equivalence between  $F_i$ -safe point and centerpoint through an example. Consider six points in  $\mathbb{R}^2$  in Figure 4(a). The gray region is the set of all centerpoints. At the same time, the gray region is also the set of all 1-safe points. It means that no matter which one of the six points corresponds to an adversarial agent, every point in the gray region is guaranteed to lie in the convex hull of remaining five points corresponding to normal agents. For example, in Figure 4(b), the red point is an adversary and the yellow region is the convex hull of normal agents' points. We observe that all 1-safe points (gray region) lie inside the convex hull of normal agents' points. The same is illustrated for other possibilities for a single adversarial point in Figures 4(c)–(g).

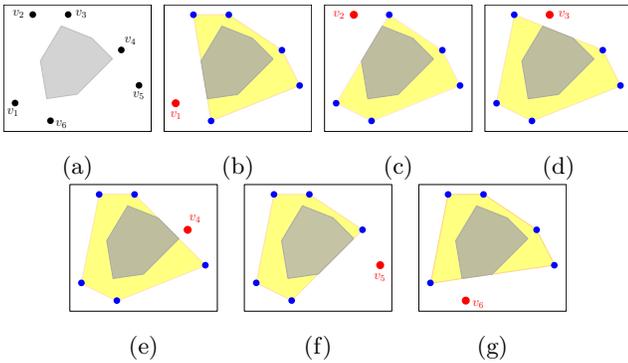


Fig. 4. The region of centerpoints (gray area) is same as the region of 1-safe points.

#### 4.2 Necessary Condition for $F_i$ -Safe Point

We note that  $\frac{1}{d+1}$  is the best possible fraction (for  $F_i$ ) in Theorem 4.3, that is, there exist general node positions

where allowing more adversary nodes would mean that there is no safe point at all. For example, consider a set of three points  $a, b, c$ , in  $\mathbb{R}^2$  that lie on the vertices of a non-degenerate triangle.<sup>3</sup> If one of these points is malicious then there is no 1-safe point in the plane. Clearly, no point in the interior of the triangle can be 1-safe because it does not lie in the convex hull of normal points for any choice of point corresponding to an adversary. Similarly, any point that lies on an edge of the triangle, say  $\overline{ab}$ , can not be 1-safe because if one of the endpoint, say  $a$ , is adversarial then the convex hull of the remaining points  $b$  and  $c$  does not contain this point. Now in  $\mathbb{R}^3$ , consider four points  $a, b, c, d$ , lying on the vertices of a non-degenerate tetrahedron. Assume one of these points, whose identity is unknown, is adversarial. A point  $x$  in the interior of tetrahedron can not be 1-safe because if  $a$  is malicious then  $x$  is not contained in the convex hull of normal points  $b, c, d$ , as shown in Figure 5(a). Similarly, consider a point  $x$  that lies on a facet, say  $\overline{abc}$ , of the tetrahedron. The point  $x$  can not be 1-safe because if one of the points on its facet, say  $b$ , is adversarial then  $x$  is not in the convex hull of the remaining points  $a, c, d$ , as shown in Figure 5(b). In the following, we show that this example can be generalized to arbitrary number of points and arbitrary dimension.

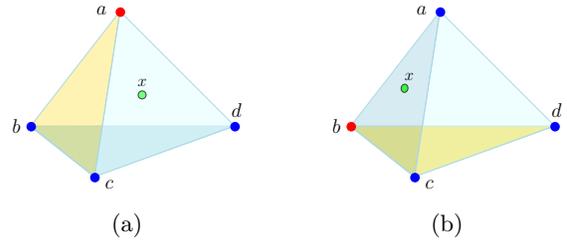


Fig. 5. An example of 4 points in  $\mathbb{R}^3$ . (a) A point  $x$  in the interior of tetrahedron can not be 1-safe because if  $a$  is adversarial, then  $x$  is not contained in the convex hull of the normal points. (b) If a point  $x$  lies on facet, say  $\overline{abc}$ , and  $b$  is adversarial, then  $x$  is not contained in the convex hull of normal point  $a, c, d$ .

**Proposition 4.4** *For a set of  $N_i$  nodes in general positions, if  $F_i \geq \left\lceil \frac{N_i}{d+1} \right\rceil$ , then there exist general examples in which an  $F_i$ -safe point does not exist.*

*Proof:* Let  $a_1, a_2, \dots, a_{d+1}$  be points that lie on the vertices of a non-degenerate  $d$ -simplex in  $\mathbb{R}^d$  where one of the points is adversarial. A point in the interior of simplex can not be 1-safe because it does not lie in the convex hull of normal points for an arbitrary choice of adversarial point. Consider a point  $x$  that lies on a facet of  $d$ -simplex defined by, say, points  $a_1, a_2, \dots, a_d$ . If one of the points, say  $a_1$ , on the vertices of such a facet is adversarial, then point  $x$  can not lie on  $(d-1)$ -simplex or

<sup>3</sup> A non-degenerate triangle has three points that do not all lie on a line.

the convex hull of the remaining points,  $a_2, a_3, \dots, a_{d+1}$ . Thus, such a point can not be a 1-safe point.

