Optimal Circle Search Despite the Presence of Faulty Robots

Kostantinos Georgiou[1⋆], Evangelos Kranakis[2⋆], Nikos Leonardos[3], Aris Pagourtzis[4], and Ioannis Papaioannou[4]

[1] Department of Mathematics, Ryerson University, Toronto, Ontario, Canada.
konstantinos@ryerson.ca
[2] School of Computer Science, Carleton University, Ottawa, Ontario, Canada.
kranakis@scs.carleton.ca
[3] Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens, Ilissia, Greece.
nikos.leonardos@gmail.com
[4] School of Electrical and Computer Engineering,
National Technical University of Athens, Zografou, Greece.
pagour@cs.ntua.gr, ipapaioannou@corelab.ntua.gr

Abstract. We consider (n, f)-*search on a circle*, a search problem of a hidden exit on a circle of unit radius for $n > 1$ robots, f of which are faulty. All the robots start at the centre of the circle and can move anywhere with maximum speed 1. During the search, robots may communicate wirelessly. All messages transmitted by all robots are tagged with the robots' unique identifiers which cannot be corrupted. The search is considered complete when the exit is found by a non-faulty robot (which must visit its location) and the remaining non-faulty robots know the correct location of the exit.

We study two models of faulty robots. First, crash-faulty robots may stop operating as instructed, and thereafter they remain nonfunctional. Second, Byzantine-faulty robots may transmit untrue messages at any time during the search so as to mislead the non-faulty robots, e.g., lie about the location of the exit.

When there are only crash fault robots, we provide optimal algorithms for the (n, f)-search problem, with optimal worst-case search completion time $1 + \frac{(f+1)2\pi}{n}$. Our main technical contribution pertains to optimal algorithms for $(n, 1)$-search with a Byzantine-faulty robot, minimizing the worst-case search completion time, which equals $1 + \frac{4\pi}{n}$.

Keywords and phrases. Adversary, Byzantine, Circle, Exit, Perimeter, Robot, Search, Speed, Wireless Communication.

1 Introduction

Search is a problem of vital importance because of its numerous critical applications in various branches of mathematics and theoretical computer

⋆ Research supported in part by NSERC Discovery grant.

science. Several linear search models concerning non-communicating agents have been the focus of investigation in numerous research publications, e.g., see the books Ahlswede and Wegener [1], Alpern and Gal [2], and Stone [15].

In this paper we consider searching for an exit placed at an unknown location on the perimeter of a unit radius disk by dimensionless robots (also referred to as mobile agents) that may communicate wirelessly, some of which are either crash-faulty or Byzantine-faulty. Crash-faulty robots may stop operating, in which case they can transmit no information. Byzantine robots are malicious in that they may falsify the information they transmit to peers by misleading them and thus delaying the overall worst-case search time of the system. Our approach differs from traditional models to search with mobile agents whereby the faults were restricted to the underlying search domain (e.g., graph, continuous infinite line, plane, etc.), in that we are interested in completing the search successfully when one of the mobile agents is faulty thus placing an additional strain on the mobile agents. Before giving details of our main results we formally describe the capabilities of the mobile agents and the computation model.

1.1 Computation model

Our overall purpose is to design search algorithms which find the exit and whose running time, as measured by the time it takes the first robot to find the exit and convince the rest of the robots, is worst-case optimal. In this subsection we define the main parameters of the model which include communication, robot movement, crash and Byzantine faults, and the power of the adversary.

Communication. The robots can communicate wirelessly and instantaneously (no delay) at any time and regardless of their distance from each other. A typical message may involve information about their location, how far they have moved from their starting location, whether or not they found the exit, etc. Robots can deduce their current relative location from each other's messages, they are equipped with a pedometer (to measure distances) but otherwise have no need for GPS. Each message is tagged with the robot's unique identifier which cannot be altered by any robot.

Robot movement. Robots start their movement at the centre of a unit radius disk. Their maximum speed is 1, and this is the same for all the robots. During their movement, they may recognize the perimeter of the disk and the exit if they are at its location as well as move along the perimeter. They are also allowed to take "shortcuts" by moving in the interior of the disk.

