Resilient Vector Consensus using Centerpoint

M. Shabbir[1], J. Li[2], W. Abbas[2], and X. Koutsoukos[2]

[1]*Information Technology University, Lahore, Pakistan*

[2]*Vanderbilt University, Nashville, TN*

ACC2020 · DENVER

American Control Conference, 2020

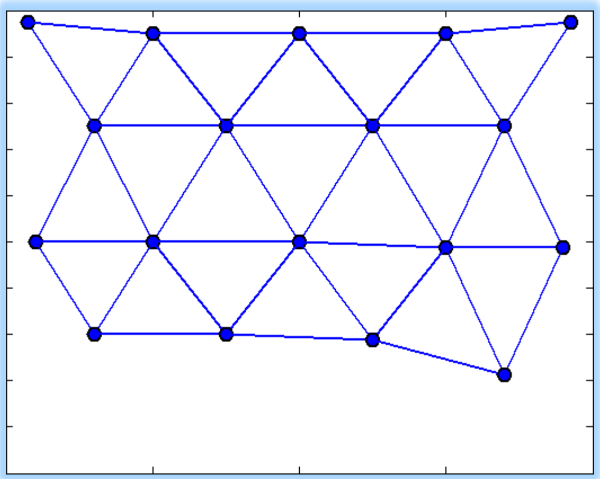
Vector Consensus

A network of agents modeled by an undirected graph $G = (V,E)$.

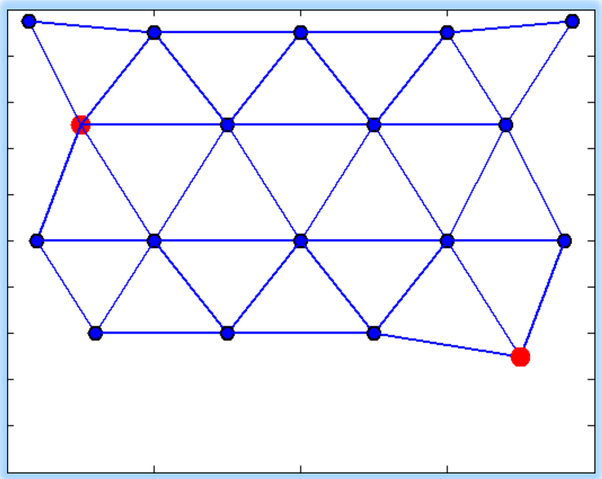
State of agent *i* at time *t* is $x_i(t) \in \mathbb{R}^d$, where $d \geq 2$.

Applications

- Control of moving **vehicles (UAVs)**
- Information processing in **sensor networks**
- Design of **distributed optimization** algorithms
- Parameter **estimation** etc.



Consensus

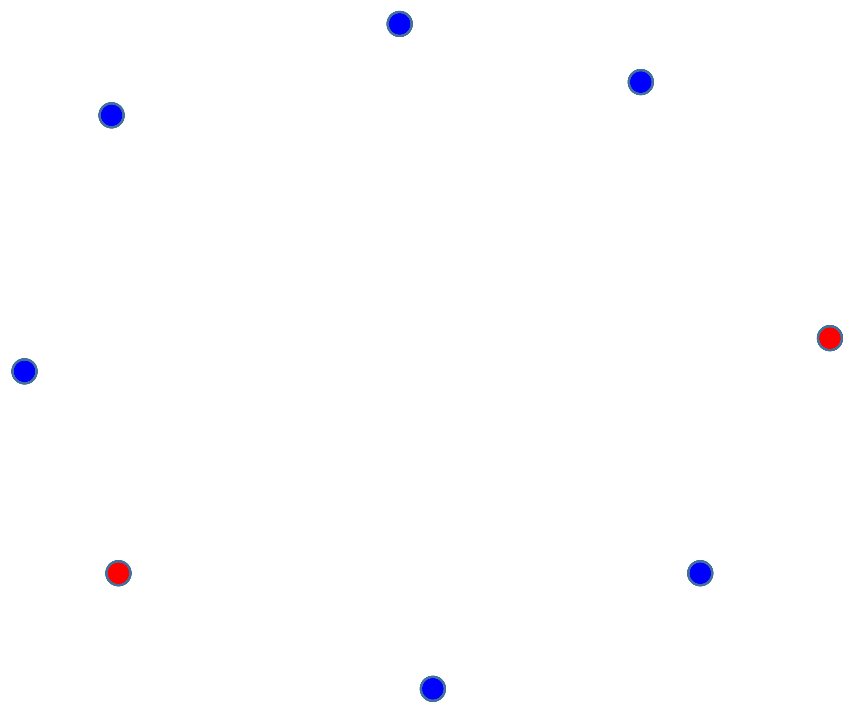


No Consensus due to **adversaries**

How can we design a ***resilient vector consensus*** algorithm?