Now, if we replace each point on a vertex of  $d$ -simplex by  $\frac{N_i}{d+1}$  points (for simplicity, we assume  $N_i$  to be a multiple of  $d+1$ ), then we get a total of  $N_i$  points that are clustered on  $d+1$  vertices. Any point  $x$  in the interior of simplex can not be  $\frac{N_i}{d+1}$ -safe because if all  $\frac{N_i}{d+1}$  points on an arbitrary vertex of the simplex are adversarial, then the convex hull of the remaining points does not contain the point  $x$ . Let us say we choose a point  $x$  on a facet of the  $d$ -simplex. If  $\frac{N_i}{d+1}$  points lying on a vertex of this facet are adversarial, then the convex hull of the remaining points does not contain  $x$ . Thus, no point  $x$  can be an  $\frac{N_i}{d+1}$ -safe point for this configuration of points. It is easy to see that argument still works if instead of placing all  $\frac{N_i}{d+1}$  points on a vertex of  $d$ -simplex, we place them arbitrary close to the vertex. Thus, the statement holds for points in general positions as well. ■

We note that our proof for the necessary condition for the existence of an  $F_i$ -safe point is based on the proof of the Centerpoint Theorem's sharpness. Similar constructions of extreme examples of pointsets are well-studied in the literature [16,46].

#### 4.3 Sufficient Condition for $F_i$ -Safe Point in Arbitrary Dimension

In the previous section, we showed that if there are at least  $\lceil \frac{N_i}{d+1} \rceil$  adversarial nodes, then an  $F_i$ -safe point does not exist. In this section, we show that for *any*  $d$ , if the number of adversarial nodes  $F_i$  is less than  $\lfloor \frac{N_i}{d+1} \rfloor$ , then an  $F_i$ -safe point always exists. The main idea will be to show that for a set  $\mathcal{S}$  of  $N_i$  points in  $\mathbb{R}^d$ , there exists an interior point of depth  $\lfloor \frac{N_i}{d+1} + 1 \rfloor$ . In other words, there exists an interior point in the intersection of all convex sets containing at least  $\lfloor \frac{N_i}{d+1} + 1 \rfloor$  points of  $\mathcal{S}$ . For this, we first prove some intermediate results.

**Lemma 4.5** *Let  $\mathcal{S}$  be a set of  $N$  points in  $\mathbb{R}^d$  in general positions, where  $d \geq 1$ . For a set of  $d+1$  convex sets each containing at least  $\lfloor \frac{Nd}{d+1} + 1 \rfloor$  points from  $\mathcal{S}$ , there are at least  $d+1$  points of  $\mathcal{S}$  contained in the intersection of all halfspaces.*

*Proof:* Let  $X_1, X_2, \dots, X_{d+1}$  be the convex sets, each containing at least  $k$  points of  $\mathcal{S}$ . We will refer to the number of points of  $\mathcal{S}$  contained in a set  $X_i$  as the *size* of set  $X_i$  and denote it as  $|X_i|$ . Now, for the first set  $X_1$  we have,

$$|X_1| \geq k.$$

Since there are only  $N - k$  points left, the set  $X_2$  must contain at least  $k - (N - k)$  points from  $X_1$ . Thus,

$$|X_1 \cap X_2| \geq 2k - N.$$

Similarly,  $X_3$  must contain at least  $k - (N - |X_1 \cap X_2|)$  points from  $X_1 \cap X_2$ , that is,

$$|X_1 \cap X_2 \cap X_3| \geq 3k - 2N.$$

In general, we have,

$$|X_1 \cap X_2 \cap \dots \cap X_i| \geq k - (N - |X_1 \cap X_2 \cap \dots \cap X_{i-1}|).$$

It follows that,

$$|X_1 \cap X_2 \cap \dots \cap X_{d+1}| \geq (d+1)k - dN.$$

Substituting the value of  $k$ , we get the following,

$$|X_1 \cap X_2 \cap \dots \cap X_{d+1}| \geq (d+1) \left\lceil \frac{Nd}{d+1} + 1 \right\rceil - dN.$$