Fault types. In our algorithms, robots agree on the specific trajectory that they are supposed to traverse, and in particular they know each others' trajectories. Thus, the location of a robot may be deduced by other robots based on the timing of a message transmission (and the type of the message). A crash-faulty robot may at any time stop functioning, meaning that it permanently remains idle and/or fails to communicate any messages, i.e. it crashes. A Byzantine robot is malicious in that it may alter its trajectory and provide (or hide) information whose purpose is to confuse the rest of the robots on the location of the exit. Note that a Byzantine robot may exhibit the behavior of a crash-faulty robot.

Adversary. For the worst case analysis of our algorithms, we consider an adversary who controls the location of the exit and the behaviour of the malicious robot (its trajectory as well as the messages it will broadcast) so as to maximize the resulting search completion time. A search is complete if the exit has been visited by a non-faulty robot and the rest of the agents, if any, can be convinced (provably) of the (correct) location of the exit.

1.2 Related work

There has been extensive literature on line search starting with the seminal papers of Beck and Bellman [4, 5] and Baeza-Yates et.al. [3]. Both cases are concerned with linear search: a single mobile agent searching for an exit placed at an unknown location on an infinite line; in the former case the setting is stochastic and in the latter deterministic. This line of research continued by several authors and culminated with the seminal books by Ahlswede and Wegener [1], Alpern and Gal [2], and Stone [15]. Several other models for line search algorithms were subsequently investigated, e.g., by Kao et. al, [14] for randomized line search and Demaine et. al. [12] for taking into account the turn cost, just to mention a few. An interesting variant to the linear search above has to do with the presence of faulty robots. The two main papers in this line of research are [10] for crash-faulty robots and [9] for Byzantine-faulty robots.

The circle search model (considered in our paper) for n non-faulty robots was introduced as an evacuation problem (completion time with respect to the last finder of the hidden exit) in [6] and analyzed in both the wireless and face-to-face communication models. Since then there have been numerous related research papers mainly on evacuation, e.g. [13] in the face -to-face model, [11] in equilateral triangles, etc. The interested reader could consult the recent survey [8] for additional related literature.

Directly related to our current work is [7]. In this paper, authors investigate *evacuation* of robots in the presence of crash and/or Byzantine faults. Evacuation is different from search in that it measures completion of the algorithm by the time it takes the last non-faulty robot to find the exit (i.e., all the robots have to go to the exit), unlike search as studied in our current paper which measures success by the the time it takes the first non-faulty robot to find the exit so that all non-faulty robots are convinced (provably) that the exit has been found and also know its location. To the best of our knowledge, the circle search model with a Byzantine-faulty robot has not been considered in the past.

1.3 Preliminaries and notation

Assume that n is the number of robots, f of which are faulty. Robots are dimensionless and are initially located on the centre of a unit radius disk. The exit is located on the *unit circle*, which is the circumference of the disk. Robots can move with maximum speed 1. In our algorithms, all honest agents move at the maximum speed, therefore at each time point all agents know the location of every agent that follows the protocol. The n mobile agents are denoted by $a_0, a_1, \ldots, a_{n-1}$ and exactly f of them are faulty. The indices are treated as elements of $\mathbb{Z}_n$; in particular, index addition and subtraction are performed modulo n. Throughout the paper, we call our problem (n, f)-search, meaning search for $n > 1$ robots, f of which are faulty. Robots will be searching the unit circle either clockwise (cw) or counter-clockwise (ccw). It is assumed throughout that whenever an honest agent finds the exit it announces this fact, and whenever it realizes that an announcement of another agent is faulty it also announces this to everybody.

Our main contribution pertains to the $(n, 1)$-search problem with a Byzantine-faulty robot. By $S(n)$ we denote the infimum, over all algorithms, of the time required for the first non-faulty robot to reach the exit so that all robots also know (provably) the correct location of the exit. Similarly, for the (n, f)-search problem with crash-faulty robots, we denote the optimal search completion time by by $S_c(n, f)$.

1.4 Results of the paper

For $n \geq 2$, we give optimal algorithms for problem $(n, 1)$-search. Our main result is that $(n, 1)$-search on a circle admits a solution with search completion time $1 + \frac{4\pi}{n}$ and this is worst-case optimal. In Section 2 we prove a lower bound for f crash-faulty robots, hence for Byzantine robots too. In Section 3 we match the previous lower bound for crash-faulty robots with a tight upper bound. Then in Section 4 we focus on the upper bounds for searching with 1 Byzantine robots. In particular, in Subsection 4.1 we analyze the case of 3 robots, in Subsection 4.2 the case of 4 robots, and in Subsection 4.3 the general case of n robots. In Section 5, we conclude with a relevant discussion and open problems.