Resilient Vector Consensus

The state of agent *i* at time *t* is $x_i(t) \in \mathbb{R}^d$, where $d \geq 2$.



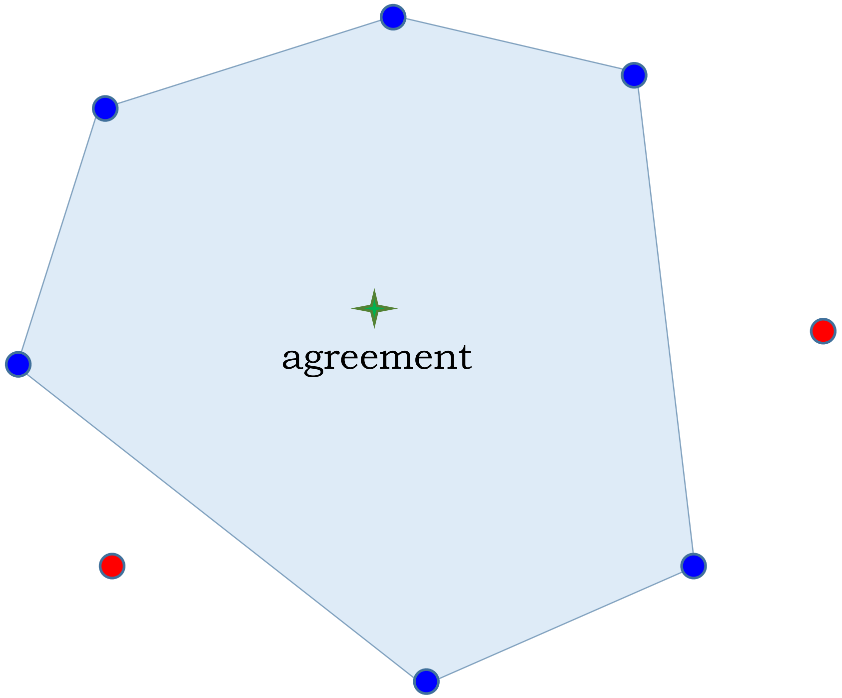
Agents' initial positions

Blue → normal

Red → adversary

Resilient Vector Consensus

The state of agent *i* at time *t* is $x_i(t) \in \mathbb{R}^d$, where $d \geq 2$.



Convex hull of normal agents' initial positions

Safety:

At all times, every normal agent should *remain inside the convex hull* of all normal agents' initial positions.

Agreement:

All normal agents should eventually *converge* at a common point.

Blue → normal

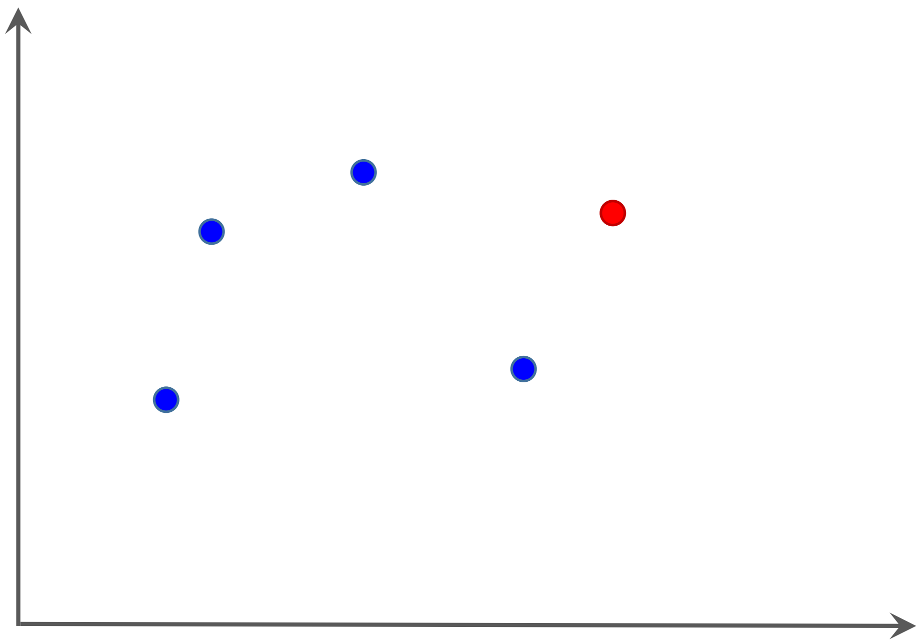
Red → adversary

Can We Use Resilient Scalar Consensus?

There are well studied resilient scalar consensus ($x_i(t) \in \mathbb{R}$) algorithms.[1]

First Approach:

Implement scalar resilient consensus algorithm in each dimension separately.



