Search-and-Fetch with 2 Robots on a Disk: Wireless and Face-to-Face Communication Models [*]

Konstantinos Georgiou[†] [¶] George Karakostas[‡] [¶] Evangelos Kranakis[§] [¶]

April 30, 2019

Abstract

We initiate the study of a problem on *searching and fetching*, motivated by real-life surveillance and search-and-rescue operations where unmanned vehicles, e.g. drones, search for victims in areas of a disaster. In *treasure-evacuation*, we are interested in designing algorithms that minimize the time it takes for a treasure (a victim) to be discovered and brought (fetched) to the exit (shelter) by any of two robots (rescuers) which are performing in a distributed environment (the case of searching and fetching with 1 robot has been previously considered in [27]). The communication protocol between the robots is either *wireless*, where information is shared at any time, or *face-to-face*, where information can be shared only if the robots meet. For both models we obtain upper bounds for fetching the treasure to the exit. Our algorithms make explicit use of the distance between the treasure and the exit, which is assumed to be known in advance, showing this way how partial information of the unknown input can be beneficial. Our main technical contribution pertains to the face-to-face model. More specifically, we demonstrate how robots can exchange information without meeting, effectively achieving a highly efficient treasure-evacuation protocol which is minimally affected by the lack of distant communication. Finally, we complement our positive results above by providing a lower bound in the face-to-face model.

1 Introduction

We introduce the study of a distributed problem on *searching and fetching* called *treasure evacuation*. Two robots are placed at the center of a unit disk, while an exit and a treasure lie at unknown positions on the perimeter of the disk. Robots search with maximum speed 1, and they detect a point of interest (either the treasure or the exit) only if they pass over it. The exit is immobile, while the treasure can be carried by any of the robots. The goal of the search is for at least one of the robots to bring (fetch) the treasure to the exit, i.e. evacuate the treasure, in the minimum possible completion time. The robots do not have to evacuate, they only need to co-operate, possibly by sharing information, so as to learn the locations of the points of interest and bring the treasure to the exit. Contrary to previous work, this is the first time an explicit ordering on the tasks to be performed is imposed on two moving robots. This makes the problem inherently different in nature and more difficult than similarly looking results.

[¶] Research supported in part by NSERC of Canada.

[†] Dept. of Mathematics, Ryerson University, Toronto, ON, Canada, konstantinos@ryerson.ca

[‡] Dept. of Computing & Software, McMaster University, Hamilton ON, Canada, karakos@mcmaster.ca

[§] School of Computer Science, Carleton University, Ottawa ON, Canada, kranakis@scs.carleton.ca

[*] An extended abstract of this paper appeared in the proceedings of the 6th International Conference on Operations Research and Enterprise Systems (ICORES17), available at the SCITEPRESS Digital Library.

Special to our problem is also the underlying advice-model we consider, i.e. that even though the locations of the exit and the treasure are unknown, their distance is known and is considered part of the input. Interestingly, finding an optimal algorithm turns out to be a challenging task even when the robots have this knowledge. We propose treasure-evacuation protocols in two communication models. In the *wireless* model robots exchange information instantaneously and at will, while in the *face-to-face* model information can be exchanged only if the robots meet. We aim at incorporating the knowledge of the exact distance between the exit and the treasure into our algorithm designs. We offer algorithmic techniques such as planning ahead, timing according to the explicit task ordering, and retrieval of unknown information through inference and not communication.

Our problem is motivated by real-life surveillance and search-and-rescue operations where unmanned vehicles, e.g. drones, search for victims in areas of a disaster. Indeed, consider a group of rescuer-mobile-agents (robots), initially located strategically in a central position of a domain. When alarm is triggered and a distress signal is received, robots need to locate a victim (the treasure) and bring her to safety (the exit). Our problem shares similarities also with classic and well-studied cops-and-robbers games; robots rest at a central position of a domain (say, in the centre of a disk as in our setup) till an alarm is triggered by some "robber" (the treasure in our case). Then, robots need to locate the stationary robber and subsequently bring him to jail (the exit). Interestingly, search-and-fetch type problems resemble also situations that abound in fauna, where animals hunt for prey which is then brought to some designated area, e.g. back to the lair. As such, further investigation of similar problems will have applications to real-life rescue operations, as well as to the understanding of animal behavior, as it is common in all search problems.

1.1 Problem Definition & Contributions

A treasure and an exit are located at unknown positions on the perimeter of a unit-disk and at arc distance α (in what follows all distances will be arc-distances, unless specified otherwise). Robots, denoted by R_1, R_2, start from the center of the disk, and can move anywhere on the disk at constant speed 1. Each of the robots detects the treasure or the exit only if its trajectory passes over that point on the disk. Once detected, the treasure can be carried by a robot at the same speed. We refer to the task of bringing the treasure to the exit as *treasure-evacuation*. We use the abbreviations T, E for the treasure and the exit, respectively. For convenience, in the sequel we will refer to the locations of the exit and the treasure as *points of interest (PoI)*. For a PoI I on the perimeter of the disk, we also write $I = E$ ($I = T$) to indicate that the exit (treasure) lies in point I. For a point B, we also write $B = null$ to denote the event that neither the treasure nor the exit is placed on B.

We focus on the following online variants of treasure-evacuation with 2 robots, where the exact distance α between T, E is known, but not their positions:
- In **2-TE**$_w$ (Section 2), information between robots is shared continuously in the time horizon, i.e. messages between them are exchanged instantaneously and at will with no restrictions and no additional cost or delays.
- In **2-TE**$_{f2f}$ (Section 3), the communication protocol between the robots is face-to-face (non-wireless)—abbreviated F2F (or f2f), where information can be exchanged only if the robots meet at the same point anywhere.

Part of our contribution is that we demonstrate how robots can utilize the knowledge of the arc-distance α between the points of interest. We propose protocols that induce worst case evacuation time $1 + \pi - \alpha + 4\sin(\alpha/2)$ for the wireless model and $1 + \pi - \alpha/2 + 3\sin(\alpha/2)$ for the face-to-face model. The worst case cost for the two problems becomes $1 + 2\sqrt{3} + \frac{\pi}{3} \approx 5.5113$ (when $\alpha = \frac{2\pi}{3} \approx 2.0944$) and $1 + 2\sqrt{2} + \pi - \sec^{-1}(3) \approx 5.73906$ (when $\alpha = 2\sec^{-1}(3) \approx 2.46192$), respectively. The upper bound in the face-to-face model, which is our main contribution, is the result of a non-intuitive evacuation protocol that allows robots to exchange information about the topology without meeting, effectively bypassing their inability to communicate from distance. Note that our results induce upper bounds with respect to competitive analysis as well. Indeed, the optimal solution, given that the input is known, equals $opt_\alpha = 1 + 2\sin(\alpha/2)$, hence the competitive ratio we achieve, for fixed α, can be computed by scaling the worst case performance we achieve by opt_α. In both cases, the worst case competitive ratio becomes $1 + \pi$, for $\alpha \to 0$. Finally, we complement our results above by showing that any algorithm in the face-to-face model needs time at least $1 + \pi/3 + 4\sin(\alpha/2)$, if $\alpha \in [0, 2\pi/3]$ and at least $1 + \pi/3 + 2\sin(\alpha) + 2\sin(\alpha/2)$, if $\alpha \in [2\pi/3, \pi]$. A graphical comparison of our results can be seen in Figure 1.