2 Lower Bound

In this section we give a lower bound for our search problem. This result builds on the work in [7]; we extend their arguments to the case of f crash-faulty robots (hence, Byzantine too).

Theorem 1 (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*

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

Proof. (Theorem 1) Since the maximum speed of the robots is 1, it takes at least time 1 for a robot to reach the perimeter of the disk. Further, every point on the perimeter must be traversed by at least $f + 1$ robots; for if not, the adversary will make the at most f robots visiting this point all faulty in that they remain silent and therefore the non-faulty robots will miss the exit.

Let ℓ_i be the perimeter lengths explored by exactly i robots, where $0 \leq i \leq n$. It is clear from the above discussion that $\ell_0 = \ell_1 = \cdots \ell_f = 0$ and $\ell_{f+1} + \ell_{f+2} + \cdots + \ell_n = 2\pi$. The sum of the parts of the perimeter explored by the robots is $(f + 1)\ell_{f+1} + (f + 2)\ell_{f+2} + \cdots + n\ell_n$. If the robots accomplish this task by exploring the perimeter for time t (after the perimeter of the disk is reached for the first time), then it must be true that

$$\begin{aligned} nt &\geq (f + 1)\ell_{f+1} + (f + 2)\ell_{f+2} + \cdots + n\ell_n \\ &\geq (f + 1)(\ell_{f+1} + \ell_{f+2} + \cdots + \ell_n) \\ &= (f + 1)2\pi. \end{aligned}$$

It follows that $t \geq (f+1)2\pi/n$. This completes the proof. $\square$

Since $S(n) \geq S_c(n,1)$, we immediately obtain the following corollary.

Corollary 1 (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 $S(n) \geq 1 + \frac{4\pi}{n}$.*

3 Searching with Crash Faults

In this section we match the lower bound of Theorem 1 when we search with crash-faulty robots.

Theorem 2 (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*

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

Proof. Let $\theta := 2\pi/n$. Our algorithm is as follows. For each $k = 0, \ldots, n-1$, agent a_k moves to the point $k\theta$ of the unit circle and searches ccw for $(f+1)\theta$ radians. When (and if) exit is found, it is reported instantaneously.

Clearly, every sector S_j of the circle would be visited by $f+1$ robots if they all followed the protocol. Since there are at most f faulty robots, there must be at least one honest robot that will visit S_j and announce the correct location. As there can only be crash failures there will not be any contradicting announcements. $\square$

4 Search with one Byzantine Fault

In this section we analyze upper bounds for our search problem with a Byzantine agent. Our main theorem is the following.

Theorem 3 (Upper Bound for (n,1)-Search). *The worst-case search time $S(n)$ for $n \geq 2$ robots exactly one of which is faulty satisfies*

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

Thus, combining Corollary 1 with Theorems 3, we conclude that the worst-case search completion time for $(n,1)$-search satisfies $S(n) = 1 + \frac{4\pi}{n}$.

First observe that it is trivial to prove $S(2) = 1 + 2\pi$, for $(2, 1)$-search since one of the two robots is faulty and the other non-faulty, hence the non-faulty has no other option but to search the entire perimeter.

In the next two Subsections (4.1 and 4.2) we show the upper bound for the cases $(3, 1)$-search and $(4, 1)$-search. Although the algorithms for these cases can be seen as special cases of the algorithm for the general case (Subsection 4.3), this is not the case for their analysis. In addition, presenting them separately allows to better clarify and illustrate the techniques and notions that we employ.

4.1 $(3, 1)$-search with a Byzantine-faulty robot

Lemma 1 ((3,1)-Search). *The worst-case search time for* 3 *robots exactly one of which is faulty satisfies*

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

Proof. We will prove the claim by presenting an algorithm for this case. Consider agents a_0, a_1, a_2 and set $\theta = 2\pi/3$. We describe below the agents' actions in phases (time intervals) $[0, 1)$, $[1, 1 + \theta)$ and $[1 + \theta, 1 + 2\theta)$ and we explain why all agents know the location of the exit at time $1 + 2\theta$.