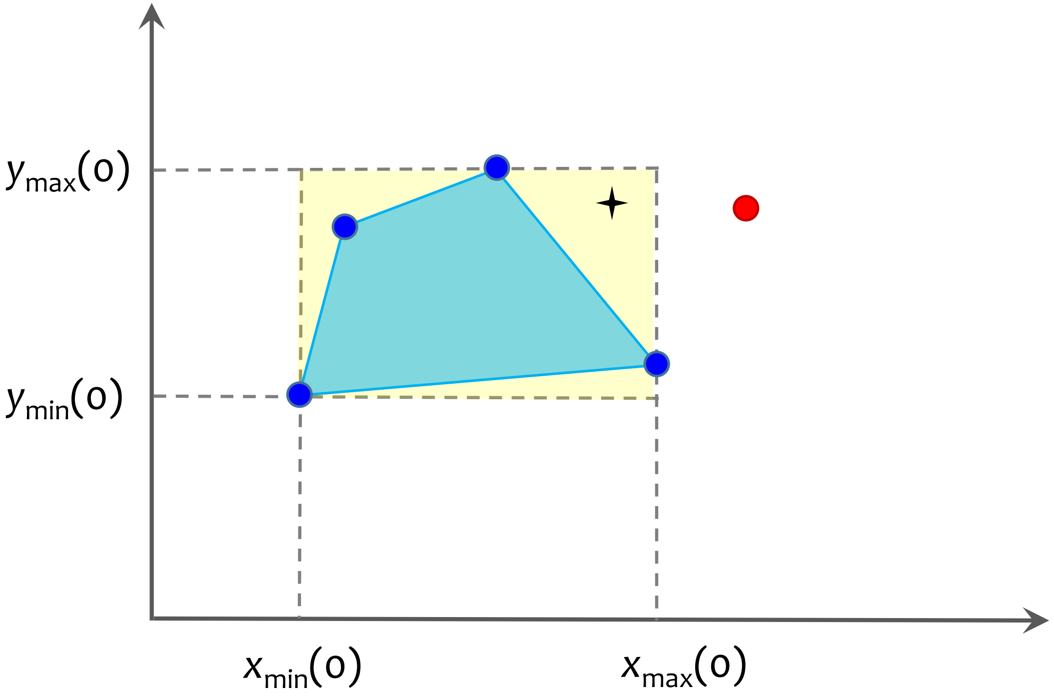
[1] H. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J Sel. Areas Comm., 2013.*

Can We Use Resilient Scalar Consensus?

There are well studied resilient scalar consensus ($x_i(t) \in \mathbb{R}$) algorithms.[1]

First Approach:

Implement scalar resilient consensus algorithm in each dimension separately.



Normal agents can end up converging outside of the convex hull of their initial positions.

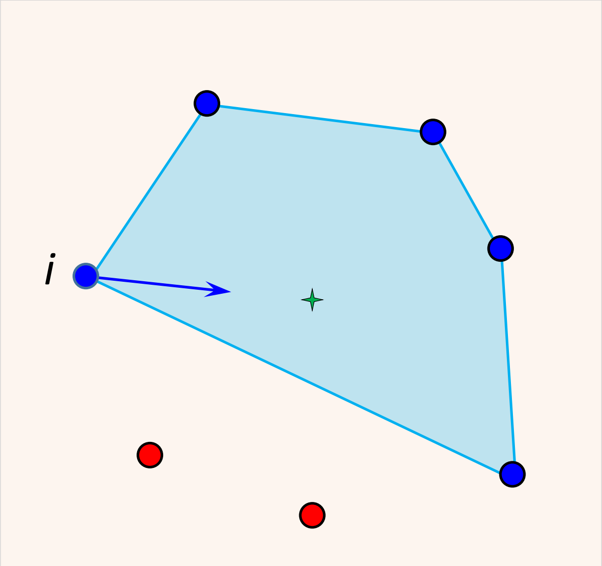
Implementing multiple instances of scalar resilient consensus does not work.

[1] H. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J Sel. Areas Comm.*, 2013.

Approximate Distributed Robust Convergence (ADRC)

ADRC is resilient vector consensus algorithm proposed by Park and Hutchinson.[2]

1. In each iteration *t*, a normal agent *i* finds a point $s_i(t)$ that lies in the convex hull of its normal neighbors' states.
2. Agent *i* updates its state by moving towards $s_i(t)$.

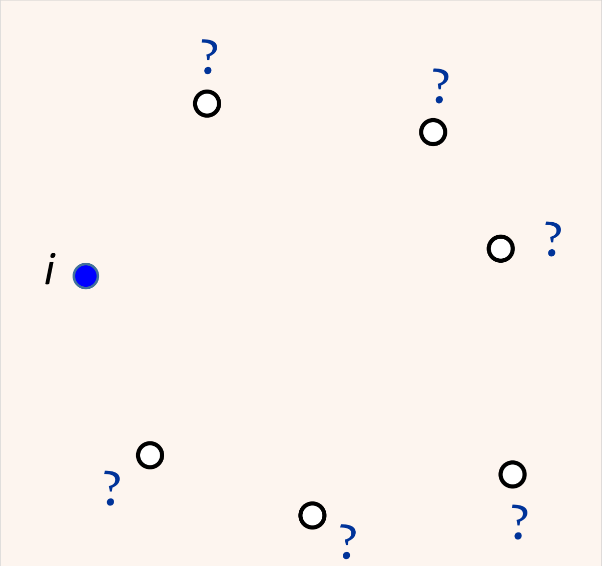


[2]H. Park and S. Hutchinson, "Fault-tolerant rendezvous of multirobot systems," *IEEE Trans. Robotics, 2017.*