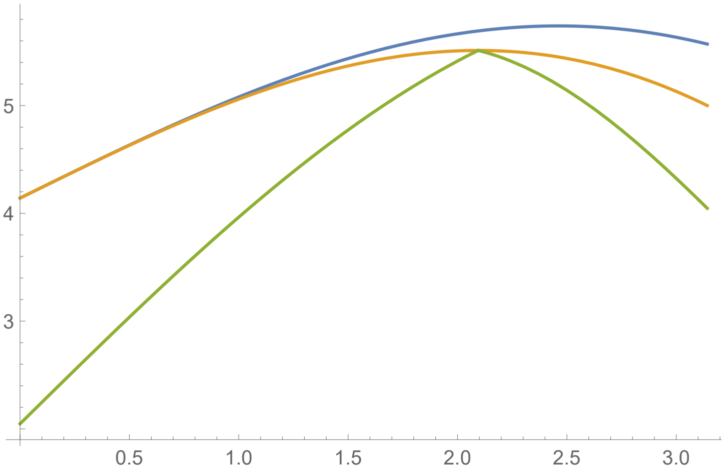

Figure 1: Comparison between the performance of the wireless algorithm (yellow curve), the performance of the f2f algorithm (blue curve) and the provided lower bound (green curve) depicted on the vertical axis, as a function of α (horizontal axis).

1.2 Problem and Model Motivation

From a technical perspective, our communication models are inspired by the recent works on evacuation problems [18, 19, 20]. Notably, the associated search problems are inherently different than our problem which is closer in nature to search-type, treasure-hunt, and exploration problems. Also, our mathematical model features (a) a distributed setting (b) with objective to minimize time, and (c) where different communication models are contrasted. None among (a),(b),(c) are well understood for search games, and, to the best of our knowledge, they have not been studied before in this combination.

Specific to the problem we study are the number of robots (2 and not arbitrarily many - though our results easily extend to swarms of robots), the domain (disk), and the fact the robots have some knowledge about the PoI. Although extending our results to more generic situations is interesting in its own right, the nature of the resulting problems would require a significantly different algorithmic approach. Indeed, our main goal is to study how limitations in communication affect efficiency, which is best demonstrated when the available number of robots, and hence computation power, is as small as possible, i.e. for two robots. In fact, it is easy to extend our algorithms for the n-robot case.

Notably, search-and-fetch problems are challenging even for 1 robot as demonstrated in [27]. In particular, the work of [27] implies that establishing provably optimal evacuation protocols for 2-robots is a difficult task, even when the domain is the disk. Indeed, the best-known trajectory for 1 robot in [27], which is also conjectured to be optimal, exhibits delicate "jumps" that effectively save an almost negligible amount of the termination time, still they improve upon the naive approach. Nevertheless, we view the disc domain that we study as natural. Indeed, a basic setup in search-and-rescue operations is that rescuer-robots inhabit in a base-station, and they stay inactive till they receive a distress signal. As it is common in real-life situations, the signal may only reveal partial information about the location of a victim, e.g. its distance from the base-station, along with the distance between the points. When there are more than one PoI to be located, this kind of information suggests that the points lie anywhere on co-centric circles. When the points are equidistant from the base-station, robots need only consider a disk, as it is the case in our problem. We believe that with enough technical and tedious work, our results can also extend to non-equidistant points, however the algorithmic significance of the proposed distributed solutions may be lost in the technicalities.

Finally, specific to our problem is the underlying advice model. Indeed, robots have access to partial information (the exact distance of the hidden objects) about the unknown input (the exact locations of the hidden objects). Partial knowledge of the input is interesting due to efficiency-information tradeoffs that are naturally induced by online problems, and which are commonly studied in competitive analysis, e.g. see [29, 21] and [27]. In our search-and-fetch problem, the partial information of the distance of the hidden objects demonstrates that robots with primitive communication capabilities are in fact not much less powerful than in the wireless model.The reader may also view this piece of advice as an algorithmic challenge in order to bypass the uncertainty regarding the locations of the PoI. Notably, our algorithms adapt strategies as a function of the distance of the PoI, trying to follow protocols that would allow them to detect the actual positions of the points without necessarily visiting them. As an easy example, note that if a robot has explored already a contiguous arc of length $\alpha+\epsilon$, the discovery of a PoI reveals the location of the other α-arc distant away PoI (our algorithm makes use of distance α in a much more sophisticated way). As a result, had we assumed that distances are unknown, robots may not be able to deduce such important information about the topology using partial exploration, and the problem would require an inherently different algorithmic approach.

1.3 Related Work

Traditional search is concerned with finding an object with specified properties within a search space. Searching in the context of computational problems is usually more challenging especially when the environment is unknown to the searcher(s) (see [1, 4, 40]). This is particularly evident in the context of robotics whereby exploration is taking place within a given geometric domain by a group of autonomous but communicating robots. The ultimate goal is to design an algorithm so as to accomplish the requirements of the search (usually locating a target of unknown a priori position) while at the same time obeying the computational and geographical constraints. The input robot configuration must also accomplish the task in the minimum possible amount of time [11].

Search has a long history. There is extensive and varied research and several models have been proposed and investigated in the mathematical and theoretical computer science literature with particular emphasis on probabilistic search [40], game theoretic applications [4], cops and robbers [12], classical pursuit and evasion [38], search problems as related to group testing [1], searching a graph [36], and many more. A survey of related search and pursuit evasion problems can be found in [17]. In pursuit-evasion, pursuers want to capture evaders who try to avoid capture. Examples include *Cops and Robbers* (whereby the cops try to capture the robbers by moving along the vertices of a graph), *Lion and Man* (a geometric version of cops and robbers where a lion is to capture a man in either continuous or discrete time), etc. Searching for a motionless point target has some similarities with the lost at sea problem, [28, 31], the cow-path problem [9, 10], and with the plane searching problem [7]. This last paper also introduced the "instantaneous contact model", which is referred to as wireless model in our paper. When the mobile robots do not know the geometric environment in advance then researchers are concerned with exploring [2, 3, 25, 30]. Coordinating the exploration of a team of robots is a main theme in the robotics community [14, 41, 42] and often this is combined with the mapping of the terrain and the position of the robots within it [35, 39].

Evacuation for grid polygons has been studied in [26] from the perspective of constructing centralized evacuation plans, resulting in the fastest possible evacuation from the rectilinear environment. There are certain similarities of our problem to the well-known evacuation problem on an infinite line (see [6] and the recent [15]) in that the search is for an unknown target. However, in this work the adversary has limited possibilities since search is on a line. Additional research and variants on this problem can be found in [24] (on searching with turn costs), [34] (randomized algorithm for the cow-path problem), [33] (hybrid algorithms), [8] (searching with different speeds), and many more.

A setting similar to ours is presented in the recent works [18, 21, 19, 20, 37, 13, 16] where algorithms are presented in the wireless and non-wireless (or face-to-face) communication models for the evacuation of a team of robots. The "search domain" in [18, 21, 19, 16] is a unit circle (while in [20] the search domain is a triangle or square), however, unlike our search problem, in these papers all the robots are required to evacuate from an unknown exit on the perimeter. Moreover, in none of these papers is there a treasure to be fetched to the exit. Finally, in some more recent papers [23, 22], Czyzowicz et al. considered the problem of evacuating a distinguished (as in our case) mobile (unlike our case) robot.