It follows that

$$|X_1 \cap X_2 \cap \dots \cap X_{d+1}| \geq d+1,$$

which is the desired claim. ■

The above lemma implies that any  $d+1$  convex sets, each of which contains *enough* points from a set  $\mathcal{S}$ , must contain at least  $d+1$  points from the same set. Now, if the original pointset  $\mathcal{S}$  is in general position, then so must be the  $d+1$  points contained in all convex sets. Furthermore, every  $d+1$  points in general positions span a  $d$ -simplex that contains a ball of radius  $\epsilon$  in its interior for sufficiently small value of  $\epsilon > 0$ . Thus, we get the following:

**Corollary 4.6** *Let  $\mathcal{S}$  be a set of  $N$  points in  $\mathbb{R}^d$  in general positions, where  $d \geq 1$ . For a set of  $d+1$  convex sets each containing at least  $\lfloor \frac{Nd}{d+1} + 1 \rfloor$  points from  $\mathcal{S}$ , there exists  $\epsilon > 0$  such that the intersection of all convex sets contains a ball of radius  $\epsilon$ .*

Next, we state the following version of *Helly's Theorem* [47] that will be used in the proof of Theorem 4.8.

**Theorem 4.7** [47] *If  $\mathcal{K}$  is a finite family of convex sets (or an infinite family of compact convex sets) in  $\mathbb{R}^d$ ,  $C$  is a convex set such that for every  $d+1$  members of  $\mathcal{K}$ , there exists a translate<sup>4</sup> of  $C$  contained in their intersection, then there is a suitable translate of  $C$  contained in the intersection of all sets in  $\mathcal{K}$ .*

Now we state a crucial result to prove a sufficient condition for an  $F_i$ -safe point in arbitrary  $d$ .

<sup>4</sup> For a set  $C \subset \mathbb{R}^d$ , a translate of  $C$  is the set  $\{c+x : c \in C\}$  for some vector  $x \in \mathbb{R}^d$ .

**Theorem 4.8** For any set  $\mathcal{S}$  of  $N$  points in  $\mathbb{R}^d$  in general positions, where  $d \geq 1$ , there exists an interior point in the intersection of all convex sets containing at least  $\left\lceil \frac{Nd}{d+1} + 1 \right\rceil$  points of  $\mathcal{S}$ .

*Proof:* From Lemma 4.5 and Corollary 4.6, we know that every  $d + 1$  convex sets contain an  $\epsilon$  radius ball for some  $\epsilon > 0$ . Let  $\epsilon_0$  be the minimum over all such  $\epsilon$  for all choices of  $d + 1$  convex sets.<sup>5</sup> Let  $C$  be a ball of radius  $\epsilon_0$ , and let  $\mathcal{K}$  be the family of all convex sets that contain points of  $\mathcal{S}$ . Then every  $d + 1$  members of  $\mathcal{K}$  contain a translate of  $C$ . It follows from Theorem 4.7 that a suitable translate of  $C$  is contained in the intersection of all sets in  $\mathcal{K}$ . Since  $C$  is a ball of non-zero radius, it has an interior. It follows that intersection of every convex set that contains points from  $\mathcal{S}$  contains a ball with an interior. ■

Next, we present the main result in this subsection, that is, a sufficient condition for the existence of an  $F_i$ -safe point in an arbitrary dimension  $d$ .

**Theorem 4.9** For a set of  $N_i$  nodes in general positions in  $\mathbb{R}^d$ , if the number of adversarial nodes is less than  $\lfloor \frac{N_i}{d+1} \rfloor$ , then there always exists an  $F_i$ -safe point.

*Proof:* From the assumption of Theorem 4.9, we have a set of  $N_i$  points in general position of which there are less than  $\lfloor \frac{N_i}{d+1} \rfloor$  adversarial. If we take the convex hull of the remaining points, such of convex hull will contain at least  $N_i - \left( \lfloor \frac{N_i}{d+1} \rfloor - 1 \right)$  normal points. A family  $\mathcal{K}$  of convex sets each of which contains at least  $\left\lceil \frac{N_i d}{d+1} + 1 \right\rceil$  points from a given set of  $N_i$  must include the one that contains the convex hull of normal points. Theorem 4.8 states that there is a point in the interior of the intersection of family  $\mathcal{K}$ . Therefore, existence of an interior point in the convex hull normal points, i.e., an  $F_i$ -safe point, is implied. This completes the proof. ■

#### 4.4 Improvement in the Resilience of ADRC Consensus Algorithm

If a normal agent  $i$  uses approximate Tverberg partition [43] to compute an  $F_i$ -safe point, as is done in [4], then the algorithm is resilient to  $F_i \leq \frac{N_i}{2^d} - 1$  adversaries in  $\mathcal{N}_i$  as compared to the best possible (theoretical) bound  $F_i \leq \frac{N_i - 1}{d+1}$ . Here, we discuss the usefulness of centerpoint in computing an  $F_i$ -safe point to improve the ADRC algorithm's resilience. The main point is that we can compute a centerpoint exactly in two and three dimensions and can compute it with better approximation

<sup>5</sup> In general, there are infinitely many such sets but we observe that any such set must contain convex hull of some  $d + 1$  points. Thus, it suffices to only consider  $\binom{n}{d+1}$  convex sets, i.e., the convex hulls of any  $d + 1$  points.