Approximate Distributed Robust Convergence (ADRC)

ADRC is resilient vector consensus algorithm proposed by Park and Hutchinson.[2]

1. In each iteration *t*, a normal agent *i* finds a point $s_i(t)$ that lies in the convex hull of its normal neighbors' states.
2. Agent *i* updates its state by moving towards $s_i(t)$.



Challenge:

A normal agent doesn't know who is normal/adversary in its neighborhood.

Safe point

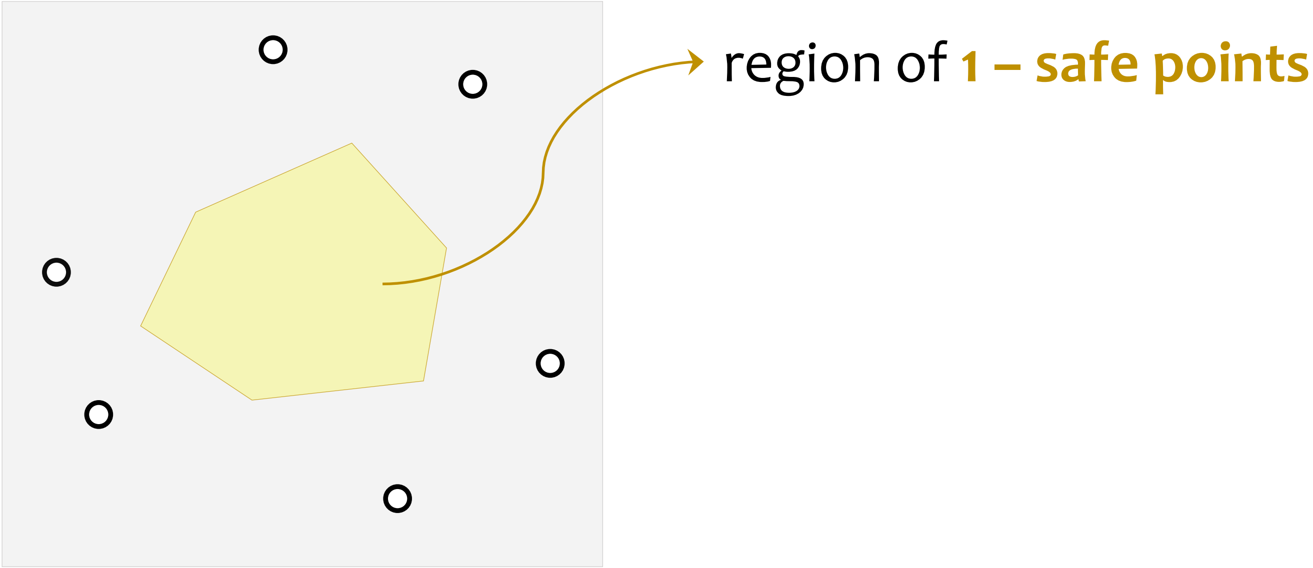
[2]H. Park and S. Hutchinson, "Fault-tolerant rendezvous of multirobot systems," *IEEE Trans. Robotics, 2017.*

F – Safe Points

F – Safe Point:

Given a set of *N* points in R^d, of which *any* of the *F* points can be adversarial (corresponding to adversarial agents).

Then, a point that is guaranteed to lie in the convex hull of (*N* – *F*) normal points is an *F* – Safe point.



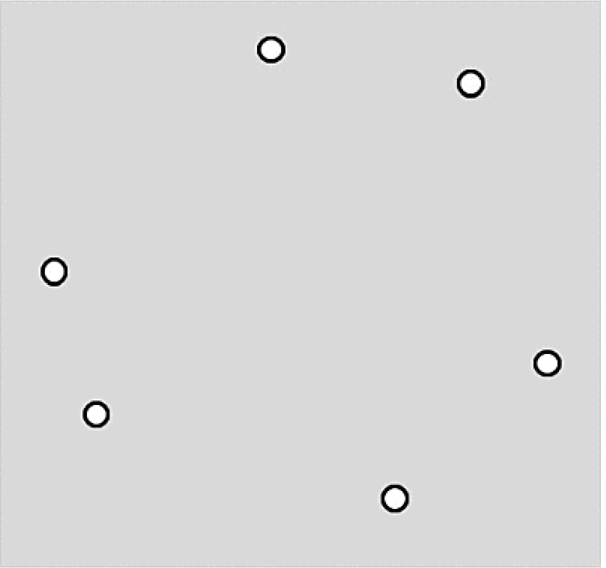
N = 6, d = 2, F = 1

F – Safe Points

F – Safe Point:

Given a set of N points in R^d, of which *any* of the F points can be adversarial (corresponding to adversarial agents).