The problem we consider is a direct generalization of the search-and-fetch problem of [27] with 1 robot. Unlike in our problem, searching only with 1 robot requires an almost orthogonal approach in order to improve upon the naive strategies. Indeed, the best known trajectory of [27] employs alternating moves along chords and arcs whose lengths depend on the distance of the hidden items. The induced gain is comparable to the difference between the length of an arc and its corresponding chord, and even though this quantity is not significant, it is conjectured that it is indeed the best one can achieve. In contrast, when searching with 2 robots there are more significant gains by carefully synchronizing the moves of the robots.

Our work is also an attempt to analyze theoretically search-and-fetch problems that have been studied by the robotics community since the 90's, e.g. see [32]. A scenario similar to ours (but only for 1 robot) has been introduced by Alpern in [5], where the domain was discrete (a tree) and the approach/analysis resembled that of standard search-type problems [4]. In contrast, our problem is of distributed nature, and our focus is to demonstrate how robots' communication affects efficiency under the assumption that partial information about the input is known.

2 Wireless Model

As a warm-up we present in this section an upper bound for the wireless model, which will also serve as a reference for the more challenging face-to-face model. The algorithmic solution we propose is simple and it is meant to help the reader familiarize with basic evacuation trajectories that will be used in our main contribution pertaining to the face-to-face model.

Theorem 2.1. *For every $\alpha \in [0, \pi]$, problem 2-TE$_w$ can be solved in time $1 + \pi - \alpha + 4\sin(\alpha/2)$.*

To prove Theorem 2.1, we propose Algorithm 1 that achieves the promised bound. Intuitively, our algorithm follows a greedy like approach, adapting its strategy as a function of the distance α of the PoI. If α is small enough, then the two robots move together to an arbitrary point on the disk and start exploring in opposing directions. Otherwise the two robots move to two antipodal points and start exploring in the same direction. Exploration continues till a PoI is found. When that happens, the robot that can pick up the treasure and fetch it to the exit in the fastest time (if all locations have been revealed) does so, otherwise remaining locations are tried exhaustively. Detailed descriptions of the evacuation protocol can be seen in Algorithm 1, complemented by Figure 2.

Noticeably, the performance analysis we give is tight, meaning that for every $\alpha \geq 0$, there are configurations (placements of the PoI) for which the performance of the algorithm is exactly $1+\pi-\alpha+4\sin(\alpha/2)$. Most importantly, the performance analysis makes explicit that two specific naive algorithms that do not adapt strategies together with α are bound to perform strictly worse than our upper bound. Also, the achieved upper bound should be contrasted to the upper bound for the face-to-face model (which is achieved by a much more involved algorithm), which at the same time is only $\alpha/2-\sin(\alpha/2)$ more costly than the bound we show in the wireless model.

Algorithm 1 takes advantage of the fact that robots can communicate to each other wirelessly. This also implies that lack of message transmission is effectively another method of information exchange. In what follows point A will always be the starting point of R_2, and A' denotes its antipodal point. For the sake of the analysis and w.l.o.g. we will assume that R_2 is the one that first finds a PoI $I=\{E,T\}$, say at time $x:=\overset{\frown}{AI}$. We call B,C the points that are at clockwise and counter-clockwise arc-distance α from I respectively. Figure 2 depicts the PoI encountered.

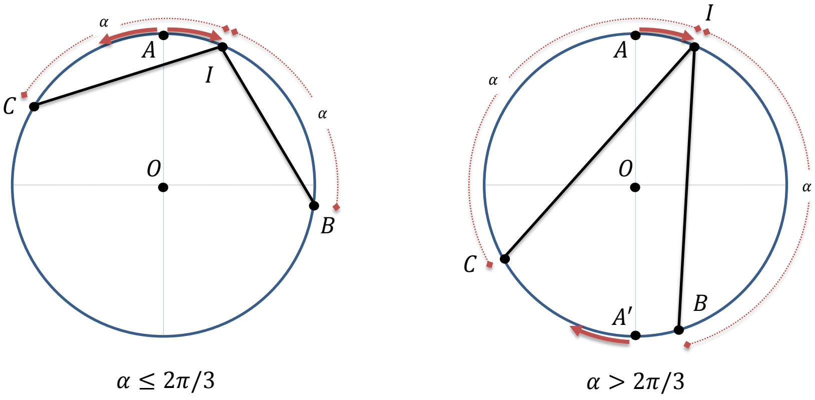

Figure 2: The points of interest for our Algorithm 1.

The description of Algorithm 1 is from the perspective of the robot that finds first a PoI, that we always assume is R_2. Next we assume that the finding of any PoI is instantaneously observed by the two robots. Also, if at any moment, the positions of the PoI are learned by the two robots, then the robots attempt a "confident evacuation" using the shortest possible trajectory. This means for example that if the treasure is not picked up by any robot, then the two robots will compete in order to pick it up and return it to the exit, moving in the interior of the disk.

Correctness of Algorithm 1 is straightforward, since the two robots follow a "greedy-like evacuation protocol" (still, they use different starting points depending on the value of α). Also, the performance analysis of the algorithm, effectively proving Theorem 2.1, is a matter of a straightforward case-analysis. We note that our worst-case analysis is tight, in that for every $\alpha \geq 0$ there exist configurations in which the performance of Algorithm 1 is exactly as promised by Theorem 2.1. Moreover, we may assume that $\alpha > 0$ as otherwise the problem is solved when one PoI is found.

Note that our algorithm performs differently when $\alpha \leq 2\pi/3$ and when $\alpha > 2\pi/3$. Let $x:=\overset{\frown}{AI}$ be the time that R_2 has spent searching till first PoI I is discovered. Then it must be that $x \leq \alpha/2$ and $x \leq \pi-\alpha$

Algorithm 1 Wireless Algorithm

```
Step 1. If $\alpha \leq 2\pi/3$, the two robots move together to an arbitrary point on the ring and start exploring in
        opposing directions, else they move to arbitrary antipodal points $A, A'$ on the cycle and start moving
        in the same direction.
Step 2. Let $I$ be the first PoI discovered by $R_2$, at time $x := \overset{\frown}{AI}$. Let $B, C$ be the points that are at clockwise
        and counter-clockwise arc-distance $\alpha$ from $I$, respectively.
Step 3. If $x \geq \alpha/2$ then robots learn that the other PoI is in $B$, else $R_2$ moves to $B$, $R_1$ moves to $C$.
Step 4. Evacuate
```

for the cases $\alpha \leq 2\pi/3$ and $\alpha > 2\pi/3$ respectively (see also Figure 2). This will be used explicitly in the proof of the next two lemmata. We also assume that R_2 always moves clockwise starting from point A. R_1 either moves counter-clockwise starting from A, if $\alpha \leq 2\pi/3$, or it moves clockwise starting from the antipodal point A' of A, if $\alpha > 2\pi/3$. In every case, the two robots move along the perimeter of the disk till time x when R_2 transmits the message that it found a PoI.

The performance of Algorithm 1 is described in the next two lemmata which admit proofs by case analyses. Each of them examines the relative position of the starting point of robot R_2 (which finds a PoI first) and the two PoI.

Lemma 2.2. *Let A be the starting point of R_2 which is the first to discover a PoI I. Let also the other PoI be at C, where $\overset{\frown}{CI} = \alpha$. If A lies in the arc $\overset{\frown}{CI}$, then the performance of Algorithm 1 is $1 + \pi - \alpha + 4\sin(\alpha/2)$, for all $\alpha \in [0, \pi]$.*

Proof. For the case analysis, we refer to Figure 3. Note that robots spend time 1 to reach the periphery of the disk. Below we calculate the remaining time until evacuation. At time x the cases are as follows.