(compared to Tverberg partition) in higher dimensions using known algorithms. Next, we state the resilience of the centerpoint-based computation of an  $F_i$ -safe point.

**Proposition 4.10** Given a set of  $N_i$  points in general positions in  $\mathbb{R}^d$ , of which  $F_i < \mathcal{F}_{max}$  are adversarial, then it is possible to compute an  $F_i$ -safe point using centerpoint if

$$\begin{aligned} \mathcal{F}_{max} &= \left\lfloor \frac{N_i}{d+1} \right\rfloor, & \text{for } d = 2, 3, \text{ and} \\ \mathcal{F}_{max} &= \Omega\left(\frac{N_i}{d^2}\right) & \text{for } d > 3. \end{aligned} \quad (4)$$

Moreover, such an  $F_i$ -safe point can be computed in  $O(N)$  and  $O(N^2)$  times in  $d = 2$  and  $3$ , respectively, and in  $O(N^{c \log d} (2d)^d)$  time in  $d > 3$  dimensions.

*Proof:* By Theorem 4.3, we know that an  $F_i$ -safe point is an interior centerpoint of  $N_i$  points, where  $F_i \leq \frac{N_i}{d+1} - 1$ . An interior centerpoint can be computed by taking the centroid of  $d + 1$  distinct centerpoints whenever an interior centerpoint exists. Therefore, we only need algorithms to compute centerpoints. If  $d = 2$ , a centerpoint can be computed in linear time using the algorithm presented in [48]. If  $d = 3$ , a centerpoint can be computed in  $O(N_i^2)$  time by running Chan's algorithm [49] four times and then computing the centroid of the four centerpoints returned. In higher dimensions ( $d \geq 4$ ), the complexity of finding a centerpoint in the general dimension is unknown. However, Miller and Sheehy provided an algorithm in [50] to compute an approximate centerpoint of depth  $\Omega\left(\frac{N}{d^2}\right)$  in  $O(N_i^{c \log d} (2d)^d)$  time. As a result, we can use the algorithm to get an  $F_i$ -safe point with  $F_i < \mathcal{F}_{max} = \Omega\left(\frac{N}{d^2}\right)$ . ■

**Remark 1** By comparing (3) and (4), observe that centerpoint-based computation of  $F_i$ -safe point improves the resilience of ADRC algorithm as compared to the Tverberg partition-based computation. Using centerpoint, we achieve optimal resilience in two and three dimensions, whereas in higher dimensions, the resilience improves from  $\Omega(N_i/2^d)$  (based on Tverberg partition) to  $\Omega(N_i/d^2)$  adversaries in the neighborhood of a normal agent  $i$ .

**Remark 2** The point returned by the algorithm by Miller and Sheehy [50] has a centerpoint depth of  $\Omega\left(\frac{N_i}{d^{r/r-1}}\right)$  for any positive integer  $r > 1$ . For instance, if we consider  $r = 2$ , then we get a point of depth  $\Omega(N_i/d^2)$ . By increasing  $r$ , the quality of approximation, and hence the bound on the number of adversarial agents improves and approaches  $\Omega(N_i/d)$ . However, it comes at the cost of increasing time complexity as the runtime of the algorithm is  $O\left(N_i^{c \log d} (rd)^d\right)$  for an integer  $r > 1$  [50]. Moreover, in another work, Clarkson

et al. has proposed an algorithm in [51] that computes a point of depth  $\Omega(N_i/d^2)$  with high probability in time  $O(\log^2 d \log N_i)$ .

#### 4.5 Comparison of the Centerpoint-based Computation of an $F_i$ -Safe Point with Other Approaches

In [9, Section 6.2], a linear program (LP) is presented to compute an  $F_i$ -safe point, where  $F_i < N_i/(d + 1)$ . The proposed LP has a total of  $\binom{N_i}{N_i - F_i}(d + 1 - N_i - F_i)$  constraints in  $d + \binom{N_i}{N_i - F_i}$  variables. Hence, their algorithm’s time complexity is at least exponential in  $F_i$ . Since adversarial agents can be as many as a fraction of the total agents, the LP complexity in [9] is exponential in  $N_i$ . Consequently, as  $F_i$  grows with  $N_i$ , their algorithm’s time complexity is too high to be practical even for agents in a two-dimensional space. We note that a centerpoint can also be computed exactly (thus, achieving the same resilience in  $O(N_i^{d-1})$  time for  $d \geq 3$  using a better LP method proposed in [49]. This is a significant improvement over the LP in [9] because the time complexity is exponential in  $d$  instead of the number of agents. Nevertheless, even the algorithm in [49] becomes impractical in higher dimensions. Therefore, we propose that an approximate centerpoint be computed in higher dimensions using [50], resulting in the resilience of  $\Omega(N_i/d^2)$ .