Then, a point that is guaranteed to lie in the convex hull of (N – F) normal points is an F – Safe point.



N = 6, d = 2, F = 1

- normal
- adversary

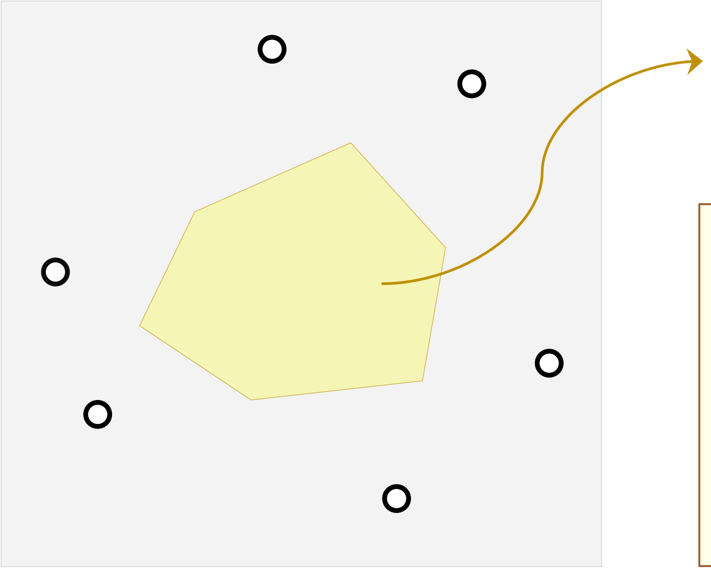
1 – safe region (yellow) always lies in the **convex hull (blue)** of normal nodes, regardless of the selection of the adversary node.

F – Safe Points

F – Safe Point:

Given a set of N points in R^d, of which *any* of the F points can be adversarial (corresponding to adversarial agents).

Then, a point that is guaranteed to lie in the convex hull of $(N - F)$ normal points is an F – Safe point.



region of **1 – safe points**

N = 6, d = 2, F = 1

Challenges:

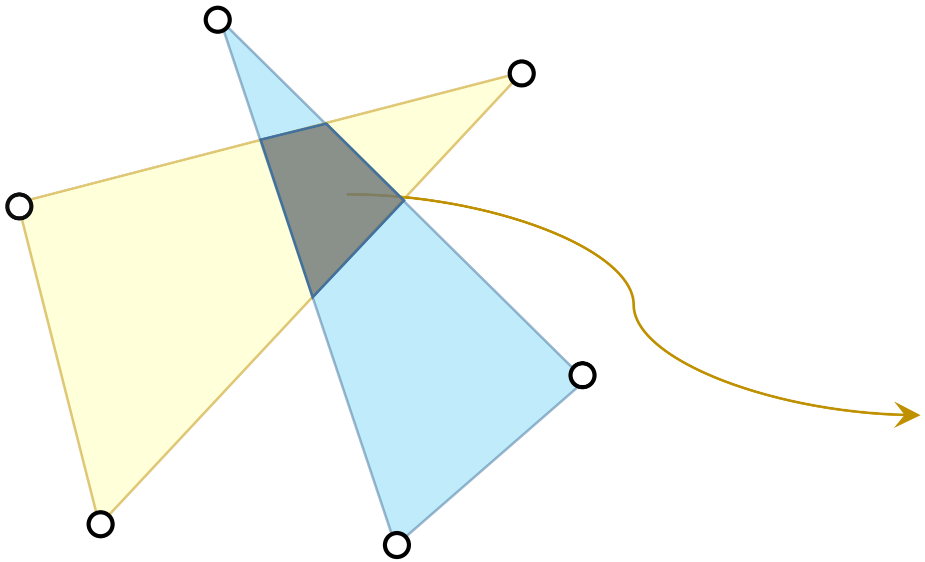
- When can we guarantee ***existence*** of an F – safe point?
- How can we ***find*** it?

Safe Point Using Tverberg Partition (TP)

Park and Hutchinson[1] used **Tverberg partitions (TP)**[2] to compute safe points.

Basic Idea of TP:

Partition points into subsets such that their convex hulls have a non-empty intersection.



Let,
S = no. of subsets in the partition
F = no. of adversary nodes.

$F \leq S - 1$

implies that the intersection contains **F – safe points.**

To compute a safe point, find a point in the intersection.

