Optimal Circle Search Despite the Presence of Faulty Robots

K. Georgiou[1], E. Kranakis[2], N. Leonardos[3], A. Pagourtzis[4], I. Papaioannou[4]

September 12, 2019

[1]Ryerson University
[2]Carleton University
[3]University of Athens
[4]National Technical University of Athens

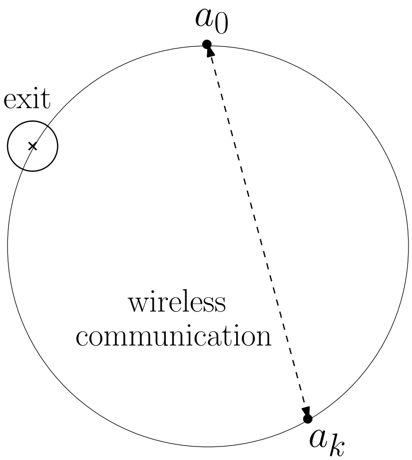
Search problem

Searching an environment to find an exit (or target) in the minimum possible time:

- Autonomous robots (mobile agents)
- (may) cooperate
- exchange messages

Searching for an exit placed at an unknown location:

- on the perimeter of a unit circle
- wireless communication
- crash/byzantine faults



Computational model

Communication

- wirelessly and instantaneously
- anytime and regardless of distance
- message tagged with robot's unique ID (cannot be altered)

Robot movement

- start possition: center of the circle
- maximum speed is 1 (same for all robots)
- recognize and move along the perimeter of the circle
- recognize exit if they are at its location
- allowed to take "shortcuts" by moving in the interior of the disk
- trajectories known, and can be deduced

Computational model

Fault types

- Crash faults
  - stop functioning at any time
  - permanently remains idle and/or fails to communicate
- Byzantine faults
  - may alter trajectory
  - provide (or hide) information to confuse the rest
  - can exhibit behavior of a crash-faulty robot

Adversary

- controls the location of the exit
- controls the behaviour of the malicious robot
- goal to maximize the resulting search completion time

Related Work

Linear Search

Single mobile agent searching for an exit placed at an unknown location on an infinite line. *(Cow Path Problem)*

- *Bellman (1963)*, *Beck (1964)*, *Baeza-Yates et al. (1993)*
- *Ahlswede and Wegener*, *Alpern and Gal*, *Stone*

Presence of faulty robots

- Crash-faulty: *Czyzowicz, Kranakis et al. (2016)*
- Byzantine-faulty: *Czyzowicz, Georgiou et al. (2016)*

Circle Search

- Introduced as an *evacuation* problem (minimize evacuation time) and analyzed in *Czyzowicz et al. (2014)*.
- *Czyzowicz et al. (2017)* investigate *evacuation* in the presence of crash and/or Byzantine faults.

Search with faults

A search is complete if:

- exit visited by a non-faulty robot
- the rest of the agents can be convinced of the location of the exit

(n, f)-search on a circle:
$n > 1$ robots searching for an exit in a circle of unit radius, f of which are faulty. Robots start at the center of the circle and can move anywhere with maximum speed 1.

Our results

•**Presence of Crash Faults**:
Optimal algorithms for the (n, f)-search problem
Optimal worst case completion time $1 + \frac{(f+1)2\pi}{n}$

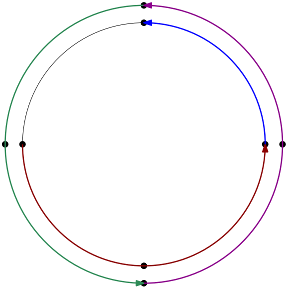
•**Presence of Byzantine Faults**:
Optimal algorithms for $(n, 1)$-search problem
Optimal worst case completion time $1 + \frac{4\pi}{n}$

Crash Faults: Lower Bound

Theorem (Lower Bound for (n, f)-Search)

The worst-case search time $S_c(n, f)$ for $n \geq f + 1$ robots exactly f of which are crash-faulty satisfies

$$S_c(n, f) \geq 1 + (f + 1)\frac{2\pi}{n}$$



Example:
(4,1) - Adversary wins!

We must traverse every point of the circle with at least $(f + 1)$ robots.

If not the adversary will make at most f robots visit the exit and remain silent.