Phase $[0, 1)$: Each agent a_k, $k \in \{0, 1, 2\}$, moves along a radius to the point $k\theta$ of the unit circle.

Phase $[1, 1 + \theta)$: Agent a_k searches ccw the arc $[k\theta, (k + 1)\theta)$.

Phase $[1 + \theta, 1 + 2\theta)$:

(i) If no announcements were made in time interval $[1, 1 + \theta)$ then in time interval $[1+\theta, 1+2\theta)$ either there will be one correct announcement or two announcements. In the latter case the third agent, say a_k, is honest and the correct announcement is the one by a_{k+1} (otherwise, a_k would have seen in time interval $[1, 1 + \theta)$ the exit announced by a_{k-1}).

(ii) If exactly one announcement was made in time interval $[1, 1 + \theta)$, say by agent a_{k-1}, then agent a_k moves directly (along a chord) to the location of the announcement and a_{k+1} searches ccw for another θ radians. This takes time at most $2 < \frac{2\pi}{3}$. If a_k or a_{k+1} confirms the announcement then it is correct; otherwise, a_{k+1} in this time interval announces the correct exit point. This case is depicted in Figure 1.[5]

[5] Figures in this paper depict robot trajectories during the execution of our search algorithm. They restrict to cases where the first announcement is made while robots search their first sector of length $\theta = \frac{2\pi}{n}$, and no other announcement is made until

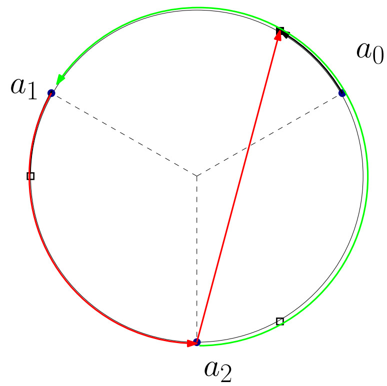

Fig. 1. $(3,1)$-search: robot trajectories in case $t < \frac{2\pi}{3}$.

(iii) If two announcements were made in time interval $[1, 1+\theta)$, then they are in consecutive sectors. The silent agent is certainly non-faulty and will visit one of these sectors in this phase and will thus be able to determine which announcement was the correct one.

This completes the description of the algorithm and the proof. □

4.2 $(4,1)$-search with a Byzantine-faulty robot

We will first describe an algorithm for this case. Let $\theta = \pi/2$. Each agent a_k moves with speed one to its *starting point* $k\theta$ and then continues ccw. We call the arc from one starting point to the next a *sector*. We think of each agent being responsible for the arc of length π that begins at its starting point and covers at most two consecutive sectors ccw.

Let t denote the length of the arc from the point of the first announcement to the starting point that corresponds to the agent that made the announcement (note, there is always an announcement for some $t \leq \pi$). If $t \geq \frac{\pi}{2}$, then each robot checks both sectors that are assigned to it. Otherwise, set $y = \pi - 2$ and suppose an announcement is made by a_0 (w.l.o.g.) at $t < \frac{\pi}{2}$. We consider two cases.

time $1+\theta$. It is assumed that agent a_0 makes the first announcement. A black square shows the location of the announcement; a white square shows the locations of other agents at that time. A solid dot shows the starting positions of the robots on the unit circle (starting from the center of the circle, they move directly, in time 1, to their starting positions). Recall that the arc length between the starting position of a_0 and the point of the announcement is denoted by t (hence, the announcement takes place in time $1+t$).

If $t < y$, then a_1 and a_3 will search the two sectors that each is responsible for and a_2 will move along the diameter to check the announcement. This case is depicted in Figure 2 below.

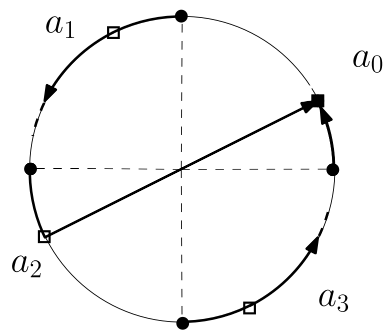

Fig. 2. $(4, 1)$-search: robot trajectories in case $t < y$.

