Evacuation from a Disc in the Presence of a Faulty Robot

Jurek Czyzowicz[167], Konstantinos Georgiou[26], Maxime Godon[1], Evangelos Kranakis[36], Danny Krizanc[4], Wojciech Rytter[57], and Michał Włodarczyk[58]

[1] Université du Québec en Outaouais, Gatineau, Canada
[2] Ryerson University, Department of Mathematics, Toronto, ON, Canada
[3] School of Computer Science, Carleton University, Ottawa, Canada
[4] Department of Mathematics, Wesleyan University, Middletown, CT, USA
[5] Institute of Informatics, University of Warsaw, Warsaw, Poland
[6] Research supported in part by NSERC Discovery grant.
[7] Research supported in part by grant NCN2014/13/B/ST6/00770 of the Polish Science Centre.
[8] The author was supported by the National Science Centre of Poland Grant UMO-2016/21/N/ST6/00968.

Abstract. We consider the evacuation problem on a circle for three robots, at most one of which is faulty. The three robots search for an exit placed at an unknown location on the perimeter of the circle. During the search, robots can communicate wirelessly at any distance. The goal is to minimize the time that the latest non-faulty robot reaches the exit.
Our main contributions are two intuitive evacuation protocols for the non-faulty robots to reach the exit in two well-studied fault-models, Crash and Byzantine. Moreover, we complement our positive results by lower bounds in both models. A summary of our results reads as follows:

- *Case of Crash Faults:* Lower Bound ≈ 5.082; Upper Bound ≈ 6.309,
- *Case of Byzantine Faults:* Lower Bound ≈ 5.948; Upper Bound ≈ 6.921,

For comparison, it is known (see [11]) that in the case of three *non-faulty* robots with wireless communication we have a lower bound of 4.159, and an upper bound.of 4.219 for evacuation, while for two non-faulty robots $1 + 2\pi/3 + \sqrt{3} \approx 4.779$ is a tight upper and lower bound for evacuation.

Key words and phrases. Algorithm, Byzantine Faulty, Crash Faulty, Evacuation, Robot, Search.

1 Introduction

Searching an environment to find an exit (or target) placed at an unknown location has been studied extensively in computer science and robotics. The searchers are autonomous robots which (may) cooperate during their search by exchanging messages so that at least one of them can find the target in minimum possible time. Another form of search recently introduced in [11] is called *evacuation* and it has the additional requirement that all the robots must go to the exit. Thus, optimality in evacuation is measured by the time it takes for the last robot to reach the exit, whereas in traditional search, optimality is measured by the time it takes the first robot to reach the exit.

In this paper we consider an evacuation problem for three robots which are able to communicate wirelessly. Initially, the robots are located at the center of a disc of radius one and must find an exit located on the circumference of the disc and then gather at the location of the exit. We consider two scenarios in which exactly one robot is faulty. In the first scenario, one robot can experience crash faults, which prevents it from either communicating or locating the exit. In the second scenario, one robot can experience Byzantine faults, which allow it to lie, e.g., to claim to have found an exit–where there is none– or even to fail to report (communicate) the location of the exit to the other robots. Note that the evacuation problem is considered to be solved when both non-faulty robots find the exit. For both scenarios, we provide upper and lower bounds.

1.1 Preliminaries/The Model

There are three robots initially located at the center of a unit disc. The robots can move with maximum speed 1 (thus, they may stop or change direction at no cost), and are required to find an exit (whose location is unknown to the robots) located somewhere on the circumference of the disc and then gather at this location as fast as possible. On the perimeter of the disc the robots have a sense of direction and can distinguish between clockwise and counterclockwise direcion of movement. A robot can find the exit only when it is in the same location as the exit. During their search the robots employ a *wireless communication model*, which means that they can exchange information instantaneously and at no cost and at any time, no matter the distance that separates them during their search.

The search problem to be studied is concerned with all non-faulty robots evacuating from the (unknown) exit. The search task is complicated by the fact that one of the three robots, chosen by an adversary, experiences faults, chosen by the adversary as well. We consider two scenarios. In the first scenario, the faulty robot experiences *crash* faults while in the second the robot experiences *Byzantine* faults. In both cases, the goal is to minimize the time till the last non-faulty robot reaches the exit.

- Crash-Evacuation: A *crash* fault can be thought of as a passive fault rending: a robot is either unable or incapable to either detect or report the exit when it reaches it. Thus, such a robot is not expected to find the exit, only non-faulty robots can. However, we assume that in other aspects, a faulty robot moves like a non-faulty robot, and thus non-faulty robots cannot detect which robots are faulty.
- Byzantine-Evacuation: A *Byzantine* faulty robot not only can fail to detect or report the target even after reaching it, it can also make malicious claims about having found the target when in fact it has not. Given the presence of such a faulty robot, the search for the target can only be concluded when the two non-faulty robots have sufficient verification that the target has been found.

All the messages being transmitted by the robots are tagged with the robot's unique identifier, which cannot be altered.

1.2 Related work

Searching an environment to find an exit placed at an unknown location is a well studied problem in computer science and robotics. The searchers are autonomous mobile robots that may also possess partial knowledge of their environment. Many researchers, starting with the seminal work of Bellman [5] and Beck [4], have studied the optimal (length) trajectory traced by a single robot when searching for a target placed at an unknown location on a line. The aim of the algorithmic designer is to minimize the competitive ratio, that is, the supremum, over all possible target locations, of the ratio between the distance traveled by the robot until it finds the exit, and the distance of the exit from the robot's starting position. For the case of a single robot on a line, the optimal trajectory uses a zig-zag, doubling strategy according to which if the robot fails to find the exit after travelling a certain distance in a particular direction it returns to its starting position and doubles its searching distance in the opposite direction. This trajectory has a competitive ratio of 9 and this can be shown to be optimal (e.g., see Baeza-Yates et al. [3]).