($I = E, B = null, C = T$ & $\alpha \leq 2\pi/3$): Let R_1 be at point D, i.e. $\overset{\frown}{DA} = x$, see also Figure 3i. Then R_1 moves along the chord CD, it locates the treasure and returns it to the exit I, with total cost

$$\begin{aligned} \overset{\frown}{DA} + \overline{DC} + \overline{CI} &= x + 2\sin(\alpha/2 - x) + 2\sin(\alpha/2) \\ &\overset{(Lemma\ A.1g)}{\leq} \pi - \alpha + 4\sin(\alpha/2)\,. \end{aligned}$$

($I = E, B = null, C = T$ & $\alpha > 2\pi/3$): Let R_1 be at point D, i.e. $\overset{\frown}{DA'} = x$, see also Figure 3ii. Then R_1 moves along the chord CD, it locates the treasure and returns it to the exit I, with total cost

$$\begin{aligned} \overset{\frown}{DA'} + \overline{DC} + \overline{CI} &\leq x + 2\sin(\pi - \alpha - x/2) + 2\sin(\alpha/2) \\ &\overset{(x \leq \pi - \alpha)}{\leq} \pi - \alpha + 4\sin(\alpha/2)\,. \end{aligned}$$

($I = T, B = null, C = E$ & $\alpha \leq 2\pi/3$): When R_2 finds the treasure, it picks it up, and start moving along chord IB, see also Figure 3iii. Meanwhile, R_1 at time x is at some point, say, D, and crosscuts through CD to check the possible point C. When R_1 visits C, R_2 learns where the exit is, so starting

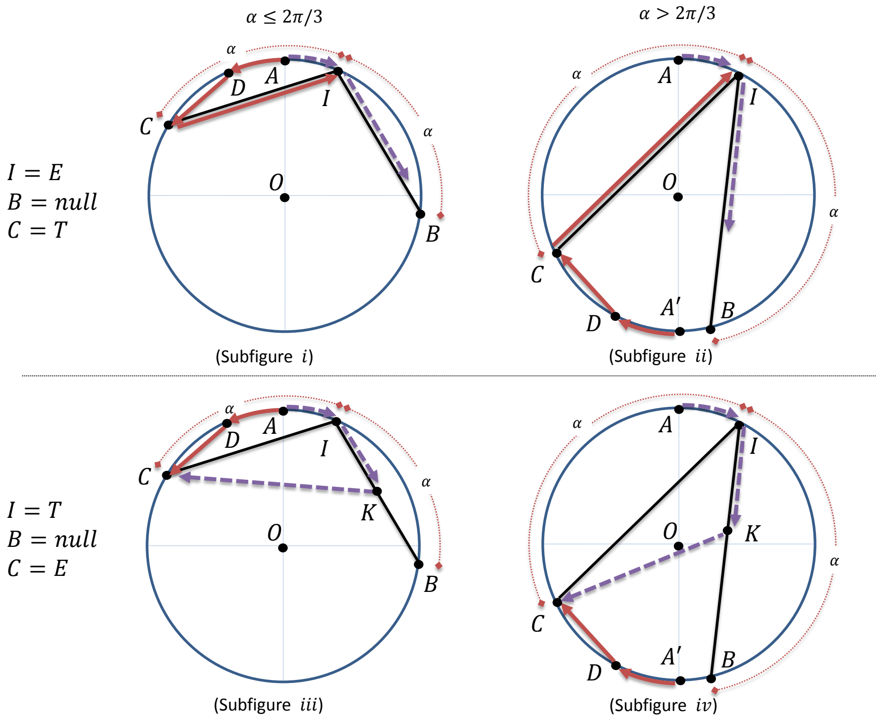

Figure 3: The performance of the wireless algorithm, when the starting point A lies in the arc $\overset{\frown}{CI}$ of the two PoI. The trajectory of R_2 is depicted with the dotted purple curve, while the trajectory of R_1 with the solid red curve.

from point, say, K, it moves along the line segment KC and evacuates. Note that K lies always in the the line segment IB, since $\overline{CD} \leq \overline{CI} = \overline{IB}$). The total cost then is

$$\begin{aligned}
&\overset{\frown}{AI} + \overline{IK} + \overline{KC} = \overset{\frown}{AI} + \overline{CD} + \overline{KC} \\
&\leq \overset{\frown}{AI} + \overline{CD} + \max\{\overline{CI}, \overline{CB}\} \\
&= x + 2\sin(\alpha/2 - x) + \max\{2\sin(\alpha/2), 2\sin(\alpha)\} \\
&\overset{(Lemma\ A.1i)}{\leq} x + 2\sin(\alpha/2 - x) + 2\sin(\alpha) \\
&\overset{(Lemma\ A.1g)}{\leq} \pi - \alpha + 4\sin(\alpha/2).
\end{aligned}$$

($I = T, B = null, C = E$ & $\alpha > 2\pi/3$): When R_2 finds the treasure, it picks it up, and start moving along chord IB, see also Figure 3iv. Meanwhile, R_1 at time x is at some point, say, D, and crosscuts through CD to check the possible point C. When R_1 visits C, R_2 learns where the exit is, so starting from point, say K, it moves along the line segment KC and evacuates. Note that K lies always in the the line segment IB, since

$$\begin{aligned}
\overline{CD} &= 2\sin(\pi/2 - \alpha/2 - x) \\
&\overset{(Lemma\ A.1f)}{\leq} 2\sin(\alpha/2) = \overset{\frown}{IB}.
\end{aligned}$$

In return, the cost becomes

$$\begin{aligned}\overset{\frown}{AI}+\overline{IK}+\overline{KC}=&\overset{\frown}{AI}+\overline{CD}+\overline{KC}\\ \leq&\overset{\frown}{AI}+\overline{CD}+\max\{\overline{CI},\overline{CB}\}\\ =&\,x+2\sin\left(\pi/2-\alpha/2-x\right)+\max\{2\sin\left(\alpha/2\right),2\sin\left(\alpha\right)\}\\ \overset{(Lemma\ A.1i)}{\leq}&\quad x+2\sin\left(\pi/2-\alpha/2-x\right)+2\sin\left(\alpha/2\right)\\ \overset{(Lemma\ A.1h)}{\leq}&\quad \pi-\alpha+4\sin\left(\alpha/2\right).\end{aligned}$$

□

Lemma 2.3. *Let A be the starting point of R_2 which is the first to discover a PoI I. Let also the other PoI be at B, where $\overset{\frown}{IB}=\alpha$. If A lies outside the arc $\overset{\frown}{IB}$, then the performance of Algorithm 1 is $1+\pi-\alpha+4\sin\left(\alpha/2\right)$, for all $\alpha\in[0,\pi]$.*

Proof. For the case analysis below, we rely on Figure 4. As before, robots spend time 1 to reach the periphery of the disk. Below we calculate the remaining time until evacuation. At time x the cases we consider are as follows.