[1] H. Park and S. Hutchinson, "Fault-tolerant rendezvous of multirobot systems," *IEEE Trans. Robotics, 2017.*

[2] H. Tverberg, "A generalization of Radon's theorem," *J. of the London Math. Society, 1966.*

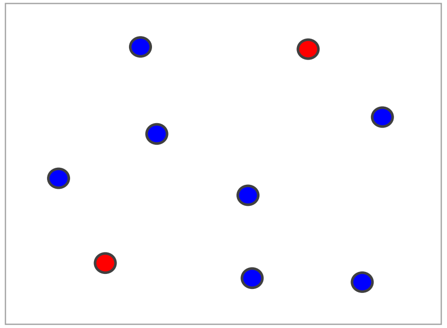
Safe Point Using Tverberg Partition (TP)

Let,

d = dimension of state

N_i = total no. of nodes in the neighborhood of agent i

F_i = no. of adversary agents in the neighborhood of i.



$d = 2$, $N_i = 9$, $F_i = 2$

A **sufficient** condition for the **existence** of an F_i – safe point is

$$F_i \leq \frac{N_i}{d+1} - 1$$

A normal agent can **compute** an F_i – safe point using **TP** in $d^{O(1)} N_i$ time if

$$F_i \leq \left\lceil \frac{N_i}{2^d} \right\rceil - 1$$

$$d \leq 8$$

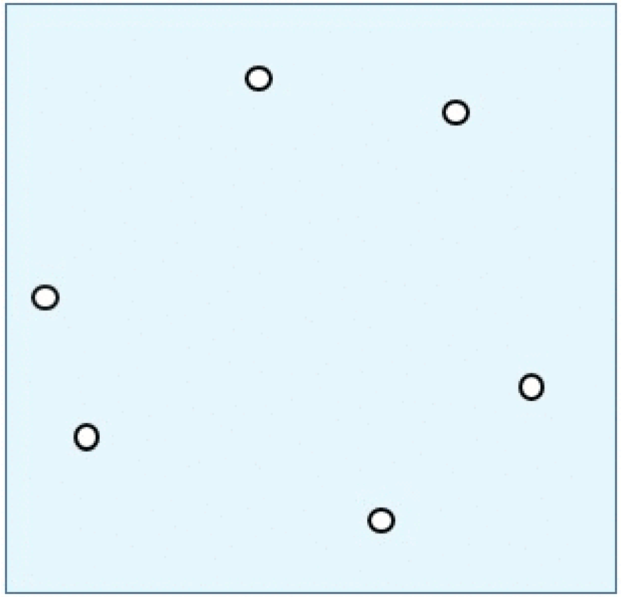
Questions

- **Necessary** condition?
- Can we **improve** the (practical) resilience bound?
- What if ***d* > 8**?

Safe Point Using Centerpoint (CP)

We utilize the notion of centerpoint from discrete geometry.

Centerpoint: For any set S of N points in R^d, a centerpoint c of the set S is a point (not necessarily in S) such that each halfspace containing c contains at least $\frac{N}{d+1}$ points of S.

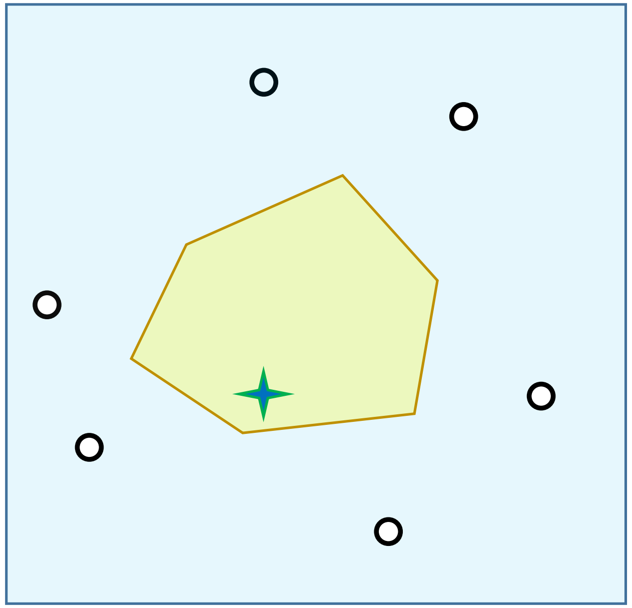


$N = 6, \quad d = 2$

Safe Point Using Centerpoint (CP)

We utilize the notion of centerpoint from discrete geometry.

Centerpoint: For any set *S* of *N* points in R^d, a centerpoint *c* of the set *S* is a point (not necessarily in *S*) such that each halfspace containing *c* contains at least $\frac{N}{d+1}$ points of S.