Several authors considered the problem of searching in the two-dimensional plane by one or more searchers, including [2, 3]. The evacuation problem on a unit disc for multiple robots considered in our present work is a form of two-dimensional search that was first considered in [11]. In that paper the authors studied evacuation algorithms in the wireless and face-to-face communication models. New algorithms for the face-to-face communication model were subsequently analyzed for two robots in [14] and later in [7]. The problem has also been considered in other domains, like triangles and squares in [16]. However, all these papers concern evacuation only for non-faulty robots.

One of the novelties of our current work is that we consider the two-dimensional evacuation problem with fault tolerance. There are numerous studies of fault tolerance in distributed computing, (see, e.g., [19, 22, 23]). Network failures were most frequently related to static elements of the networked environment (i.e., nodes and links) as opposed to its mobile components. Malfunctions of this kind were sometimes modelled by dynamic alteration of the network [8, 21]. Distributed computation arising when having some of the mobile robots are faulty were investigated in the context of the problems of gathering [1, 17, 18, 24], convergence [6, 9], flocking [25], and patrolling [12]. Several researchers also investigated the case of unreliable or inaccurate robot sensing devices, e.g., [10, 20, 24]. Related to our study is also the research in [12], where a collection of robots, some of which are unreliable, perform efficient patrolling of a fence. Most relevant to our current study for its perspective on search and fault tolerance is the research of [15] and [13] which propose search algorithms for faulty robots that may suffer from crash and Byzantine faults, respectively.

1.3 Outline and results of the paper

An outline of this paper can be described as follows. Section 2 is dedicated to upper bounds. In Sections 2.1 and 2.2 we provide evacuation protocols along with their (worst case) analyses for the Crash-Evacuation problem and the Byzantine-Evacuation problem, respectively. Then, in Section 3 we give lower bounds for both problems. Section 4 gives a discussion of possibilities for further research. The main results of the paper are summarized in Table 1. Notably, since the

Problem	Lower Bound	Upper Bound
Crash-Evacuation	≈ 5.082 (Theorem 3)	≈ 6.309 (Theorem 1)
Byzantine-Evacuation	≈ 5.948 (Theorem 3)	≈ 6.921 (Theorem 2)

Table 1. Comparison of Crash vs Byzantine: the first column gives the type of fault, the middle column lower bounds, and the right column upper bounds for the corresponding type of faults.

optimal offline algorithm for both problems Crash-Evacuation and Byzantine-Evacuation would have the robots move directly to the exit at time 1, the time bounds of Table 1 can be also understood as bounds for the competitive ratio of the underlying online problems.

It is interesting to compare the results obtained in our paper to the case of non-faulty robots. It is known (see [11]) that in the case of three *non-faulty* robots with wireless communication we have a lower bound of 4.159, and an upper bound.of 4.219 for evacuation, while for two non-faulty robots $1 + 2\pi/3 + \sqrt{3} \approx 4.779$ is a tight upper and lower bound for evacuation.

2 Evacuation Protocols

In this section we propose evacuation algorithms for crash and Byzantine faults, respectively.

2.1 Evacuating with Crash-Faults

The main contribution is as follows.

Theorem 1. Crash-Evacuation *can be solved in time* ≈ 6.309.

We prove Theorem 1 by identifying the *best* among a special family of natural algorithms that we call *persistent*. These are algorithms that force all robots to immediately go to the circumference of the disc, and only allow a robot to stop exploring its segment of the disc (either by changing direction, by becoming idle or by leaving the circumference entirely) when it receives information about the exit. Since in this model, a faulty robot can only stay silent, any report about the exit has to be valid. As such, once the location of the exit is received by a robot, the robot moves along the shortest chord toward the reported exit, and evacuates.

We further classify persistent algorithms in two categories: the *symmetric-persistent* that have all the robots begin their exploration in the same direction (either all clockwise or all counter-clockwise), and the *asymmetric-persistent* that have one robot go in a direction, and the other two robots go in the opposite direction. It turns out that the best *asymmetric-persistent* algorithm outperforms the best *symmetric-persistent* algorithm (and also proves Theorem 1). Nevertheless, and as a warm-up, we begin by providing a tight analysis for the family of *symmetric-persistent* algorithms.

Lemma 1. *The best* symmetric-persistent *algorithms deploys the three robots at equidistant points on the disk (at arc-distance $4\pi/3$), and its performance is* $1 + \frac{4\pi}{3} + \sqrt{3}$.

Proof. (Lemma 1) Consider a symmetric-persistent algorithm that deploys robots r_1, r_2, r_3 so that their pairwise anti-clock-wise distance is β, γ and α respectively, as also depicted in Figure 1 (where also arcs A, B, C are defined). Without loss of generality, assume the robots move in clockwise direction.

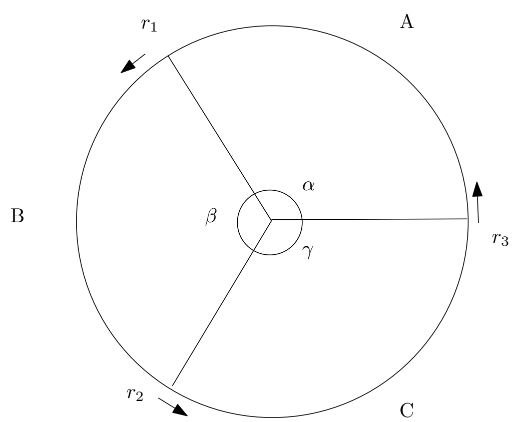

Fig. 1. All robots move counter-clockwise. Arc A includes r_3 and excludes r_1; arc B includes r_1 and excludes r_2; and arc C includes r_2 and excludes r_3.