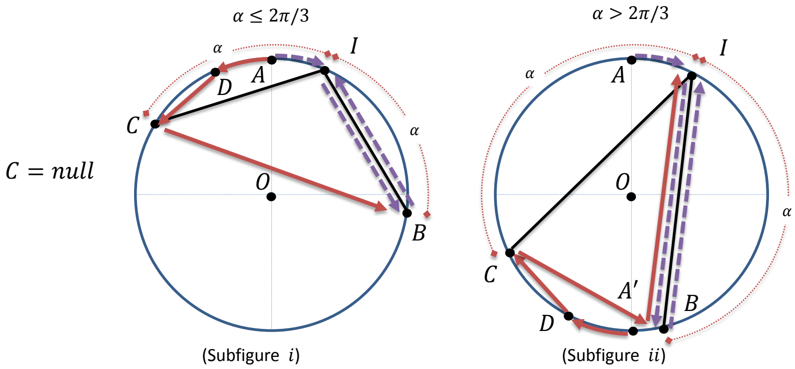

Figure 4: The performance of the wireless algorithm, when the starting point A lies outside the arc $\overset{\frown}{IB}$ of the two PoI. The trajectory of R_2 is depicted with the dotted purple curve, while the trajectory of R_1 with the solid red curve.

($C = null$ & $\alpha\leq 2\pi/3$): After R_2 discovers I it will move along chord IB to discover the other PoI, see also Figure 4i. In particular, since I is visited before C we have $x\leq\alpha/2$. If the treasure is in B, then the total cost would be

$$\begin{aligned}\overset{\frown}{AI}+2\overline{IB}=&x+2\sin\left(\alpha/2\right)\\ \leq&\alpha/2+4\sin\left(\alpha/2\right)\\ \leq&\pi-\alpha+4\sin\left(\alpha/2\right),\end{aligned}$$

(since $\alpha\leq 2\pi/3$), while if the treasure is in I, then the cost would be $\pi-\alpha+2\sin\left(\alpha/2\right)$.

($C = null$ & $\alpha > 2\pi/3$): After R_2 discovers I it will move along chord IB to discover the other PoI, see also Figure 4ii. In particular, since I is visited before C we have that $x \leq \pi - \alpha$. If the treasure is in B, then the two robots are competing as to which will reach the treasure first. Even if R_2 reaches the treasure first, the cost would be $\widehat{AI} + 2\overline{IB} = x + 2\sin(\alpha/2) \leq \pi - \alpha + 4\sin(\alpha/2)$, while if R_1 reaches the treasure first, the total time will be even less than our promised upper bound. Finally, if the treasure is I, then the cost would be by $2\sin(\alpha/2)$ less than our promised upper bound.

□

It follows from Lemmata 2.2, 2.3 that, for all $\alpha \in [0, \pi]$, the overall performance of Algorithm 1 is no more than $1 + \pi - \alpha + 4\sin(\alpha/2)$ concluding Theorem 2.1.

3 Face-to-face Model

The main contribution of our work pertains to the face-to-face model and is summarized in the following theorem.

Theorem 3.1. *For every $\alpha \in [0, \pi]$, problem 2-TE$_{f2f}$ can be solved in time $1 + \pi - \alpha/2 + 3\sin(\alpha/2)$.*

Next we give the high-level intuition of the proposed evacuation-protocol, i.e. Algorithm 2, that proves the above theorem (more low level intuition, along with the formal description of the protocol appears in Section 3.1).

Denote by β the upper bound provided by the theorem above. It should be intuitive that when the distance of the PoI α tends to 0, there is no significant disadvantage due to lack of communication. And although the wireless evacuation-time might not be achievable, a protocol similar to the wireless case should be able to give efficient solutions. Indeed, our face-to-face protocol is a greedy algorithm when α is not too big, i.e. the two robots try independently to explore, locate the PoI and fetch the treasure to the exit without coordination (which is hindered anyways due to lack of communication). Following the worst case analysis, it is easy to observe that as long as α does not exceed a special threshold, call it α_0 (which we define formally later and which is approximately 1.22353), the evacuation time is β, and the analysis is tight.

When α exceeds the special threshold α_0, the lack of communication has a more significant impact on the evacuation time. To work around it, robots need to exchange information which is possible only if they meet. For this reason (and under some technical conditions), robots agree in advance to meet back in the center of the disk to exchange information about their findings, and then proceed with fetching the treasure to the exit. Practically, if the rendezvous is never realized, e.g. only one robot reaches the center up to some time threshold, that should deduce that PoI are not located in certain parts of the disk, potentially revealing their actual location. In fact, this protocol works well, and achieves evacuation time β, as long as α does not exceed a second threshold, which happens to be $2\pi/3$.

The hardest case is when the two PoI are further than $2\pi/3$ apart. Intuitively, in such a case there is always uncertainty as to where the PoI are located, even when one of them is discovered. At the same time, the PoI, hence the robots, might be already far apart when some or both PoI are discovered. As such, meeting at the center of the disk to exchange information would be time consuming and induces evacuation time exceeding β. Our technical contribution pertains exactly to this case. Under some technical conditions, the treasure-finder might need to decide which of the two possible exit-locations to consider next. In this case, the treasure-holder follows a trajectory not towards one of the possible locations of the exit, rather a trajectory closer to that of its peer robot aiming for a rendezvous. The two trajectories are designed carefully so that the location of the exit is revealed no matter whether the rendezvous is realized or not.

3.1 Algorithm & Correctness

In our main Algorithm 2, robots R_1, R_2 that start from the centre of the circle, move together to an arbitrary point A on the circle (which takes time 1). Then they start moving in opposing directions, say, counter-clockwise and clockwise respectively till they locate some PoI.

In what follows we describe only the trajectory of R_2 which is meant to be moving clock-wise (R_1 performs the completely symmetric trajectory, and will start moving counter clock-wise). In particular all point references in the description of our algorithm, and its analysis, will be from the perspective of R_2's trajectory which is assumed to be the robot that first visits either the exit or the treasure at position I. By B, C, D we denote the points on the circle with $\overset{\frown}{DC}=\overset{\frown}{CI}=\overset{\frown}{IB}= \alpha$ (see Figure 5). As before, and in what follows, $I \in \{E, T\}$ represents the position on the circle that is first discovered in the time horizon by any robot (in particular by R_2), and that holds either the treasure or the exit. Finally, O represents the centre of the circle, which is also the starting point of the robots.

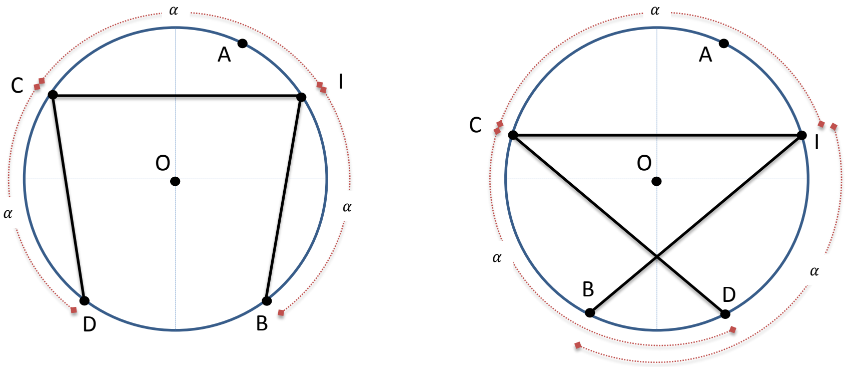

Figure 5: The points of interest from the perspective of R_2, when $\alpha \leq 2\pi/3$ on the left, and when $\alpha \geq 2\pi/3$ on the right.

According to our algorithm, R_2 starts moving from point A till it reaches a PoI I at time $x := \overset{\frown}{AI}$. At this moment, our algorithm will decide to run one of the following subroutines with input x. These subroutines describe evacuation protocols, in which the treasure must be brought to the exit. Occasionally, the subroutines claim that robots evacuate (with the treasure) from points that is not clear that hold an exit. As we will prove correctness later, we comment on these cases by writing that "correctness is pending".