In [35], a quadratic program is presented to solve the resilient convex combination problem to compute an  $F_i$ -safe point. The complexity of the algorithm ([35, Section 3.2]) is  $O(d(N_i r)^3)$ , which is linear in  $d$ . Here, the parameter  $r$  is defined (in Section 3.1 of their paper) as  $r = \binom{N_i - \sigma}{N_i - \sigma - F_i}$ . Assuming  $\sigma = 0$ ,<sup>6</sup> the complexity of computing an  $F_i$ -safe point using the quadratic program is  $O(d(N_i^{N_i+1})^3)$ , where a fraction of the agents are allowed to be adversarial. As a result, the time complexity of the quadratic program in [35] is worse than the LP proposed in [9]. Similarly, in [13, Section IV] authors present an algorithm to compute an  $F_i$ -safe point (that they call a “middle point”) in time  $O((pr)^3)$ , where  $p = dF_i + 1$ , and  $r = \binom{(d+1)F_i+1}{F_i}$ . Note that the parameter  $r$  is exponential in the number of adversarial agents.

In summary, the centerpoint-based approach provides fundamental limits on the computation of an  $F_i$ -safe point. An immediate result of this approach is that in general, computation of an  $F_i$ -safe point, where the maximum value of  $F_i$  is  $\Omega(N_i/d)$ , is a challenging problem. Since checking whether a given point is a centerpoint or not is *co-NP-Complete* [52], we deduce that checking

<sup>6</sup> This is so because in our setting, a normal agent is unaware of the identities of the normal and adversarial agents in its neighborhood.

whether a point is an  $F_i$ -safe point or not in general is also *co-NP-Complete*. Further, our centerpoint approach and the proposed algorithms offer improvements in time complexity over the algorithms in [9,13,35].

## 5 Numerical Evaluation

We perform simulations to illustrate resilient consensus in multirobot systems in two dimensions using centerpoint, and compare it with the one using approximate Tverberg partition [4].<sup>7</sup> We model interconnections between robots using the following graphs:

- A disk graph  $\mathcal{G}_d = (\mathcal{V}, \mathcal{E}(t))$ , in which each node  $i \in \mathcal{V}$  (representing a robot) has a sensing radius  $r$  and  $(j, i) \in \mathcal{E}(t)$  if and only if  $\|x_j(t) - x_i(t)\| \leq r$ .
- A fixed undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  whose edge set does not change over time.

At each iteration  $t$  of the consensus algorithm, a normal robot  $i$  computes an  $F_i$ -safe point  $s_i(t)$  of its neighbors’ positions (using centerpoint or approximate Tverberg partition), and calculates its new position using (1). In our experiments, we set  $\alpha_i(t) = 0.8$ , for all  $t$ .

### 5.1 Centerpoint based resilient consensus

We simulate a group of 120 robots, out of which 100 are normal and 20 are adversarial. They are deployed in a planar region  $\mathcal{W} = [-1, 1] \times [-1, 1] \in \mathbb{R}^2$  as shown in Figure 6(a), where nodes in blue are normal robots executing consensus and nodes in red represent adversarial robots. We consider three types of adversarial behaviors:

- *Stationary* – each red node has a fixed position that does not change throughout the simulation.
- *Oscillating* – each red node changes its position by moving from corner to corner within the square of length 0.1, as depicted in Figure 6(b).
- *Move-away* – each red node moves towards the closest corner of the region  $\mathcal{W}$  from its initial position and then stays there, as shown in Figure 6(c).

The interconnection topology is captured by a disk graph with a fixed sensing radius of  $r = 0.45$  for every normal robot. Each normal robot executes consensus algorithm based on centerpoint. Consensus is guaranteed if the number of adversarial robots in the neighborhood of each normal robot  $i$  is  $F_i \leq (\lfloor \frac{N_i}{3} \rfloor - 1)$  and the sequence of graphs induced by normal agents is repeatedly reachable. Both of these conditions are satisfied in all the simulations here. Consequently, normal robots achieve consensus in the presence of all three types of adversarial nodes, as shown in Figures 7 and 8.