If $y \leq t < \frac{\pi}{2}$, then a_1 continues to cover distance $\sqrt{2}$ (unless $t \geq \sqrt{2}$) and then moves along a chord to check the announcement; a_2 finishes its first sector and then moves back along a chord to its starting point and continues cw to check the arc that a_1 didn't check; a_3 continues searching its two sectors. This case is depicted in Figure 3 below.

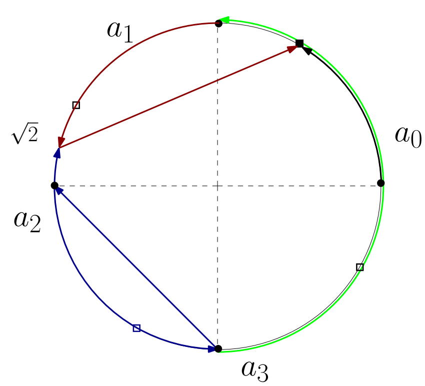

Fig. 3. $(4, 1)$-search: robot trajectories in case $y \leq t < \frac{\pi}{2}$.

This completes the description of the algorithm. We will now prove the correctness and the upper bound on the execution time.

Lemma 2 ((4,1)-Search). *The search time for* 4 *robots exactly one of which is faulty satisfies*

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

Proof. Recall that we denote by t the length of the arc searched on the circle by the agent who made the first announcement, at the time of the announcement.

For $t \geq \frac{\pi}{2}$ we argue that when every robot has checked the sectors it is responsible for (at time $1+\pi$), all of them know the location of the exit. First, note that if only one announcement is made, then it has to be a valid one. Therefore, assume two announcements are made (note that both are no earlier than $\frac{\pi}{2}$). Observe that they have to come from consecutive sectors: the exit must be at the first sector of the faulty robot, say a_3 since nobody spoke earlier than $\frac{\pi}{2}$, and it is discovered by a_2, while searching its second sector, who makes a correct announcement. The only other announcement can be made by a_3 and is faulty. Therefore, all agents know that the location is at the first of the two sectors in the ccw direction.

For $t < \frac{\pi}{2}$ suppose the first announcement was made by a_0. We claim that in this case the first announcement is checked by two more agents (namely, by a_3 and either a_1 or a_2) and every point of the perimeter is searched by one of the three other agents (unless a second announcement is made in which case it is not necessary to search the whole circle as one of the two must be correct). Assuming this claim, if the first announcement is verified by any other agent, then clearly it is valid. If not, then two agents reject it, thus it must be fake. It follows that another announcement was made which has to be valid. We next verify the claim and the execution time.

Consider the case $t < y$. Note that y was defined so that a_2 reaches the announcement in time less than $1+y+2 = 1+\pi$. Thus, the announcement is checked by a_2 and a_3 in time, while a_1 and a_3 search every point of the perimeter.

Consider now $y \leq t < \frac{\pi}{2}$. First, to see that every sector was searched by the first three agents by time $1+\pi$, we need to argue that a_1 and a_2 covered the first sector. Indeed, a_2 searched an arc of length $\frac{\pi}{2}$ to finish his first sector, a chord of length $\sqrt{2}$ to go back to his starting point, and an arc of length at most $\frac{\pi}{2} - \sqrt{2}$ that was left uncovered by a_1; this sums up to at most $\frac{\pi}{2} + \sqrt{2} + \frac{\pi}{2} - \sqrt{2} = \pi$ as desired. Next, we need to argue that the announcement location was reached by a_1 in time $1+\pi$. This is clear if $t \geq \sqrt{2}$. Otherwise, it is not hard to see that the worst case is $t = y$. In this case, the chord a_1 walks corresponds to an arc of length $\phi = \sqrt{2} + \frac{\pi}{2} - y = 2 + \sqrt{2} - \frac{\pi}{2}$. Thus, the total time it needs is $1 + \sqrt{2} + 2\sin\frac{\phi}{2} < 1 + \pi$. $\square$

4.3 $(n, 1)$-search with a Byzantine-faulty robot, $n \geq 5$

We will first give the description of the algorithm for this case. For each $k \in \mathbb{Z}_n$, agent a_k moves to the *k-th starting point* P_k located at $k\theta$, $\theta = 2\pi/n$, and then continues ccw. We denote the arc of size θ from the k-th starting point to the next by S_k and call it the k-th sector. We think of sectors S_k and S_{k+1} as being *assigned* to agent a_k, who is supposed to search them in the ccw direction.