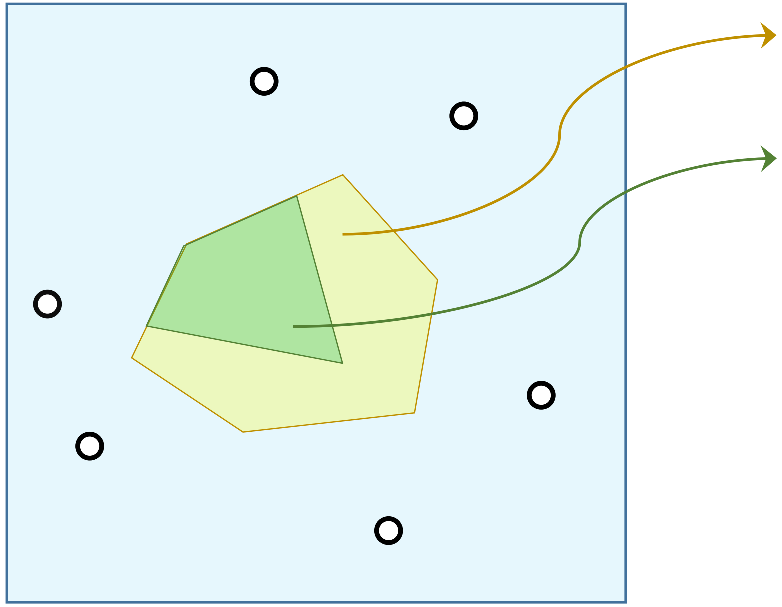
$N = 6, \quad d = 2$

- CP can be viewed as a *generalization of median* in higher dimensions.
- CP *always exists* (CP Theorem).
- CP is *not unique* (CP region).

Safe Point Using Centerpoint (CP)

Theorem: For a set of N points in R^d and $F = \frac{N}{d+1} - 1$, the region of F – safe points is same as the centerpoint region.

(a point is F – safe *if and only if* it is a centerpoint)



1 – safe region ↔ *centerpoint region*

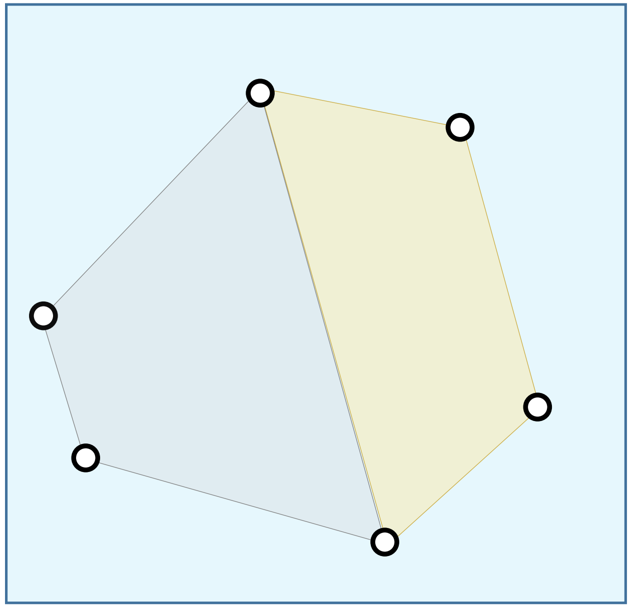
Tverberg points region (not unique)

$N = 6, \quad d = 2, \quad F = 1$

Centerpoint provides a complete characterization of *F*-safe points, whereas Tverberg partitions do not.

Safe Point Using Centerpoint (CP)

For a set of N points in R^d (general positions) and $F \geq \frac{N}{d+1}$, there exist general examples in which an F – safe point does not exist.



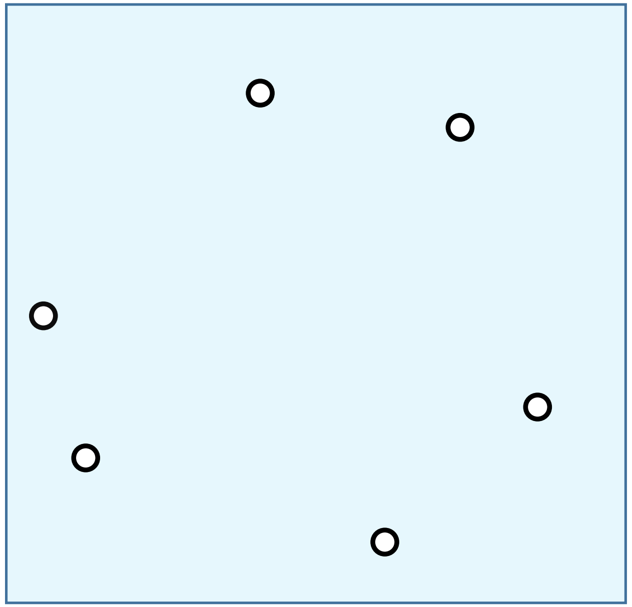
$N = 6$, $d = 2$, $F = 2$

There is no 2 – safe point.

(Why? There are two sets with 4 points each such that their convex hulls have an empty intersection.)

Safe Point Using Centerpoint (CP)

For a set of N points in R^d (general positions) and $F \geq \frac{N}{d+1}$, there exist general examples in which an F – safe point does not exist.