<sup>7</sup> Our code is available at <https://github.com/JianiLi/MultiRobotsRendezvous>

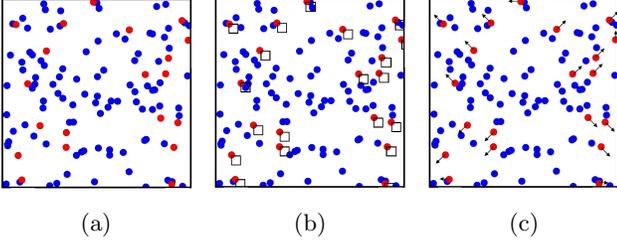


Fig. 6. Robots’ initial positions with (a) stationary, (b) oscillating and (c) move-away adversaries.

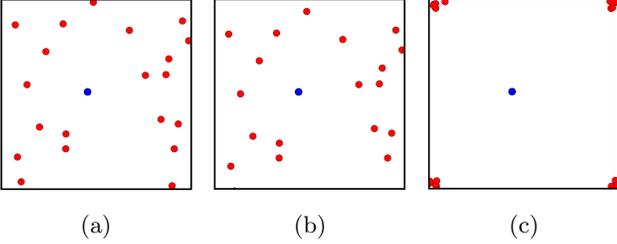


Fig. 7. Robots’ positions after consensus with (a) stationary, (b) oscillating, and (c) move-away adversaries.

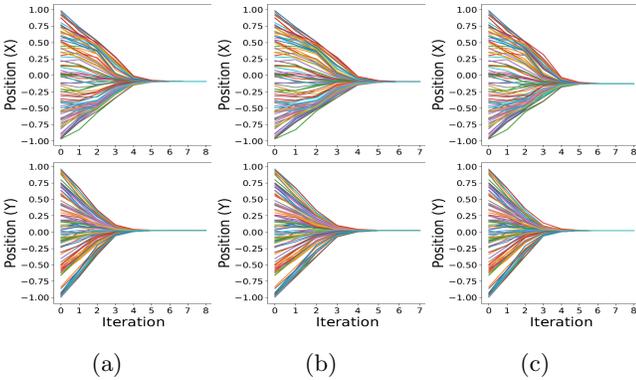


Fig. 8. Normal robots’ positions as functions of iterations in the case of (a) stationary, (b) oscillating, and (c) move-away adversaries.

### 5.2 Comparison of Approximate Tverberg Partition and Centerpoint-based Resilient Consensus

To demonstrate the improved resilience of centerpoint-based consensus compared to the Tverberg partition-based one, we consider stationary adversarial nodes and assume that the network graphs are fixed. In our first example, we consider a group of 28 robots deployed in a planar region  $\mathcal{W} = [-1.5, 1.5] \times [-0.375, 0.375] \in \mathbb{R}^2$  as shown in Figure 9. There are six adversarial nodes (red) on the left and six on the planar region’s right side. At the same time, 16 normal robots (blue) are divided equally into two clusters in the middle part of the region, each containing eight robots. All of the 16 normal robots are pair-wise adjacent to each other. Moreover, each normal robot is adjacent to six adversarial robots located on its side. Figures 9 shows the evolution of robots’ positions using approximate Tverberg partition-

based and centerpoint-based consensus algorithms. As illustrated, the approximate Tverberg partition-based algorithm fails to make the normal robots converge to one point, but the centerpoint-based algorithm succeeds. The main reason is that each normal robot  $i$  has  $N_i = 22$  robots in its neighborhood (including the robot itself), of which 6 are adversarial. The approximate Tverberg partition-based algorithm is resilient to  $(\lceil \frac{N_i}{4} \rceil - 1) = 5$  adversaries, whereas the centerpoint based algorithm is resilient to  $(\lfloor \frac{N_i}{3} \rfloor - 1) = 6$  adversaries.

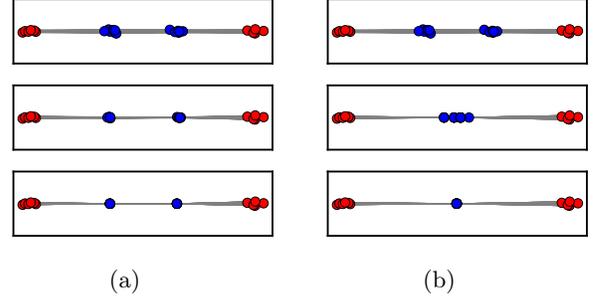


Fig. 9. The evolution of robots’ positions (from top to down) in the case of (a) approximate Tverberg partition-based and the (b) centerpoint-based algorithms.