Honest robots will miss the exit

Crash Faults: Lower Bound

Theorem (Lower Bound for (n, f)-Search)

The worst-case search time $S_c(n, f)$ for $n \geq f + 1$ robots exactly f of which are crash-faulty satisfies

$$S_c(n, f) \geq 1 + (f + 1)\frac{2\pi}{n}$$

Corollary (Lower Bound for Byzantine $(n, 1)$-Search)

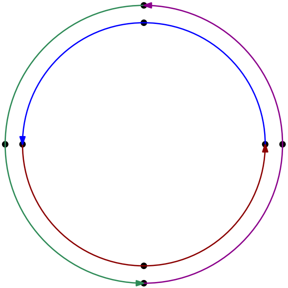
The worst-case search time $S(n)$ for $n \geq 2$ robots exactly one of which is Byzantine-faulty satisfies $S(n) \geq 1 + \frac{4\pi}{n}$.

Crash Faults: Upper Bound

Theorem (Upper Bound for (n, f)-Search with Crash Faults)

The worst-case search time $S_c(n, f)$ for $n \geq 2$ robots exactly f of which are prone to crash failures satisfies

$$S_c(n, f) \leq 1 + (f + 1)\frac{2\pi}{n}$$



Example:
(4,1): Search complete!

- $\theta = \frac{2\pi}{n}$
- a_k moves t $k\theta$ and searches ccw for $(f + 1)\theta$ radians

Crash Faults: Upper Bound

Theorem (Upper Bound for (n, f)-Search with Crash Faults)

The worst-case search time $S_c(n, f)$ for $n \geq 2$ robots exactly f of which are prone to crash failures satisfies

$$S_c(n, f) \leq 1 + (f + 1)\frac{2\pi}{n}$$

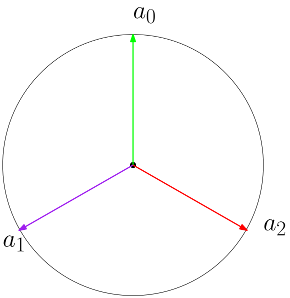
Bounds for crash faults are tight.

Byzantine fault: (3,1)-search

Lemma ((3,1)-Search)

The worst-case search time for 3 robots exactly one of which is Byzantine-faulty satisfies

$$S(3) \leq 1 + \frac{4\pi}{3}$$

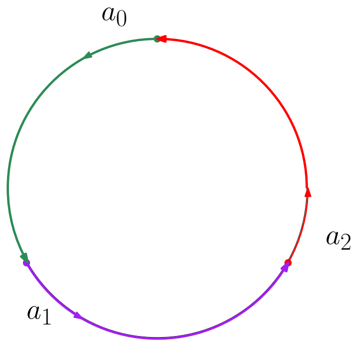


Step 0: time phase [0,1)

- Divide circle in 3 sectors, set $\theta = \frac{2\pi}{3}$
- agent a_k move to location $k\theta$
- in the next phase $[1, 1+\theta)$, agent a_k searches ccw arc $[k\theta, (k+1)\theta]$

Byzantine fault: (3,1)-search

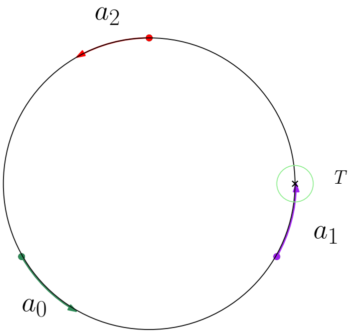
\# announcements = **0**



Step 1: time phase $[1, 1+\theta)$

- exit on territory of faulty robot

\# announcements = **1**

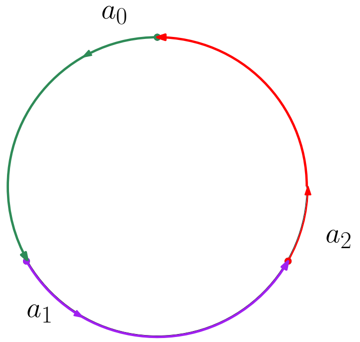


Step 2: time phase $[1+\theta, 1+2\theta)$

- correct announcement

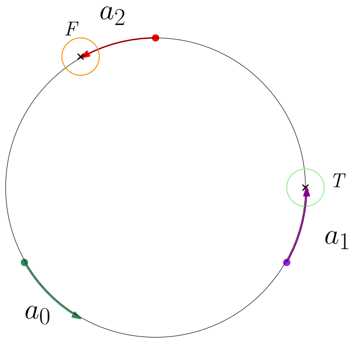
Byzantine fault: (3,1)-search

\# announcements = **0**



Step 1: time phase $[1, 1 + \theta)$

\# announcements = **2**



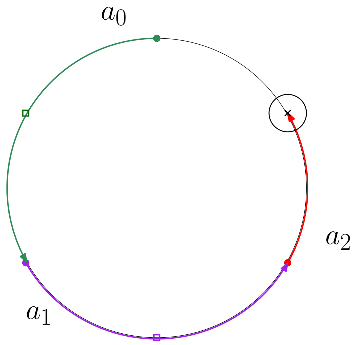
Step 2: time phase $[1 + \theta, 1 + 2\theta)$