$\mathcal{A}_1$**(x)** (Figure 6 i): If $I = T$, pick up the treasure and move to B along the chord IB. If $B = E$ evacuate, else go to C along the chord BC and evacuate.
(Figure 6 ii): If $I = E$ move to B along the chord IB. If $B = T$, pick up the treasure, and return to I along the chord BI and evacuate. If $B = null$, then go to C along the chord BC. If the treasure is found at C, pick it up, and move to I along the chord CI and evacuate (else abandon the process).

$\mathcal{A}_2$**(x)** (Figure 6 iii): At the moment robots leave point A, set the timer to 0.
If $I = T$, pick up the treasure and go to the centre O of the circle. Wait there till the time $t_0 := \max\{x, \alpha - x + 2\sin(\alpha/2)\} + 1$. If R_1 arrives at O by time t_0, then go to C and evacuate (correctness is pending). Else (if R_1 does not arrive at O by time t_0) go to B and evacuate (correctness is pending).
(Figure 6 iv): If $I = E$, move to B along the chord IB. If $B = T$, pick up the treasure, and return to I along the chord BI and evacuate. If $B = null$, then go to the centre O and halt.

$\mathcal{A}_3$(x) (Figure 6 v): If $I = T$ pick up the treasure. If R_1 is already at point I go to C and evacuate (correctness pending). If R_1 is not at point I, then move along chord ID for additional time $y := \alpha/2 - x + \sin(\alpha/2) + \sin(\alpha)$, and let K be such that $\overline{IK} = y$. If R_1 is at point K, then go to B and evacuate (correctness is pending), else (if R_1 is not at point K) go to C and evacuate (correctness pending).

(Figure 6 vi): If $I = E$, move to B along the chord IB. If $B = T$, pick up the treasure, and return to I along the chord BI and evacuate. If $B = null$, then move along chord BC until you hit C (or you meet the other robot- whatever happens first) and halt at the current point, call it K.

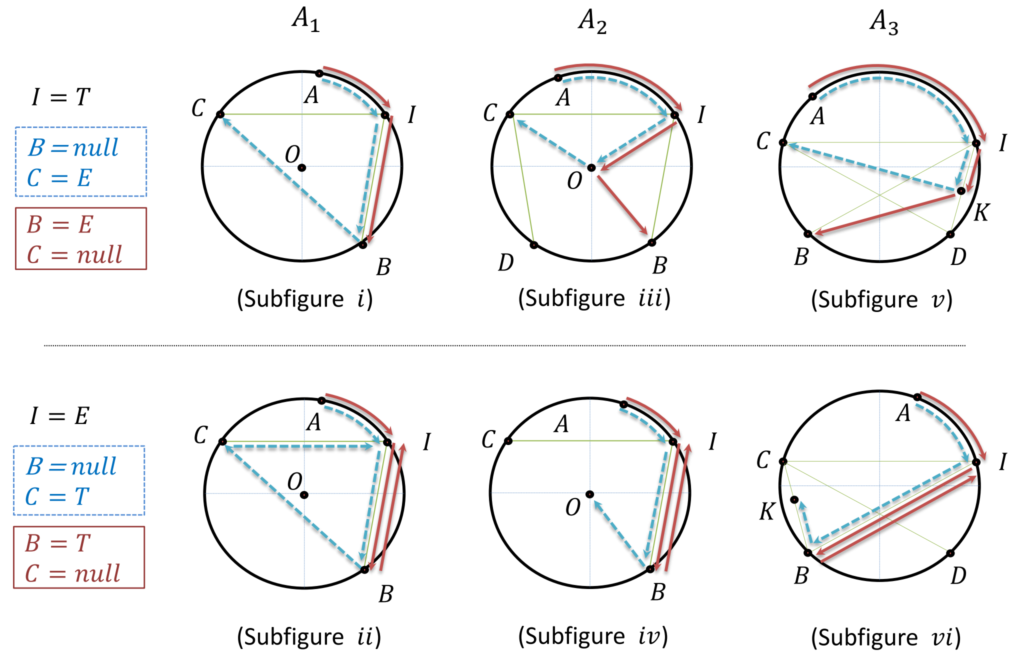

Figure 6: The non-wireless algorithm for two robots with performance $1 + \pi - \alpha/2 + 3\sin(\alpha/2)$.

It is worthwhile discussing the intuition behind the subroutines above. First note that if a robot ever finds a treasure, it picks it up. The second important property is that each robot simulates $\mathcal{A}_1$ either till it finds the treasure or till it fails to find the treasure after finding the exit. At a high level, $\mathcal{A}_1$ greedily tries to evacuate the treasure. This means that if the treasure is found first, then the robot tries successively the possible locations of the exit (using the shortest possible paths) and evacuates. If instead the exit is found, then it successively tries the (at most) two possibilities of the treasure location, and if the treasure is found, it returns it to the exit.

$\mathcal{A}_2$ and $\mathcal{A}_3$ constitute our main technical contribution. Both algorithms are designed so that in some special cases, in which the exact locations of the PoI are not known, the two robots schedule some meeting points so that if the meeting (rendezvous) is realized or even if it is not, the treasure-holder can deduce the actual location of the exit. In other words, we make possible for the two robots to exchange information without meeting. Indeed after finding the treasure, in $\mathcal{A}_2$, R_2 goes to the centre of the ring and waits some finite time till it makes some decision of where to move the treasure, while in $\mathcal{A}_3$, R_2 moves along a carefully chosen (and non-intuitive) chord, and again for some finite time, till it makes a decision to move to a point on the ring. If instead the exit is found early, then the trajectories in $\mathcal{A}_2, \mathcal{A}_3$ are designed to support the other robot which might have found the treasure in case the latter does not follow $\mathcal{A}_1$.

The next non-trivial and technical step would be to decide when to trigger the subroutines above. Of course, once this is determined, i.e. once the trajectories are fixed, correctness and performance analysis is a matter of exhaustive analysis.

We are ready to define our main non-wireless algorithm. We remind the reader that the description is for R_2 that starts moving clockwise. R_1 performs the symmetric trajectory by moving counter-clockwise.

Our main algorithm uses parameter $\overline{x}(\alpha) := 3\alpha/2 - \pi - \sin(\alpha/2) + 2\sin(\alpha)$, which we abbreviate by $\overline{x}$ whenever α is clear from the context. By Lemma A.1a, $\alpha_0 \approx 1.22353$ is the unique root of $\overline{x}(\alpha) = 0$, while $\overline{x}$ is positive for all $\alpha \in (\alpha_0, \pi)$, and negative for all $\alpha \in [0, \alpha_0)$.

Algorithm 2 Non-Wireless Algorithm

```
Step 1. Starting from $A$, move clockwise until a PoI $I$ is found at time $x := \overset{\frown}{AI}$.
Step 2. Proceed according to the following cases:
    • If $\alpha > 2\pi/3$ and $I = T$ and $\alpha > x \geq \alpha - \overline{x}$, then run $\mathcal{A}_3$(x).
    • If $\alpha > 2\pi/3$ and $I = E$ and $x \leq \overline{x}$, then run $\mathcal{A}_3$(x).
    • If $\alpha_0 \leq \alpha \leq 2\pi/3$ and $I = T$ and $\alpha > x \geq \alpha - \overline{x}$, then run $\mathcal{A}_2$(x).
    • If $\alpha_0 \leq \alpha \leq 2\pi/3$ and $I = E$ and $x \leq \overline{x}$, then run $\mathcal{A}_2$(x).
    • In all other cases, run $\mathcal{A}_1$(x).
```

Lemma 3.2. *For every $\alpha \in [0, \pi]$, Algorithm 2 is correct, i.e. a robot brings the treasure to the exit.*

Proof. First, observe that the treasure is always picked up. Indeed, if the first PoI I that is discovered (by any robot) is the treasure, then the claim is trivially true. If the first PoI I found, say, by R_2 is an exit, then R_2 (in all subroutines) first tries the possible location B for the treasure, and if it fails it tries location C (in other words it always simulates $\mathcal{A}_1$ till it fails to find the treasure after finding the exit). Meanwhile R_1 moves counter-clockwise on the ring, and sooner or later will reach C or B. So at least one of the robots will reach the treasure first. In what follows, let R_2 be the one who found first the treasure (and picks it up). We examine three cases.