## 6 Conclusion

To achieve resilient vector consensus, normal agents in a network depend on a scheme whereby they compute a state that is a convex combination of their normal neighbors’ states. We discussed the merits of using the notion of centerpoint for this purpose. For a normal agent  $i$ , we provided a geometric characterization—using centerpoint—of states that are guaranteed to lie in the convex hull of agent  $i$ ’s normal neighbors’ states when the number of adversarial agents is limited to a  $1/(d+1)$  fraction of the size of the neighborhood of  $i$ . It also followed that the upper bound on the number of adversarial agents is best possible in the worst case. We proposed to use well-known efficient algorithms to compute exact centerpoints in two and three dimensions. For higher dimensions, by using the approximate centerpoint algorithm proposed in [50], we showed that the resilience of vector consensus algorithm to adversarial agents in the neighborhood of a normal agent  $i$  could be significantly improved from  $\Omega(N_i/2^d)$ , which is due to approximate Tverberg partition [4], to  $\Omega(N_i/d^2)$ . The centerpoint-based aggregation of data in the presence of adversarial agents can be used in a various other applications, including resilient distributed diffusion and learning. In the future, we aim to further improve the resilience of vector consensus algorithms by introducing trusted and diverse agents within the network [2,53].

## References

- [1] M. Shabbir, J. Li, W. Abbas, and X. Koutsoukos, “Resilient vector consensus in multi-agent networks using centerpoints,”

- in *American Control Conference*, pp. 4387–4392, 2020.
- [2] 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, pp. 2036–2048, 2018.
  - [3] 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, 2019.
  - [4] H. Park and S. A. Hutchinson, “Fault-tolerant rendezvous of multirobot systems,” *IEEE Transactions on Robotics*, vol. 33, pp. 565–582, 2017.
  - [5] S. Sundaram and B. Ghahsifard, “Consensus-based distributed optimization with malicious nodes,” in *53rd Annual Allerton Conference on Communication, Control, and Computing*, 2015.
  - [6] L. Su and N. Vaidya, “Multi-agent optimization in the presence of Byzantine adversaries: Fundamental limits,” in *2016 American Control Conference*, pp. 7183–7188, 2016.
  - [7] L. Tseng and N. Vaidya, “Iterative approximate Byzantine consensus under a generalized fault model,” in *International Conference on Distributed Computing and Networking*, pp. 72–86, Springer, 2013.
  - [8] 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.
  - [9] H. Mendes, M. Herlihy, N. Vaidya, and V. K. Garg, “Multidimensional agreement in byzantine systems,” *Distributed Computing*, vol. 28, no. 6, pp. 423–441, 2015.
  - [10] N. H. Vaidya and V. K. Garg, “Byzantine vector consensus in complete graphs,” in *ACM Symposium on Principles of Distributed Computing*, pp. 65–73, 2013.
  - [11] N. H. Vaidya, “Iterative Byzantine vector consensus in incomplete graphs,” in *International Conference on Distributed Computing and Networking*, pp. 14–28, 2014.
  - [12] H. Park and S. Hutchinson, “Worst-case performance of rendezvous networks in the presence of adversarial nodes,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.
  - [13] J. Yan, X. Li, Y. Mo, and C. Wen, “Resilient multi-dimensional consensus in adversarial environment,” *arXiv preprint arXiv:2001.00937*, 2020.
  - [14] J. Yan, Y. Mo, X. Li, and C. Wen, “A “safe kernel” approach for resilient multi-dimensional consensus,” *arXiv preprint arXiv:1911.10836*, 2019.
  - [15] M. Shabbir, *Some results in computational and combinatorial geometry*. PhD thesis, Rutgers University, 2014.
  - [16] 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. 3, pp. 163–185, 2015.
  - [17] D. Dolev, N. A. Lynch, S. S. Pinter, E. W. Stark, and W. E. Weihl, “Reaching approximate agreement in the presence of faults,” *Journal of the ACM*, vol. 33, no. 3, pp. 499–516, 1986.
  - [18] R. Kieckhafer and M. Azadmanesh, “Low cost approximate agreement in partially connected networks,” *Journal of Computing and Information*, vol. 3, no. 1, pp. 53–85, 1993.
  - [19] R. M. Kieckhafer and M. H. Azadmanesh, “Reaching approximate agreement with mixed-mode faults,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 1, pp. 53–63, 1994.
  - [20] Z. Bouzid, M. G. Potop-Butucaru, and S. Tixeuil, “Byzantine-resilient convergence in oblivious robot networks,” in *International Conference on Distributed Computing and Networking*, pp. 275–280, 2009.
  - [21] Z. Bouzid, M. G. Potop-Butucaru, and S. Tixeuil, “Optimal Byzantine-resilient convergence in uni-dimensional robot networks,” *Theoretical Computer Science*, vol. 411, no. 34–36, pp. 3154–3168, 2010.
  - [22] D. Saldaña, L. Guerrero-Bonilla, and V. Kumar, “Resilient backbones in hexagonal robot formations,” in *Distributed Autonomous Robotic Systems*, pp. 427–440, Springer, 2019.
  - [23] 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.
  - [24] 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.
  - [25] S. M. Dibaji and H. Ishii, “Resilient consensus of second-order agent networks: Asynchronous update rules with delays,” *Automatica*, vol. 81, pp. 123–132, 2017.
  - [26] 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)*, pp. 252–258, IEEE, 2017.
  - [27] J. Usevitch and D. Panagou, “Resilient leader-follower consensus to arbitrary reference values in time-varying graphs,” *IEEE Transactions on Automatic Control*, vol. 65, no. 4, pp. 1755–1762, 2019.
  - [28] H. Zhang and S. Sundaram, “A simple median-based resilient consensus algorithm,” in *50th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1734–1741, 2012.
  - [29] W. Fu, Q. Ma, J. Qin, and Y. Kang, “Resilient consensus-based distributed optimization under deception attacks,” *International Journal of Robust and Nonlinear Control*.
  - [30] N. Ravi, A. Scaglione, and A. Nedić, “A case of distributed optimization in adversarial environment,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5252–5256, 2019.
  - [31] S. Sundaram and B. Ghahsifard, “Secure local filtering algorithms for distributed optimization,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 1871–1876, IEEE, 2016.
  - [32] S. Sundaram and B. Ghahsifard, “Distributed optimization under adversarial nodes,” *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1063–1076, 2018.
  - [33] Z. Yang and W. U. Bajwa, “ByRDIE: Byzantine-resilient distributed coordinate descent for decentralized learning,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 4, pp. 611–627, 2019.
  - [34] K. Kuwarananchaoren, L. Xin, and S. Sundaram, “Byzantine-resilient distributed optimization of multi-dimensional functions,” in *American Control Conference*, pp. 4399–4404, 2020.
  - [35] X. Wang, S. Mou, and S. Sundaram, “A resilient convex combination for consensus-based distributed algorithms,” *Numerical Algebra, Control and Optimization*, vol. 9, no. 3, pp. 269–281, 2019.
  - [36] J. Li, W. Abbas, M. Shabbir, and Koutsoukos, “Resilient distributed diffusion for multi-robot systems using centerpoint,” in *Robotics: Science and Systems (RSS)*, 2020.