Consider the case where r_1 is faulty and the robots traverse the arcs depicted in Figure 1. Clearly, there are two cases to consider depending on whether the exit is located in one of the arcs A or B, or the exit is located on arc C. If the exit is located in one of the arcs A or B, then r_3 will discover it. If the exit is located in C, then r_2 will discover it. We say that the exit is either located at a counter-clockwise arc distance of $0 \leq x < \gamma$ from r_2 if r_2 discovers the exit, or a counter-clockwise arc distance of $0 \leq y < \alpha + \beta$ from r_3 if r_3 discovers the exit. Therefore, the total amount of time required to find the exit is given by the formula

$$1 + \max \left\{ \sup_{0 \leq x < \gamma} \left(x + 2 \sin \frac{\gamma}{2} \right), \sup_{0 \leq y < \alpha + \beta} \left(y + 2 \sin \frac{\alpha + \beta}{2} \right) \right\} = 1 + \max \left\{ f(\gamma), f(\alpha + \beta) \right\},$$

where we define $f(x) := x + 2\sin\frac{x}{2}$.

Similarly, if r_2 or r_3 is faulty, then the algorithm terminates in time $1 + \max\{f(\gamma), f(\beta+\gamma)\}$ and $1+\max\{f(\beta), f(\alpha+\gamma)\}$ respectively. We conclude that the best symmetric-adaptive algorithm would choose α, β, γ (partitioning the perimeter of the circle, of length 2π) so as to minimize quantity

$$1 + \max\{f(\alpha), f(\beta), f(\gamma), f(\alpha+\beta), f(\beta+\gamma), f(\alpha+\gamma),\} \tag{1}$$

By choosing $\alpha = \beta = \gamma = \frac{4\pi}{3}$, expression (1) gives completion time $1 + \frac{4\pi}{3} + \sqrt{3}$ as promised.

Finally, we argue that no values of α, β and γ respecting α, β and $\gamma \geq 0$ and $\alpha+\beta+\gamma = 2\pi$ can improve on this bound. Say, we set $\alpha > \frac{2\pi}{3}$. Then it is clear that either $\alpha+\beta > \frac{4\pi}{3}$ or $\alpha+\gamma > \frac{4\pi}{3}$, since $\alpha+\beta+\gamma = 2\pi$. Observe that function $\alpha+\beta+2\sin\frac{\alpha+\beta}{2}$ is increasing in $\alpha+\beta$, and when $\alpha+\beta = \frac{4\pi}{3}$, then (1) is upper bounded by $1+\frac{4\pi}{3}+\sqrt{3}$. Observe also that function $\alpha+\gamma+2\sin\frac{\alpha+\gamma}{2}$ is increasing in $\alpha+\gamma$, and when $\alpha+\gamma = \frac{4\pi}{3}$, then expression (1) is upper bounded by $1+\frac{4\pi}{3}+\sqrt{3}$. We conclude that function (1) strictly increases for $\alpha > \frac{2\pi}{3}$. A similar argument shows that function (1) increases if either β or γ exceed $\frac{2\pi}{3}$. This completes the proof of Lemma 1. □

In order to proceed with the analysis of asymmetric-persistent algorithms, we need a simple technical lemma, providing a worst case analysis for a special configuration of healthy searching robots.

Lemma 2. *Consider two robots at arc distance $2\pi - s$ that are about to explore an arc of length s moving in opposing directions (toward each other). Assume also that an exit is located somewhere at the arc of length s. Then, the worst case termination time $g(s)$ is given by the formula*

$$g(s) = \begin{cases} 2\sin(s/2) & \text{,if } s < 2\pi/3 \\ s/2 - \pi/3 + \sqrt{3} & \text{,otherwise.} \end{cases}$$

Proof. (Lemma 2) By symmetry, we may assume that the exit is found after time x by one of the robots, where $0 \leq x \leq s/2$ (see Figure 2). When the message is transmitted that the exit is found, the two robots are at the endpoints of an arc of length $s - 2x$, hence at chord distance $2\sin(s/2-x)$. Hence, the time elapsed till both robots reach the exit is $x + 2\sin(s/2-x)$. The claim follows by the monotonicity of the latest function with respect to x in the interval $[0, s/2]$. This completes the proof of Lemma 2. □

We are now ready to prove Theorem 1, by determining the optimal asymmetric-persistent algorithm.

Lemma 3. *The best* asymmetric-persistent *algorithm has performance ≈ 6.309. The algorithm achieving this bound deploys two robots to the same location on the disc, which they explore in opposing directions. The third robot is deployed at arc-distance β_0 from any of the robots, and starts exploring in opposite direction of the closest robot, where β_0 is the unique root of $3\beta/2+\sqrt{3} = 4\pi/3+2\sin(\beta/2)$ in the interval $[0, 2\pi]$.*

Proof. (Lemma 3) Consider an asymmetric-persistent algorithm that deploys robots r_1, r_2, r_3 as depicted in Figure 3, where $\alpha, \beta > 0$ (the case $\beta = 0$ can be easily seen to induce worse termination time, while the case $\alpha = 0$ is identical to $\gamma = 0$).

There are a number of cases as to which the faulty robot is and where the exit is located. All the cases are summarized in Table 2, where identical cases are also grouped together.