- exit on territory of faulty robot

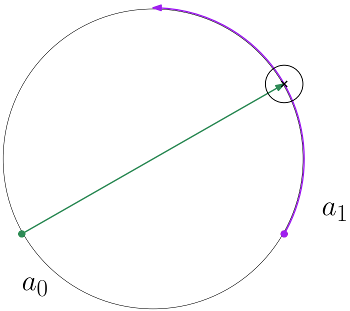
- third agent (say a_0), honest
- a_0 could find the exit in previous phase, if a_2 was honest

Byzantine fault: (3,1)-search

\# announcements = **1**



Step 1: time phase $[1, 1+\theta)$

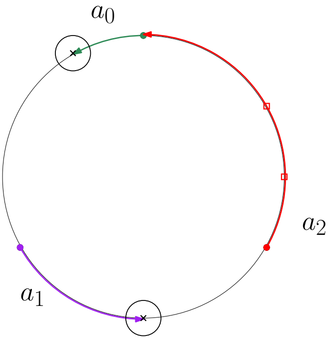


Step 2: time phase $[1+\theta, 1+2\theta)$

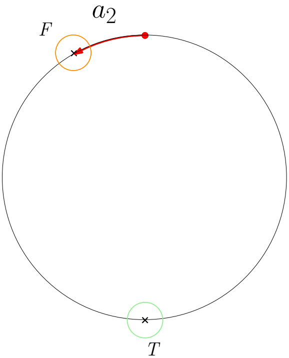
- correct if a_0 or a_1 confirms
- otherwise a_1 will find it
- max time $2 < \theta$

Byzantine fault: (3,1)-search

announcements = **2**



Step 1: time phase $[1, 1 + \theta)$



Step 2: time phase $[1 + \theta, 1 + 2\theta)$

- consecutive sectors
- silent agent is honest

- honest agent will determine the correct exit

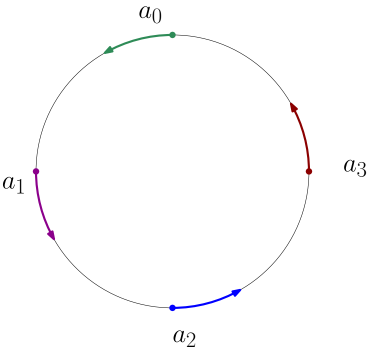
$$\text{max time } 1 + 2\theta = 1 + \frac{4\pi}{n} = 1 + \frac{4\pi}{3}$$

Byzantine fault: (4,1)-search

Lemma ((4,1)-Search)

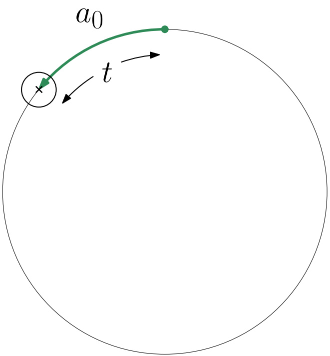
The search time for 4 robots exactly one of which is Byzantine-faulty satisfies

$$S(4) \leq 1 + \pi$$



- set $\theta = \frac{\pi}{2}$
- agent a_k moves to $k\theta$ and continues ccw
- each responsible for arc of length π

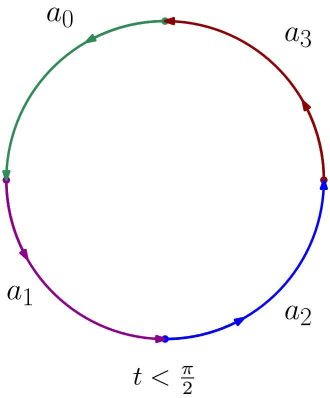
Byzantine fault: (4,1)-search

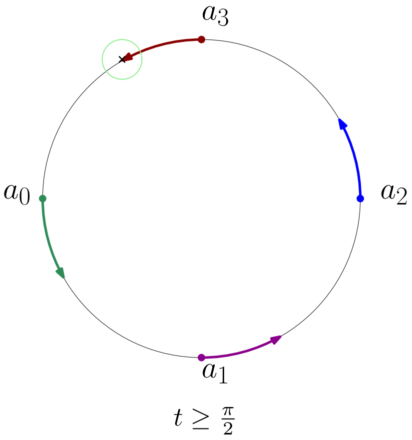


- t: length of arc searched by the agent who made first announcement, at the time of announcement