If R_2 is following subroutine $\mathcal{A}_1$, then the treasure is brought to the exit. Indeed, in that case R_2 expects no interaction from R_1 and greedily tries to evacuate (see subcases i,ii in Figure 6).

If R_2 is following subroutine $\mathcal{A}_2$, then it must be that $\alpha_0 \leq \alpha \leq 2\pi/3$, and $\alpha - \overline{x} \leq x < \alpha$, and that it has not found any other PoI before (by Lemma A.1a we have $\alpha - \overline{x} < \alpha$ and $\overline{x} > 0$, for all $\alpha > \alpha_0$). Figure 6 subcase iii depicts this scenario, where $I = T$. Note that from R_2's perspective, the exit can be either in B or in C, and R_2 chooses to go to the center. This takes total time $x + 1$. If the exit was at point C, then R_1 would have found it in time $\alpha - x \leq \overline{x}$ and that would make it to follow $\mathcal{A}_2$. So, R_1 would first check point D (where the treasure is not present), and that would make it to go to the centre arriving at time $\alpha - x + 2\sin(\alpha/2) + 1$ (an illustration of this trajectory is shown in Figure 6 subcase iv, if R_1 was moving clockwise). R_2 is guaranteed to wait at the center till time t_0 (which is the maximum required time that takes each robot to reach the centre). In that case, R_2 meets R_1 at the center (because R_1 did find the exit in C), and R_2 correctly chooses C as the evacuation point. Finally, if instead the exit was not in C, then R_1 would not make it to the centre by time t_0. That can happen only if the exit is at point B, and once again R_2 makes the right decision to evacuate from B.

In the last case, R_2 is following subroutine $\mathcal{A}_3$, and so it must be that $\alpha > 2\pi/3$, that $\alpha - \overline{x} \leq x < \alpha$, and that it has not found any other PoI before. Figure 6 subcase v depicts this scenario. Note that the exit could be either in C or in D.

If the exit is in C, then $\alpha - x \leq \overline{x}$, and R_1 would follow $\mathcal{A}_3$ too. This means, R_1 would go to point D (where there is no treasure), and that would make it travel along the chord DI (an illustration of this trajectory is shown in Figure 6 subcase vi, if R_1 was moving clockwise). If R_1 reaches I, it waits there, and when R_2 arrives in I, R_2 makes the right decision to evacuate from C. Otherwise R_1 does not reach I, and it moves up to a certain point on the chord ID similarly to R_2. Note that the meeting condition on a point K on the chord, with $y = \overline{IK}$, would be that $\overset{\frown}{AI} + \overline{IK} = \overset{\frown}{CA} + \overline{CD} + (\overline{DI} - \overline{IK})$, which translates into $y = x + \sin(\alpha/2) + \sin(\alpha) - \alpha/2$, i.e. the exact segment of ID that R_2 traverses before it changes trajectory. The longest R_2 could have traveled on the chord ID would be when $x = \alpha - \overline{x}$, but then $\overline{IK}$ would be equal to $\alpha - \pi + 3\sin(\alpha) \leq 2\sin(\alpha) = \overline{ID}$, for all $\alpha > 2\pi/3$. Therefore, the two robots meet indeed in somewhere on the chord ID. Note also that in this case, R_2 makes the right decision and goes to point C in order to evacuate.

If instead the exit is in B, then again R_2 travels till point K (which is in the interior of the chord ID). But in this case, R_1 will not meet R_2 in point K as it will not follow $\mathcal{A}_3$. Once again, R_2 makes the right decision, and after arriving at K it moves to point B and evacuates. □

3.2 Algorithm Analysis

In this section we prove that, for all $\alpha \in [0, \pi]$, the evacuation time of Algorithm 2 is no more than $1 + \pi - \alpha/2 + 3\sin(\alpha/2)$, concluding Theorem 3.1. In the analysis below we provide, whenever possible, supporting illustrations, which for convenience may depict special configurations. In the mathematical analysis we are careful not to make any assumptions for the configurations we are to analyze.

It is immediate that when a robot finds the first PoI at time $x \geq \alpha$ after moving on the perimeter of the disk, then that robot can also deduce where the other PoI is located. In that sense, it is not surprising that, in this case, the trajectory of the robots and the associated cost analysis are simpler.

Lemma 3.3. *Let x be the time some robot is the first to reach a PoI $I \in \{E, T\}$ from the moment robots start moving in opposing directions. If $x \geq \alpha$, then the performance of the algorithm is at most $1 + \pi - \alpha/2 + 3\sin(\alpha/2)$. Also, $x \geq \alpha$ is impossible, if $\alpha > 2\pi/3$.*

Proof. Note that 1 is the time it takes both robots to reach a point, say A, on the ring. So we will tailor our analysis to the evacuation time from the moment robots start moving (in opposing directions) from point A.

Let x be the time after which R_2 (without loss of generality) is the first to find a PoI $I \in \{E, T\}$. Let also B be the other PoI $\{E, T\} \setminus I$. For R_2 to reach first I, it must be the case that R_1 does not have enough time to reach B, and hence $x \leq 2\pi - \alpha - x$, that is $x \leq \pi - \alpha/2$. Since also $x \geq \alpha$, we conclude that $\alpha \leq 2\pi/3$.

Next we examine the following cases. For our analysis, the reader can use Figure 5 as reference (although A is depicted in the interior of the arc CI, we will not use that $\overset{\frown}{AI} \leq \alpha$).

Case 1 ($I = T$): R_2 picks up the treasure and moves along the chord $\overline{IB} = 2\sin(\alpha/2)$. The worst case treasure-evacuation time then is $\max_{\alpha \leq x \leq \pi - \alpha/2} \{x + 2\sin(\alpha/2)\} = \pi - \alpha/2 + 2\sin(\alpha/2)$.

Case 2 ($I = E$): According to the algorithm, R_2 moves towards the treasure point B along the chord IB, and reaches it in time $x + 2\sin(\alpha/2)$. R_1 moves counter-clock wise and will reach the position of the treasure in time $2\pi - \alpha - x$. Whoever finds the treasure first will evacuate from the exit, paying additional time $2\sin(\alpha/2)$. Hence, the total cost can never exceed

$$\begin{aligned}&\min\{x + 2\sin(\alpha/2), 2\pi - \alpha - x\} + 2\sin(\alpha/2)\\ \leq &\pi - \alpha/2 + 3\sin(\alpha/2). \qquad \text{(by Lemma A.1b)}\end{aligned}$$

Observe that in both cases, the cost of the algorithm is as promised. □

By Lemma 3.3 we can focus on the (much more interesting) case that R_2, which is the first robot that finds a PoI, arrives at I at time $x := \overset{\frown}{AI} < \alpha$. A reference for the analysis below is Figure 6 which is accurately depicting point A at most α arc-distance away from I. For the sake of better exposition, we examine next the cases $\alpha \leq 2\pi/3$ and $\alpha \geq 2\pi/3$ separately. Note that in the former case robots may run subroutines $\mathcal{A}_1$ or $\mathcal{A}_2$, while in the latter case robots may run subroutines $\mathcal{A}_1$ or $\mathcal{A}_3$. For the lemma below, the reader may consult Figures 5 and 6.