For each case we will determine the worst case running time. Then we will choose α, β, γ so as to minimize the maximum of all these running times.

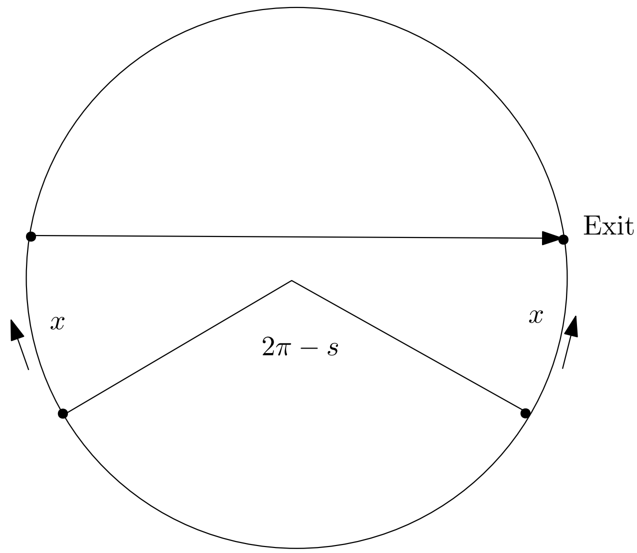

Fig. 2. Exit found and reported after time x. Worst case is $x = 0$, if $s \leq 2\pi/3$, and $x = s/2 - \pi/3$ otherwise.

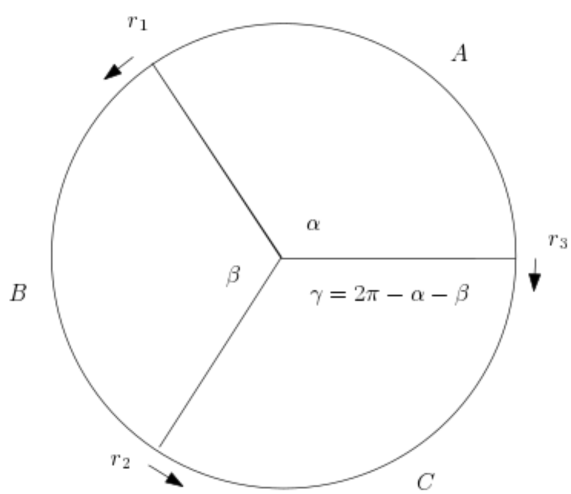

Fig. 3. Robots r_1 and r_2 move counter-clockwise; r_3 moves clockwise. A excludes the starting position of r_1 and r_3; B excludes the starting position of r_2, but includes the starting position of r_1; C includes the starting position of both r_2 and r_3.

	A	B	C
r_1	Case 1	Case 1	Case 2
r_2	Case 3	Case 4	Case 4
r_3	Case 5	Case 6	Case 5

Table 2. The columns indicate the location of the exit. The rows indicate the faulty robot. r_1's initial search position is in B, r_2 and r_3's initial search position are in C.

– *Case 1.* After time γ, robots r_2, r_3 will be at arc distance γ and they will be about to explore an arc of length $\alpha + \beta = 2\pi - \gamma$ moving in opposing directions. Also the exit is located somewhere at the arc of length $2\pi - \gamma$. Hence, by Lemma 2, the (worst case) total termination time will be $1 + \gamma + g(2\pi - \gamma)$ which simplifies to

$$e(\gamma) := \begin{cases} 1 + \gamma + 2\sin(\gamma/2) & \text{,if } \gamma > 4\pi/3 \\ 1 + \gamma/2 + 2\pi/3 + \sqrt{3} & \text{,otherwise.} \end{cases}$$

Also, it is easy to see that $e(\gamma)$ is strictly increasing, a fact we will use later on.

- *Case 2.* The setup is identical to that of Lemma 2 where the arc that holds the exit has arc length $s = \gamma$. Hence, the (worst case) total termination time will be $1 + g(\gamma)$, which is easily seen to be dominated by $e(\gamma)$ of case 1, for every $0 \le \gamma \le 2\pi$.
- *Case 3.* This situation is similar to Case 1, where (instead of γ) robots are at distance $\beta + \gamma$, and they are moving toward each other, and in an arc segment that does not contain the exit. Hence, the worst case termination time is equal to $e(\beta + \gamma)$. Since $e(\cdot)$ is strictly increasing, this case dominates the cost of case 1.
- *Case 4.* This situation is similar to Case 2, where (instead of γ) robots are at distance $\beta + \gamma$ and they are moving toward one another and toward the segment that contains the exit. The maximal total required time is therefore given by the function $1 + g(\beta + \gamma)$, which is easily seen to be dominated by $e(\beta + \gamma)$ of case 3, for all $0 \le \beta + \gamma \le 2\pi$.
- *Case 5.* We treat the case when r_3 is faulty and the exit is either in C or A together. It is clear that r_2 will be the robot that finds the exit. Assume that the exit is located at distance $0 \le x < \alpha + \gamma$ from the initial searching position of r_2 (to ensure that the exit is located in A). Then the total required search time is given by $1 + x + 2\sin\frac{\beta}{2}$, since the distance between r_1, r_2 remains invariant. Clearly, in the worst case, the total required search time is $1 + \alpha + \gamma + 2\sin\frac{\beta}{2}$.
- *Case 6.* This case is identical to case 5, where r_1 will find the exit (instead of r_2, but still β remains their invariant distance), and where the arc that contains the exit has length β (instead of $\alpha + \gamma$). Hence, worst case termination time is equal to $1 + \beta + 2\sin\frac{\beta}{2}$

It follows that the best asymmetric-persistent algorithm is determined by α, β, γ that minimize

$$\max\left\{e(\beta + \gamma), 1 + \alpha + \gamma + 2\sin(\beta/2), 1 + \beta + 2\sin(\beta/2)\right\},$$

i.e. the costs of cases 3, 5, and 6.