Byzantine fault: (4,1)-search

Case 1: # announcements in $t \geq \frac{\pi}{2}$: **1**

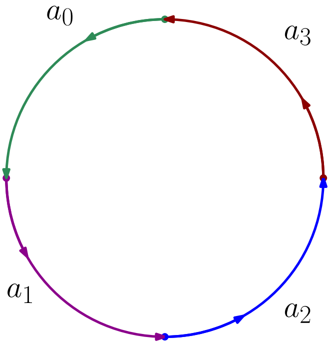




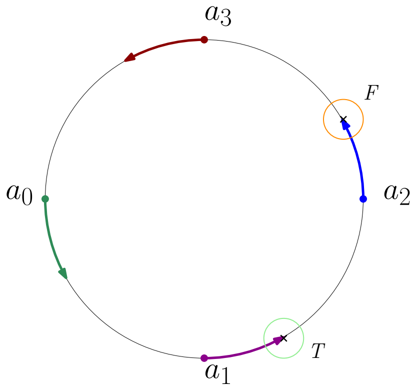
- valid one

Byzantine fault: (4,1)-search

Case 1: # announcements in $t \geq \frac{\pi}{2}$: **2**



$t < \frac{\pi}{2}$



$t \geq \frac{\pi}{2}$

- consecutive sectors
- valid the first in ccw direction

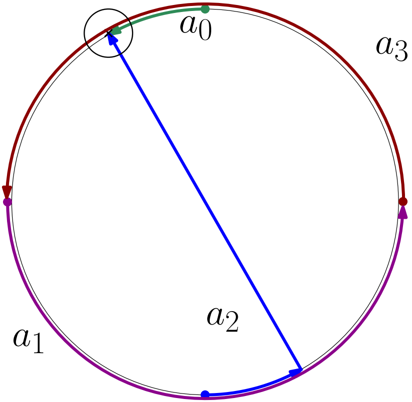
Byzantine fault: (4,1)-search

- In Case 1, where $t \geq \frac{\pi}{2}$, robots will know the correct exit when they check the sectors they are responsible for in time $1 + \pi$

- Otherwise, set $y = \pi - 2$ and suppose an announcement is made by a_0 at $t < \frac{\pi}{2}$

Byzantine fault: (4,1)-search

Case 2: announcements in $t < y$



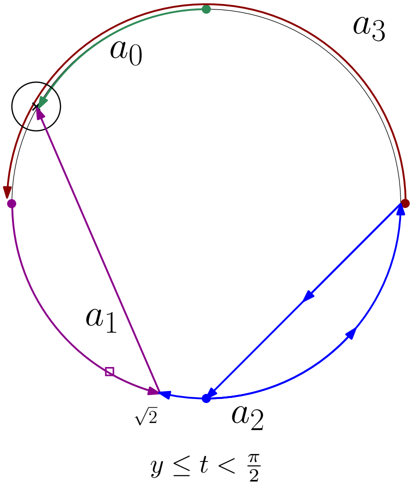
- a_1 and a_3 will search the two sectors that each is responsible for in time

- a_2 will move along a diameter to check the announcement

- y is defined so that a_2 reaches the announcement in time less than $1 + y + 2 = 1 + \pi$

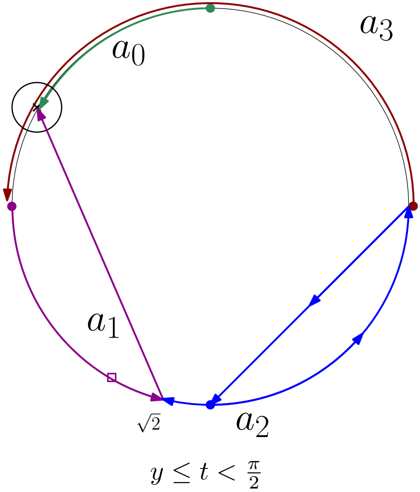
Byzantine fault: (4,1)-search

Case 3: announcements in $y \le t < \frac{\pi}{2}$



- a_1 continues to cover distance $\sqrt{2}$
- then moves along a chord to check announcement
- a_2 finishes first sector and moves back through a chord to check the arc that a_1 left unchecked
- a_3 continues to search his two sectors