Let t denote the length of the arc from the point of the first announcement to the starting point that corresponds to the agent that made the announcement. We now describe the trajectories of agents for the case that agent a_0 makes the first announcement. We will argue later (in the proof of Theorem 3) that the information they exchange is enough for all agents to learn the exit location.

If $t \geq \theta$, then each agent checks both sectors that are assigned to it. Otherwise, set

$$y = 2\theta - 2\sin\theta$$

and suppose an announcement is made by a_0 at $t < \theta$. Consider two cases.

If $t < y$, then each agent a_k with $k \notin \{0, 2\}$ will search its two sectors, while a_2 will start at time $1 + t$ to move along a chord towards the announcement in order to verify it.

If $y \leq t < \theta$, define arc-lengths x_k (in S_k but not to be searched by a_k) recursively as follows.

$$x_{n-2} = 0; \quad x_k = \theta + x_{k+1} - 2\sin\Big(\frac{\theta - x_{k+1}}{2}\Big), \text{ for } 0 < k < n - 1. \quad (1)$$

Agent a_1 continues to cover distance $\theta - x_1$ (unless $t \geq \theta - x_1$) and then moves along a chord towards the announcement in order to verify it; for $1 < k < n - 1$, agent a_k continues to cover distance $\theta - x_k$ (unless $t \geq \theta - x_k$), then moves along a chord back to its starting point, and finally searches in the cw direction the arc (of length at most x_{k-1}) that agent a_{k-1} didn't search; agent a_{n-1} continues with its two sectors. This case is depicted in Figure 4 below.

This completes the description of the algorithm. We next show its correctness and the upper bound on its running time.

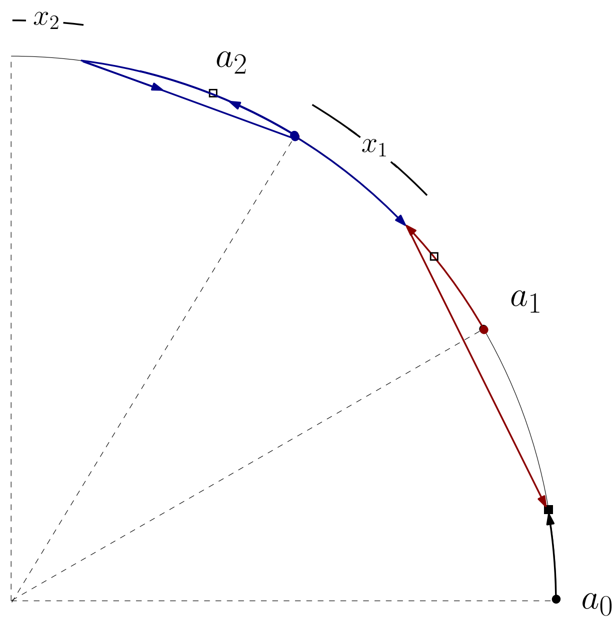

Fig. 4. $(n, 1)$-search: robot trajectories in case $y \le t < \theta$.

Lemma 3 ($(n, 1)$-Search, for $n \ge 5$). *The worst-case search time for $n \ge 5$ robots exactly one of which is faulty satisfies*

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

Proof. (Lemma 3) We are going to argue about the correctness and the execution time of the algorithm described above.

If $t \ge \theta$, then all agents have searched the sectors assigned to them by time $1 + 2\theta$. We need to show that all of them know the location of the exit. First, note that if only one announcement is made, then it has to be a valid one. Thus, assume two announcements are made. Observe that they have to come from consecutive sectors: one of them is the true one and was discovered by an honest agent, say a_k, while searching sector S_{k+1}. It follows that a_{k+1} is faulty (because it didn't make the announcement) and the other announcement must come from it. Therefore, the agents know that the location is at the first announcement encountered in the ccw direction.

Otherwise ($t < \theta$), suppose the first announcement was made by a_0. We claim the following.