First we show that the promised upper bound is achievable. Indeed, we set $\gamma = 0$, so that $\alpha + \beta = 2\pi$. Now we define β_0, by equating the costs of cases 3,5, i.e. as the root of the equation $e(\beta) = 1 + 2\pi - \beta + 2\sin(\beta/2)$. Numerical calculations yield that $\beta_0 \approx 2.96603$, or in other words (by looking at the definition of function $e(\beta)$), β_0 is defined as the solution to the equation $3\beta/2 + \sqrt{3} = 4\pi/3 + 2\sin(\beta/2)$. We conclude that $\gamma = 2\pi - \beta_0 \approx 3.31716 < 4\pi/3$, which induces worst termination time to be the same in cases 3,5 and equal to $1 + 2\pi - \beta_0 + 2\sin(\beta_0/2) \approx 6.30946$, as promised.

Now we prove the above choices are optimal. Indeed, if $\beta + \gamma > 4\pi/3$, then the total termination time cannot be better than the situation where cases 3,5 induce the same cost. Equating the resulting costs, we obtain that $\beta + \gamma + 2\sin((\beta + \gamma)/2) = \alpha + \gamma + 2\sin(\alpha/2)$. Using that $\beta + \gamma = 2\pi - \alpha$, the previous equation yields $\beta - 2\sin(\beta/2) = \alpha - 2\sin(\alpha/2)$, i.e that $\alpha = \beta$. But then $\gamma = 0$ as well. Since $\beta > 4\pi/3$, the induced cost, by case 3, is at least $1 + 4\pi/3 + \sqrt{3} \approx 6.92084$.

Finally, assume that $\beta + \gamma \le 4\pi/3$. For any fixed γ, the total termination time cannot be better than the situation where cases 3,5 induce the same cost. Equating the resulting costs, we obtain that $(\beta + \gamma)/2 + 2\pi/3 + \sqrt{3} = \alpha + \gamma + 2\sin(\beta/2)$. Since $\alpha = 2\pi - \beta - \gamma$, the optimal choice for β

should be β_γ satisfying $3\beta_\gamma/2 + \gamma/2 + \sqrt{3} = 4\pi/3 + 2\sin(\beta_\gamma/2)$. Note that β_γ is a function of γ, hence differentiating both sides of last equation with respect to γ, and after elementary calculations, we obtain that $\beta'_\gamma(3/2 - \cos(\beta_\gamma/2)) = -1/2$. Since $\beta_\gamma > 0$, we obtain that $\cos(\beta_\gamma/2) < 1$ and hence $\beta'_\gamma > -1$. This implies that expression $\beta_\gamma + \gamma$ is strictly increasing in γ, and this linear term appear in the termination time of case 3. Hence, choosing $\gamma = 0$ is indeed optimal. This concludes the proof of Lemma 3. □

2.2 Evacuating in the presence of Byzantine Faults

The main contribution is as follows.

Theorem 2. Byzantine-Evacuation *can be solved in time* $1 + \frac{4\pi}{3} + \sqrt{3} \approx 6.92084$.

Proof. (Theorem 2) The analysis relies on Figure 4. Assume that all three robots r_k, for $k \in \{1, 2, 3\}$, execute the main evacuation Algorithm 1.

The idea of the algorithm is for the robots to traverse the circumference of the disk for a time of $2\pi/3$. Depending on the calls that have been received, the robots have information to either go to the exit or continue traversing the circumference of the disk for another period of time time $\frac{2\pi}{3}$. They can now verify conflicting messages of the correct location of the exit based on the calls that have been made by the other robots so far. Details are being discussed in the sequel.

Algorithm 1: Evacuation with Byzantine Faults

Go to the circumference, at position $\frac{2\pi k}{3}$;
while *r_k's location is not the same as the exit's location* **do**
    for $\frac{2\pi}{3}$ **do**
        follow the circumference clockwise
    if *One robot claims to have found more than one exit* **then**
        Continue execution of algorithm as though the robot remained silent
    if *No information about exit* **then**
        for $\frac{2\pi}{3}$ **do**
            follow the circumference clockwise till exit is either found or reported. Finish
    if *One robot claims to have found the exit* **then**
        Go to closest part of the segment that is claimed to contain the exit;
        Explore entire segment. Finish.
    if *Two robots claim to have found the exit* **then**
        Investigate both exits. Finish.
Inform all robots of the location of the exit.

First note that one time unit is required to reach the circumference of the disc. After $\frac{2\pi}{3}$ additional time units, the entire disc has been explored once. The areas explored by the robots are contiguous but not overlapping. Observe that a Byzantine robot that claims to have found more than one exit is immediately identified as faulty by the healthy robots. Both potential exits are ignored, and the algorithm continues as though the robot had remained silent. If a non-faulty robot finds the exit, it immediately informs all other robots, then stop its exploration. Say without loss of generality that r_1 is healthy. If r_1 finds the exit during the first $\frac{2\pi}{3}$ part of the exploration, then it stops and is done with the execution of its algorithm, in a time at most $1 + \frac{2\pi}{3}$. If it does not find an exit during the first $\frac{2\pi}{3}$ part of the exploration, then we must consider three cases:

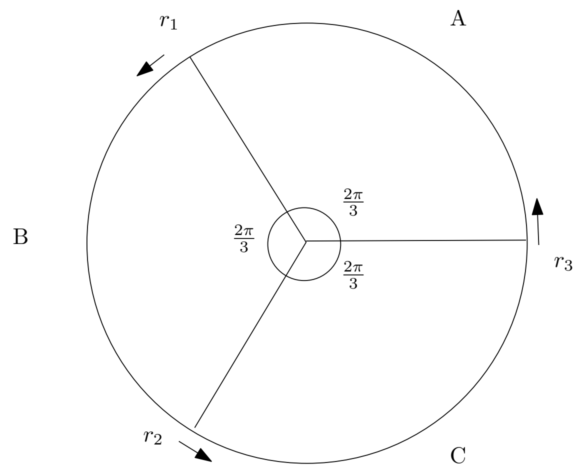

Fig. 4. The initial searching position for r_1, r_2 and r_3 in the Byzantine faults model

- *No exit location reported:* If no exit was found, then keep exploring the circumference of the disk for time $\frac{2\pi}{3}$. Notice that this means that the exit cannot be in B. If the exit is in C, then r_1 has found the exit, and its execution is complete in a time at most $1+\frac{4\pi}{3}$. If the exit is in A, then we learn that r_3 is Byzantine (otherwise, it would have claimed to have found the exit during the first $1+\frac{2\pi}{3}$ of the execution of the algorithm), and r_2 will have correctly identified the location of the exit (Notice that r_1 needs to finish exploring the second arc C to make sure that it was r3 that lied.) Say the exit is located at an arc distance of $0 < x < \frac{2\pi}{3}$ from r_1's current position. Then $2\sin\frac{x}{2}$ is required for r_1 to reach the exit. Since this function is monotone in x for $x \leq \pi$, r_1 can reach the exit in a total time of at most $1+\frac{4\pi}{3}+\sqrt{3}$.
- *One exit location reported:* If one robot other than r_1 claims to have found the exit, we consider two situations: (1) the robot is healthy, in which case the exit is indeed located on the segment where the announcement was made; or (2) the robot is Byzantine, in which case the other two segments have been entirely explored by healthy robots (and are therefore reliably proven to be empty), and the exit is located on the segment where the announcement was made. Notice that in both situations, the only possible location for the exit is on the segment where the announcement was made. If the announcement was made on the segment C, then r_1 explores C immediately, for a total time of at most $1+\frac{4\pi}{3}$. If the announcement was made on the segment A, then r_1 must first reach one end of segment A, which requires $2\sin\frac{2\pi}{3}=\sqrt{3}$ (both ends of the segment are equidistant from r_1's position), then explore the segment, for a total time of at most $1+\frac{4\pi}{3}+\sqrt{3}$.
- *Two exit locations reported:* If both r_2 and r_3 claim to have found an exit, then we know that one of those two claims is true. r_1 will investigate both claims, starting by the closest one. Say r_2 claims to have found the exit at a distance x from its initial searching position, and r_3 claims to have found the exit at a distance y of its initial searching position. Then r_1 must travel an additional $2\sin\frac{x}{2}+2\sin\frac{\frac{2\pi}{3}-x+y}{2}$ to reach both exits. This function is maximised for $x=y=\frac{2\pi}{3}$, for a total time of at most $1+\frac{2\pi}{3}+2\sqrt{3}$.

Observe that both robots r_2 and r_3 execute the same algorithm, and the maximal time required is therefore the same. The adversary will choose the location of the exit and the Byzantine robot in such way as to maximise the total time of execution of the algorithm. Therefore, since $\sqrt{3} < \frac{2\pi}{3}$, this algorithm solves the evacuation problem in total time $1 + \frac{4\pi}{3} + \sqrt{3}$. This completes the proof of Theorem 2. $\square$

3 Lower Bounds for Evacuation Protocols

This section is devoted in proving our main negative results.

Theorem 3. *The following lower bounds are valid.*

(a) Problem Crash-Evacuation *requires time at least* 5.082*.*
(b) Problem Byzantine-Evacuation *requires time at least* 5.948*.*

The lower bound proofs for Crash and Byzantine faults, respectively, admit a unified approach that we detail in the form of a few preliminary lemmata below.

It is easy to observe that if we consider three robots starting from the center of a unit disc then for any $\epsilon > 0$, at time $1 + \frac{2\pi}{3} - \epsilon$ there is an equilateral triangle inscribed in the circle not all of whose vertices have been explored by a robot. However, in the main proof we will make use of an even stronger property of the three robots.

Next we define a useful property $P(T)$, where $T > 0$ denotes time, to be used in the rest of the proof for a lower bound.

Definition 1 (Property $P(T)$). *For any algorithm and any time less than T there are two points on the circle at distance at least $\sqrt{3}$ and each of which was visited at most once by anyone of the three robots.*

Since Property $P(T)$ ensures the existence of two points at distance at least $\sqrt{3}$ which have been visited at most once by the robots, a simple adversarial argument will guarantee that $T + \sqrt{3}$ is a lower bound on evacuation for Byzantine faults (see Lemma 6), while $T + \sqrt{3}/2$ is a lower on evacuation for Crash faults (see Lemma 5). However, before proving these last statements, we are interested to find a T which satisfies property $P(T)$.

Note that property $P(T)$ is monotone increasing in T, in that $P(T) \wedge T' \leq T \Rightarrow P(T')$. Hence, the larger the value of the parameter T for which $P(T)$ is valid the better the lower bound that can be derived.