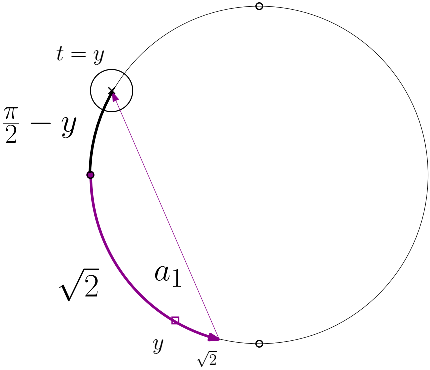
Case 3: announcements in $y \leq t < \frac{\pi}{2}$



- a_2 covered an arc of at most $\frac{\pi}{2} + \sqrt{2} + (\frac{\pi}{2} - \sqrt{2}) = \pi$

Byzantine fault: (4,1)-search

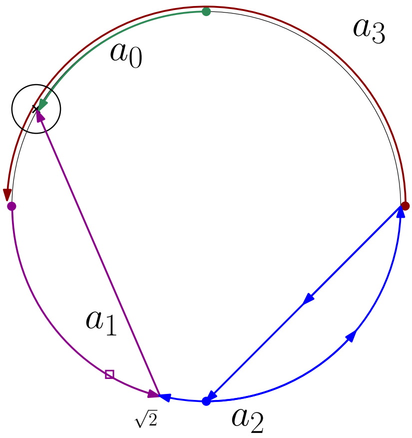
Case 3: announcements in $y \leq t < \frac{\pi}{2}$



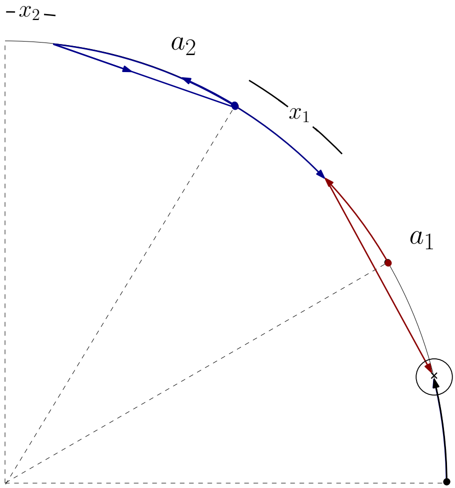
- a_1 in the worst case where $t = y$ will walk a chord that corresponds to an arc of lenght $\phi = \sqrt{2} + \frac{\pi}{2} - y = 2 + \sqrt{2} - \frac{\pi}{2}$

- total time needed is $1 + \sqrt{2} + 2\sin\frac{\phi}{2} < 1 + \pi$

max time $1 + \frac{4\pi}{n} = 1 + \pi$



(4,1)-search: $y \leq t < \frac{\pi}{2}$



(n-1)-search: $y \leq t < \theta$

Byzantine fault: $(n,1)$-search, $n \geq 5$

- Analytical proof for $n \geq 9$
- After some computational verification...

n	x_5	x_4	x_3	x_2	x_1	y	T	$S(n)$
5				0.0810	0.2285	0.611	3.51327	3.51327
6			0.047	0.135	0.3	0.36	3.07	3.09
7		0.029	0.085	0.17*	0.34*	0.2	2.74	2.79
8	0.02	0.04*	0.08*	0.16*	0.32*	0.1	2.56	2.57

Theorem (Upper Bound for (n,1)-Search)

The worst-case search time $S(n)$ for $n \geq 2$ robots exactly one of which is Byzantine-faulty satisfies

$$S(n) \leq 1 + \frac{4\pi}{n}$$

Bounds for *one* Byzantine fault are tight!

Conclusion

Search on a circle with n robots, where

- $f \geq 1$ crash-faulty
  - Optimal worst-case search time is exactly $1 + \frac{(f+1)2\pi}{n}$
- one of them is Byzantine-faulty
  - Optimal worst-case search time is exactly $1 + \frac{4\pi}{n}$

Open problems

- Multiple Byzantine-faulty robots
- Evacuation
- Other topologies
- Other communication models

Thank you!