The first announcement is checked by two more agents and every point of the perimeter is searched by at least one agent different from a_0, unless a second announcement is made.

Note first that if the first announcement is verified by one more agent, then it is proved valid to all. If not, then—assuming the claim—two agents reject it and a_0 is proved faulty to all. Furthermore, every point of the perimeter will be searched by at least one honest agent. It follows—by the second part of the claim—that another announcement will be made and will be recognized by all as valid. We next verify the claim and the execution time for the two cases on t.

Consider the case $t < y$. Note that y was defined so that a_2 reaches the announcement in time less than $1+y+2\sin\theta = 1+2\theta$. This is because it will spend less than time y on its first sector and then move along the chord that corresponds to two sectors. Thus, the announcement is checked by a_2 and a_{n-1} in time, while the other agents set forth to search every point of the perimeter.

Consider now $y \leq t < \theta$. First, we verify that every sector was searched by one of the agents $a_1, \ldots, a_{n-1}$ by time $1 + 2\theta$. It is clear that a_{n-1} searched sectors S_{n-1} and S_0. Next, we argue that, for $0 < k < n-1$, agents a_k and a_{k+1} covered sector S_k. Note that x_k is the length of S_k that was not searched by agent a_k. However, x_k is defined so that a_{k+1} has sufficient time to travel back to P_{k+1} and aid a_k. Indeed, the worst case for a_{k+1} is when $t \leq \theta - x_k$. (It is not hard to see that when $t > \theta - x_k$ he will have time to spare.) In this case, after reaching point $\theta - x_{k+1}$ of S_{k+1}, it must search a chord corresponding to an arc of $\theta - x_{k+1}$ radians and an arc of length x_k. Since it has $\theta + x_{k+1}$ time left, the definition of x_k is such that he can manage its task. Finally, we need to argue that the announcement was reached by a_1 in time $1+2\theta$. This is clear if $t \geq \theta - x_1$. Otherwise, it is not hard to see that the worst case is $t = y$. In this case, the chord a_1 searches corresponds to an arc of length $2\theta - x_1 - y$. Thus, the total time a_1 needs is

$$T = 1 + (\theta - x_1) + 2\sin\Big(\frac{2\theta - x_1 - y}{2}\Big).$$

In the sequel we will make use of the following simple facts.

Fact 1 *For* $x \in (0, \frac{\pi}{2})$, $\sin x < x$.

Fact 2 *For* $x \in (0, \frac{\pi}{2})$, $\sin x < 2\sin\frac{x}{2}$.

Fact 3 *For* $x \in (0, \frac{\pi}{4})$, $\sin x < x - \frac{x^3}{7}$.

Since, for $n \geq 4$, $2\theta - x_1 - y < \pi$, using Fact 1 (twice) and substituting $y = 2\theta - 2\sin\theta$ we obtain

$$T \leq 1 + (\theta - x_1) + (2\theta - x_1 - y) \leq 1 + 2\theta - 2x_1 + \sin\theta.$$

To provide a lower on x_1, apply Fact 1 on the recursive definition to obtain

$$x_{n-3} = \theta - 2\sin\frac{\theta}{2}; \quad x_k \geq 2x_{k+1}, \text{ for } 0 < k < n-1. \tag{2}$$

It follows that

$$x_1 \geq 2^{n-4}\Big(\theta - 2\sin\frac{\theta}{2}\Big).$$

Combining with the upper bound on T, to show $T \leq 1 + 2\theta$, it suffices to argue that

$$2^{n-3}\Big(\frac{2\pi}{n} - 2\sin\frac{\pi}{n}\Big) \geq \sin\frac{2\pi}{n}.$$

Using Fact 2, $\sin\frac{2\pi}{n} \leq 2\sin\frac{\pi}{n}$. Substituting this and rearranging, it suffices to show that

$$2^{n-3}\cdot\frac{\pi}{n} \geq \big(2^{n-3}+1\big)\sin\frac{\pi}{n}.$$

In view of Fact 3, the sufficient condition simplifies further to

$$2^{n-3} \geq \big(2^{n-3}+1\big)\Big(1 - \frac{\pi^2}{7n^2}\Big) \iff \big(2^{n-3}+1\big)\pi^2 \geq 7n^2,$$

which holds for all $n \geq 9$.