Lemma 4. *Property $P(1 + 13\sqrt{3}/7)$ is valid.*

Proof. (Lemma 4) In the sequel, to help our intuition, we prove first the weaker statement that $P(4)$ is valid and then we improve this to $P(1 + 13\sqrt{3}/7)$. Let us consider some algorithm at time $< T$, where $T = 4$, and assume by contradiction that all points that have been visited at most once by a robot are at distance less than $\sqrt{3}$ from each other. Clearly, all these points must lie on an arc of length less than $2\pi/3$. Therefore looking at the complement of this arc we find an arc of length longer than $4\pi/3$. In turn, this gives rise to a regular hexagon with five of its vertices inside this last arc each visited twice by a robot. Therefore these five vertices of the hexagon have been visited ten times in total by the three robots. Since there are three robots, it follows that at least one robot must have visited four of these vertices. However this is impossible as $T = 4$. It follows that property $P(4)$ is valid.

Now we derive the main result of the lemma by showing that $P(1 + 13\sqrt{3}/7)$ is valid. We argue as in the previous paragraph, however, instead of selecting five vertices of a regular hexagon we will choose the five points more carefully.

As in the proof of $P(4)$ above, let three points A, B, C be vertices of an equilateral triangle such that every point in the perimeter of the disc which is visited by at most one of the three robots is in the arc clockwise between A and B.

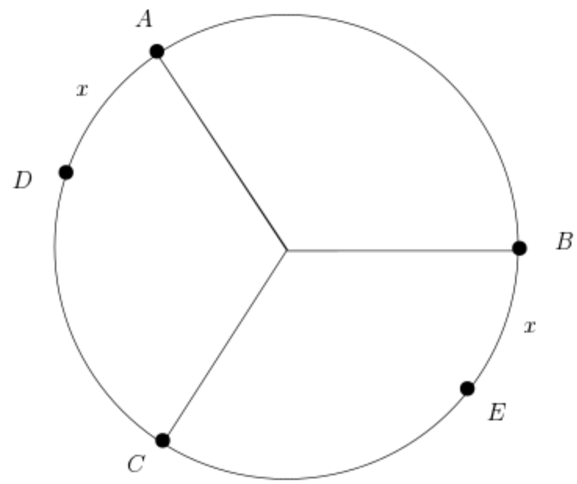

Fig. 5. Evacuation of the second truth telling robot.

In turn, this will give rise to five points on the circumference of the disc with each of its vertices visited twice by a robot; namely choose a point D located between A and C and a point E between B and C so that the length of arc AD is x and this is equal to the length of arc EB (the choice of x will be based on maximizing the length of a path visiting these vertices and will be made precise in the next paragraph). Since there are ten visitations by three robots one of the robots must have visited four consecutive points at least once.

We will show that visiting four vertices among A, B, C, D, E takes time at least $13\sqrt{3}/7 \approx 3.21$. If $x < \pi/3$ then there are 2 candidates for the shortest four-point walk, namely

$$\text{either } D \to A \to B \to E \text{ or } A \to D \to C \to E.$$

Taking into account the lengths of the corresponding chords in these two paths, it turns out that we need to maximize the function $f(x)$ defined by the equation below.

$$f(x) := \min\{\sqrt{3} + 4\sin(x/2), 2\sin(x/2) + 4\sin(\pi/3 - x/2)\}.$$

It is easily seen that the maximum of f is equal to $1 + 13\sqrt{3}/7$ and it is obtained at $x = 4/\arctan(1/(3\sqrt{3}))$. The rest of the reasoning is the same as for $T = 4$ in the first paragraph of the proof. This completes the proof of Lemma 4. □

Proof. (Theorem 3) Now we are ready to conclude the proofs of the two parts of Theorem 3 on crash and Byzantine faults, respectively.

Lower Bound for Crash-Faults The proof of Part (a) follows as a corollary of Lemma 5 below.

Lemma 5. *If property $P(T)$ holds then we can achieve a lower bound of $T + \frac{\sqrt{3}}{2}$ on evacuation in the presence of a crash-faulty robot.*

Proof. (Lemma 5) Identify two points A, B at distance $\geq \sqrt{3}$ each of which was visited at most once by anyone of the three robots. Say r_1 is the robot that visited neither of those points. Set the exit to be the point farthest away from r_1's current location. Clearly, at least $\frac{\sqrt{3}}{2}$ is required for r_1 to reach the point. This proves lemma 5. □

Lower Bound for Byzantine-Faults The proof of Part (b) follows as a corollary of Lemma 6 below.

Lemma 6. *If property $P(T)$ holds then we can achieve a lower bound of $T + \sqrt{3}$ on evacuation in the presence of a Byzantine robot.*

Proof. (Lemma 6) Identify two points A, B at distance $\geq \sqrt{3}$ each of which was visited at most once by anyone of the three robots. Assume without loss of generality that r_1 visited A. Then we have two possibilities to consider: either r_1 also visited B, or (say) r_2 visited B.

If r_1 visited both points, set r_1 to be Byzantine, then wait until either r_2 or r_3 visit either A or B. Once this first visit happens, claim that the exit is located at the other point. The robot that visited the first point will require at least $\sqrt{3}$ to reach the other point, which proves the lemma in this case.

If, say, r_2 visited point B, then have r_1 claim that the exit is located at point B, and r_2 claim that the exit is located at point A (which will happen as soon as the robots reach those points). Then r_3 will have to visit both points to find the real exit, since it has no means of distinguishing the reliable robot from the Byzantine robot. Choose the first point visited by robot r_3 not to have the exit, and set the exit at the location of the other point. Then r_3 requires at least $\sqrt{3}$ to reach the other point, which proves the lemma in this case as well.