- [37] A. H. Sayed, S.-Y. 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.
- [38] H. Tverberg, "A generalization of Radon's theorem," *Journal of the London Mathematical Society*, vol. 1, no. 1, 1966.
- [39] J. R. Reay, "An extension of Radon's theorem," *Illinois J. Math.*, vol. 12, pp. 184–189, 06 1968.
- [40] J.-P. Roudneff, "Partitions of points into intersecting tetrahedra," *Discrete Mathematics*, vol. 81, pp. 81–86, 1990.
- [41] J.-P. Roudneff, "New cases of Reay's conjecture on partitions of points into simplices with k-dimensional intersection," *European Journal of Combinatorics*, vol. 30, no. 8, 2009.
- [42] S. Har Peled and T. Zhou, "Improved approximation algorithms for tverberg partitions," *arXiv preprint arXiv:2007.08717*, 2020.
- [43] W. Mulzer and D. Werner, "Approximating tverberg points in linear time for any fixed dimension," *Discrete & Computational Geometry*, vol. 50, no. 2, pp. 520–535, 2013.
- [44] J. Matoušek, *Lectures on Discrete Geometry*. Springer, 2002.
- [45] R. Rado, "A theorem on general measure," *Journal of the London Mathematical Society*, vol. 1, no. 4, 1946.
- [46] B. Neumann, "On an invariant of plane regions and mass distributions," 1945.
- [47] V. Klee, "The critical set of a convex body," *American Journal of Mathematics*, vol. 75, no. 1, pp. 178–188, 1953.
- [48] 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.
- [49] T. M. Chan, "An optimal randomized algorithm for maximum Tukey depth," in *Proceedings of the 15th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 430–436, SIAM, 2004.
- [50] G. L. Miller and D. R. Sheehy, "Approximate centerpoints with proofs," *Computational Geometry*, vol. 43, no. 8, pp. 647–654, 2010.
- [51] K. L. Clarkson, D. Eppstein, G. L. Miller, C. Sturtivant, and S.-H. Teng, "Approximating center points with iterative radon points," *International Journal of Computational Geometry & Applications*, vol. 6, no. 3, pp. 357–377, 1996.
- [52] Shang Hua Teng, *Points, sphere, and separators: A unified geometric approach to graph partitioning*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1991.
- [53] A. Mitra, W. Abbas, and S. Sundaram, "On the impact of trusted nodes in resilient distributed state estimation of LTI systems," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4547–4552, 2018.