$N = 6, \quad d = 2, \quad F = 2$

There is no 2 – safe point.

(Why? There are two sets with 4 points each such that their convex hulls have an empty intersection.)

A **necessary** condition for the **existence** of an F – safe point is

$$F \leq \frac{N}{d+1} - 1$$

(Previously, we only had a **sufficient** condition.)

Safe Point Computation Using (CP)

Using known results for the centerpoint computation, we can compute an F – safe point if

d = 2 , 3 : $F \leq \frac{N}{d+1} - 1$

d > 3 : $F = \Omega\left(\frac{N}{d^{\frac{r}{r-1}}}\right)$ for any integer r.

Moreover, the time complexity of computing an F – safe point in

- d = 2 is $O(N)$,
- d = 3 is $O(N^2)$, and
- d > 3 is $O\left(N^{c\log d}(rd)^d\right)$ for any integer r.

These bounds are **better** than the ones obtained by using Tverberg partition.

$$F \leq \left\lceil \frac{N}{2^d} \right\rceil - 1$$

ADRC Using Centerpoint

Using centerpoints improve the resilience of ADRC algorithm as compared to Tverberg partition.

(N_i = total no. of agents in the neighborhood of a normal agent i.)

***d* = 2, 3**

CP achieves the *theoretical bound*, that is, ADRC is resilient to $\left(\frac{N_i}{d+1} - 1\right)$ Byzantine adversaries in the neighborhood of agent *i*.

***d* > 3**

Centerpoint: resilient to $\Omega\left(\frac{N_i}{d^2}\right)$ Byzantine adversaries in the neighborhood of *i*.

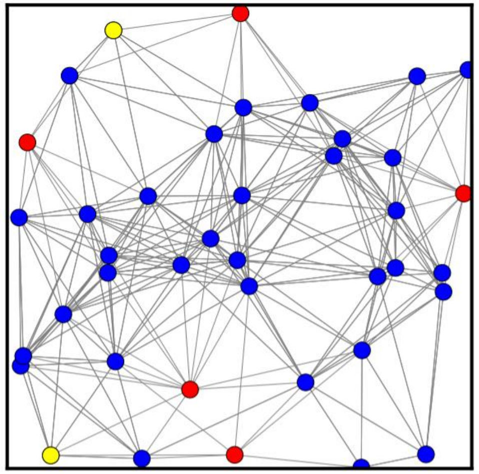
Tverberg: resilient to $\Omega\left(\frac{N_i}{2^d}\right)$ Byzantine adversaries in the neighborhood of *i*.

Simulations

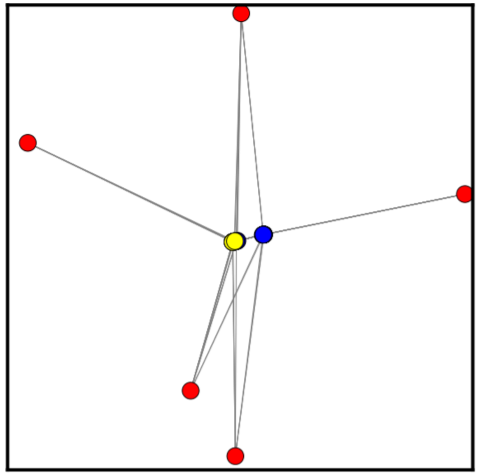
45 robots in a plane (d = 2),

- 5 robots are adversarial (red)
- normal (blue)
- normal robots having $\left(\left\lceil \frac{N_i}{4}\right\rceil - 1\right) < F_i \le \left(\left\lceil \frac{N_i}{3}\right\rceil - 1\right)$ adversaries in their neighborhoods. (yellow)

More adversaries than allowed by the Tverberg-based bound

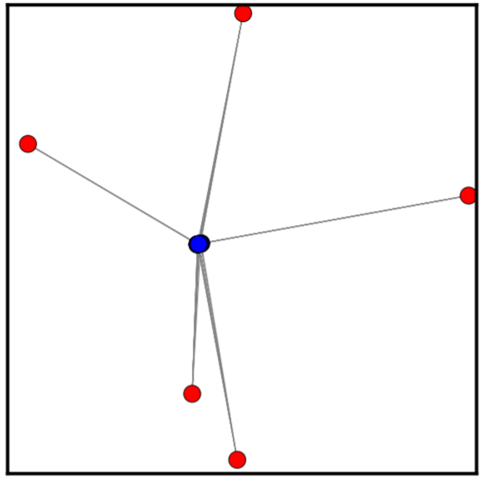


Initial positions



Final positions (**Tverberg** based algo)

No Consensus



Final positions (**Centerpoint** based algo)

Consensus

Conclusions

Resilient vector consensus using **Centerpoint**

Complete characterization of F – safe points

Necessary & sufficient conditions

Improvement in resilience (practically)

Generalization $d > 8$

Extension

Thank You