Finally cases $n \in \{5,6,7,8\}$ have been verified computationally as follows. In the table below we list values $y, x_1, \ldots, x_{n-3}$ for $n \in \{5,6,7,8\}$. These values determine the algorithm for these cases. To verify the table, it suffices to verify $y \leq 2\theta - 2\sin\theta$, $T \geq 1 + (\theta - x_1) + 2\sin(\frac{2\theta - x_1 - y}{2})$, $S(n) \leq 1 + 2\theta$, and $x_k \leq \theta + x_{k+1} - 2\sin(\frac{\theta - x_{k+1}}{2})$ (for $0 < k < n-2$). With respect to the x_k values, note that those which are double the previous one (marked with an asterisk) need not be verified in view of inequality (2).

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

This completes the proof of the lemma. □

Now we can complete the rest of the proof of Theorem 3.

Proof. (Theorem 3) Lemmas 1 and 2 prove the upper bound for $n = 3, 4$ robots respectively, and cases $n \geq 5$ are covered by Lemma 3. □

5 Conclusion

In this paper we considered search on a circle with n robots, where either $f \geq 1$ of them are crash-faulty, or one of them is Byzantine-faulty, and we proved that the optimal worst-case search times are exactly $1 + \frac{(f+1)2\pi}{n}$ and $1 + \frac{4\pi}{n}$, respectively. The optimality for the Byzantine case is quite surprising given that there are very few tight bounds for search on a circle even for the wireless model. Extending the results either to multiple Byzantine-faulty robots or to the evacuation problem are two challenging open problems in the context of circle search.

References

1. R. Ahlswede and I. Wegener. *Search problems.* Wiley-Interscience, 1987.
2. S. Alpern and S. Gal. *The theory of search games and rendezvous*, volume 55. Springer, 2003.
3. R. Baeza-Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Inf. Comput.*, 106(2):234–252, October 1993.
4. A. Beck. On the linear search problem. *Israel Journal of Mathematics*, 2(4):221–228, 1964.
5. R. Bellman. An optimal search. *Siam Review*, 5(3):274–274, 1963.
6. J. Czyzowicz, L. Gasieniec, T. Gorry, E. Kranakis, R. Martin, and D. Pajak. Evacuating robots from an unknown exit located on the perimeter of a disc. In *DISC 2014*. Springer, Austin, Texas, 2014.
7. J. Czyzowicz, K. Georgiou, M. Godon, E. Kranakis, D. Krizanc, W. Rytter, and M. Włodarczyk. Evacuation from a disc in the presence of a faulty robot. In *International Colloquium on Structural Information and Communication Complexity*, pages 158–173. Springer, 2017.
8. J. Czyzowicz, K. Georgiou, and E. Kranakis. Group search and evacuation. In P. Flocchini, G. Prencipe, and N. Santoro, editors, *Distributed Computing by Mobile Entities; Current Research in Moving and Computing*, chapter 14, pages 335–370. Springer, 2019.
9. J. Czyzowicz, K. Georgiou, E. Kranakis, D. Krizanc, L. Narayanan, J. Opatrny, and S. Shende. Search on a line by byzantine robots. In *ISAAC*, pages 27:1–27:12, 2016.
10. J. Czyzowicz, E. Kranakis, D. Krizanc, L. Narayanan, and Opatrny J. Search on a line with faulty robots. In *PODC*, pages 405–414. ACM, 2016.

11. J. Czyzowicz, E. Kranakis, K. Krizanc, L. Narayanan, J. Opatrny, and S. Shende. Wireless autonomous robot evacuation from equilateral triangles and squares. In *ADHOCNOW*, pages 181–194. Springer, 2015.
12. E. D. Demaine, S. P. Fekete, and S. Gal. Online searching with turn cost. *Theoretical Computer Science*, 361(2):342–355, 2006.
13. Czyzowicz J., Georgiou K., Kranakis E., Narayanan L., Opatrny J., and Vogtenhuber B. Evacuating using face-to-face communication. *CoRR*, abs/1501.04985, 2015.
14. M.-Y. Kao, J. H. Reif, and S. R. Tate. Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. *Information and Computation*, 131(1):63–79, 1996.
15. L. Stone. *Theory of optimal search.* Academic Press New York, 1975.