Combining these two cases, this completes the proof of Lemma 6. □

If we note the following approximations for the quantities arising in Lemma 4: $1 + 13\sqrt{3}/7 \approx 4.21$ and $4/\arctan(1/(3\sqrt{3})) \approx 0.76$, then the proof of Theorem 3 is complete. □

4 Discussion and open problems

In this paper we considered the evacuation problem on a disc for three robots exactly one of which has either crash or Byzantine faults. We analyzed the problem in both fault scenarios and gave lower bounds as well as evacuation algorithms resulting in upper bounds. There are several challenging open problems. In addition to closing the gaps between the upper and lower bounds for either robot fault (either crash or Byzantine) model with wireless communication presented in our paper, it would be interesting to investigate the evacuation problem

(a) for other types of communication models (e.g., face-to-face, or even limited visibility),
(b) for more than three robots f of which may be faulty and derive asymptotic bounds similar to the results of [11], and
(c) for robots with not necessarily identical maximum speeds.

Despite the fact that obtaining tight bounds for evacuation problems are known often to lead to functions which can be a challenge to optimize, the algorithmic insights derived by this interaction between robot mobility and communication can lead to rewarding applications of distributed computing in search and evacuation.

References

1. N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM Journal on Computing*, 36(1):56–82, 2006.
2. R. Baeza Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Information and Computation*, 106(2):234–252, 1993.
3. R. Baeza-Yates and R. Schott. Parallel searching in the plane. *Computational Geometry*, 5(3):143–154, 1995.
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. Z. Bouzid, M.G. Potop-Butucaru, and S. Tixeuil. Optimal byzantine-rezilient convergence in uni-dimensional robot network. *Theoretical Computer Science*, 411(34-36):3154–3168, 2010.
7. S. Brandt, F. Laufenberg, Y. Lv, D. Stolz, and R. Wattenhofer. Collaboration without communication: Evacuating two robots from a disk. In *Algorithms and Complexity - 10th International Conference, CIAC 2017, Athen, Greece, May 24-26, 2017. Proceedings*, 2017.
8. A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. In *Ad-hoc, mobile, and wireless networks, LNCS*, volume 6811, pages 346–359. Springer, 2011.
9. R. Cohen and D. Peleg. Convergence properties of the gravitational algorithm in asynchronous robot systems. *SIAM Journal of Computing*, 41(1):1516–1528, 2005.
10. R. Cohen and D. Peleg. Convergence of autonomous mobile robots with inaccurate sensors and movements. *SIAM Journal on Computing*, 38(1):276–302, 2008.
11. 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.
12. J. Czyzowicz, L. Gasieniec, A. Kosowski, E. Kranakis, D. Krizanc, and N. Taleb. When patrolmen become corrupted: Monitoring a graph using faulty mobile robots. In *Algorithms and Computation - Proceedings of 26th International Symposium, ISAAC 2015*, pages 343–354, 2015.
13. J. Czyzowicz, K. Georgiou, E. Kranakis, D. Krizanc, L. Narayanan, J. Opatrny, and S. Shende. Search on a line by byzantine robots. In *27th International Symposium on Algorithms and Computation, ISAAC 2016, December 12-14, 2016, Sydney, Australia*, pages 27:1–27:12, 2016.
14. J. Czyzowicz, K. Georgiou, E. Kranakis, L. Narayanan, J. Opatrny, and B. Vogtenhuber. Evacuating robots from a disk using face-to-face communication (extended abstract). In *Algorithms and Complexity - 9th International Conference, CIAC 2015, Paris, France, May 20-22, 2015. Proceedings*, pages 140–152, 2015.
15. J. Czyzowicz, E. Kranakis, D. Krizanc, Narayanan. L., and J. Opatrny. Search on a line with faulty robots. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 405–414, 2016.
16. J. Czyzowicz, E. Kranakis, D. Krizanc, L. Narayanan, J. Opatrny, and S. Shende. Wireless autonomous robot evacuation from equilateral triangles and squares. In *Ad-hoc, Mobile, and Wireless Networks - 14th International Conference, ADHOC-NOW 2015, Athens, Greece, June 29 - July 1, 2015, Proceedings*, pages 181–194, 2015.
17. X. Défago, M. Gradinariu, S. Messika, and P.R. Parvédy. Fault-tolerant and self-stabilizing mobile robots gathering. In *Proceedings of DISC 2006*, pages 46–60, 2006.
18. Y. Dieudonné, A. Pelc, and D. Peleg. Gathering despite mischief. *ACM Transactions on Algorithms (TALG)*, 11(1):1, 2014.
19. J. Hromkovič, R. Klasing, B. Monien, and R. Peine. Dissemination of information in interconnection networks (broadcasting & gossiping). In *Combinatorial network theory*, pages 125–212. Springer, 1996.
20. T. Izumi, S. Souissi, Y. Katayama, N. Inuzuka, X. Défago, K. Wada, and M. Yamashita. The gathering problem for two oblivious robots with unreliable compasses. *SIAM Journal on Computing*, 41(1):26–46, 2012.
21. F. Kuhn, N. Lynch, and R. Oshman. Distributed computation in dynamic networks. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 513–522. ACM, 2010.
22. L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.
23. N. A. Lynch. *Distributed algorithms.* Morgan Kaufmann, 1996.
24. S. Souissi, X. Défago, and M. Yamashita. Gathering asynchronous mobile robots with inaccurate compasses. *Principles of Distributed Systems*, pages 333–349, 2006.
25. Y. Yang, S. Souissi, X. Défago, and M. Takizawa. Fault-tolerant flocking for a group of autonomous mobile robots. *Journal of Systems and Software*, 84(1):29–36, 2011.