Lemma 3.4. *Let x be the time some robot is the first to reach a PoI $I \in \{E, T\}$ from the moment robots start moving in opposing directions. If $x < \alpha$, then the performance of the algorithm is at most $1 + \pi - \alpha/2 + 3\sin(\alpha/2)$, for all $\alpha \in [0, \pi]$.*

Proof. As before, we omit in the analysis below the time cost 1, i.e. the time robots need to reach the periphery of the disk. We examine the following cases for R_2, which is the robot that finds I.

($I = T, B = E, C = null$): If R_2 runs $\mathcal{A}_1$, then it must be that $x \leq \alpha - \overline{x}$, so the cost is $x + 2\sin(\alpha/2) \leq \alpha - \overline{x} + 2\sin(\alpha/2) \leq \pi - \alpha/2 + 3\sin(\alpha/2)$ (see Figure 6 i).

If R_2 runs $\mathcal{A}_2$, then it must be that $\alpha - \overline{x} \leq x < \alpha$ and $\alpha \leq 2\pi/3$, and the robot goes to the centre in order to learn where the exit is (see Figure 6 iii). Independently of where the exit is, and by Lemma 3.2, R_2 makes the right decision and evacuates in time $1 + \max_{\alpha - \overline{x} \leq x < \alpha}\{x, \alpha - x + 2\sin(\alpha/2)\} + 1 \leq \max\{\alpha, \overline{x} + 2\sin(\alpha/2)\} + 2$ which, by Lemma A.1c, is at most $\pi - \alpha/2 + 3\sin(\alpha/2)$, for all $\alpha \leq 2\pi/3$. Note that the analysis of this case is valid, even if $I = T$ is not the first PoI that is discovered, and it is from the perspective of the robot that finds the treasure.

If R_2 runs $\mathcal{A}_3$, then it must be that $\alpha - \overline{x} \leq x < \alpha$ and $\alpha > 2\pi/3$. Then the trajectory of R_2 is as in Figure 6 v, and the exit is found correctly due to Lemma 3.2. For the sake of the exposition, we will do the worst case analysis for both cases $B = E$ and $C = E$ now (i.e. we only insist in that $I = T$ and that R_2 runs $\mathcal{A}_3$).

The total time for the combined cases is $\overset{\frown}{AI} + \overline{IK} + \max\{\overline{KB}, \overline{KC}\}$, where $\overline{IK} = y$ (see definition of $\mathcal{A}_3$). Since as we have proved, K lies always in chord ID, and since $\overset{\frown}{DB} = 3\alpha - 2\pi$ we have that

$$\begin{aligned}\overline{BK} &\leq \max\{\overline{BI}, \overline{BD}\}\\ &\leq \max\{2\sin(\alpha/2), 2\sin(3\alpha/2 - \pi)\}\\ &\leq 2\sin(\alpha/2).\end{aligned}$$

We also have that $\overline{KC} \leq \overline{CI} = 2\sin(\alpha/2)$. So the cost becomes no more than

$$\begin{aligned}&x + y + 2\sin(\alpha/2) = \alpha/2 + 3\sin(\alpha/2) + \sin(\alpha)\\ \leq &\pi - \alpha/2 + 3\sin(\alpha/2), \qquad \text{(by Lemma A.1d)}\end{aligned}$$

for all $\alpha \in [0, \pi]$.

($I = T, B = null, C = E$): Since I is found first, we must have $x \leq \alpha/2$, hence both robots run $\mathcal{A}_1$, see Figure 6 i. Robot R_2 that finds the treasure will evacuate in time no more than $x + 2\sin(\alpha/2) + 2\sin(\alpha) \leq \alpha/2 + 2\sin(\alpha/2) + 2\sin(\alpha) < \pi - \alpha/2 + 3\sin(\alpha/2)$, for all $\alpha \in [0, \pi]$.

($I = E, B = T, C = null$): If R_2 is the first to find the treasure, then this case is depicted in Figure 6 i. This happens exactly when $x + 2\sin(\alpha/2) \leq 2\pi - x - \alpha$, so that the total evacuation time is $x + 4\sin(\alpha/2) \leq \pi - \alpha/2 + 3\sin(\alpha/2)$, for all $\alpha \in [0, \pi]$.

Otherwise $x > \pi - \alpha/2 - \sin(\alpha/2)$, and R_1 is the robot that reaches the treasure first. If R_1 decides to run $\mathcal{A}_1$, then the cost would be $2\pi - x - \alpha + 2\sin(\alpha/2) < \pi - \alpha/2 + 3\sin(\alpha/2)$, for all $\alpha \in [0, \pi]$. Finally, if R_1 decides to run $\mathcal{A}_2$ or $\mathcal{A}_3$, then we have already made the analysis in case $I = T, B = E, C = null$ above.

($I = E, B = null, C = T$): Note that in all cases, both robots will run the same subroutine. In particular, if robots run either $\mathcal{A}_2$ or $\mathcal{A}_3$, then we have already done the analysis in case $I = T, B = E, C = null$ above.

Finally, if both robots run $\mathcal{A}_1$, it must be either because $\alpha \leq \alpha_0$, or because $x \geq \overline{x}$, while the cost is always $\alpha - x + 2\sin(\alpha/2) + 2\sin(\alpha)$ (the case is depicted in Figure 6 ii, with reverse direction). If $\alpha \leq \alpha_0$, then the evacuation cost would be at most $\alpha + 2\sin(\alpha/2) + 2\sin(\alpha)$ which by Lemma A.1e is at most $\pi - \alpha/2 + 3\sin(\alpha/2)$, for all $\alpha \in [0, \alpha_0]$. If $x \geq \overline{x}$, then the cost would be at most $\alpha - \overline{x} + 2\sin(\alpha/2) + 2\sin(\alpha) = \pi - \alpha/2 + 3\sin(\alpha/2)$.

□

Note that Lemmata 3.3, 3.4 imply that the performance of Algorithm 2 is, in the worst case, no more than $1 + \pi - \alpha/2 + 3\sin(\alpha/2)$, concluding also Theorem 3.1.

4 Lower Bounds

We conclude the study of treasure evacuation with 2 robots by providing the following lower bound pertaining to distributed systems under the face-to-face communication model.

Theorem 4.1. *For problem 2-TE$_{f2f}$, any algorithm needs at least time* $1 + \pi/3 + 4\sin(\alpha/2)$ *if* $0 \leq \alpha \leq 2\pi/3$*, or* $1 + \pi/3 + 2\sin(\alpha) + 2\sin(\alpha/2)$ *if* $2\pi/3 \leq \alpha \leq \pi$.

For the proof, we invoke an adversary (not necessarily the most potent one), who waits for as long as there are three points A, B, C with $AB = BC = \alpha$ on the periphery such that at most one of them has been visited by a robot. Then depending on the moves of the robots decides where to place the PoI.

Proof of Theorem 4.1. Since the robots start from the center, they'll need time 1 to reach the periphery. The adversary (not necessarily the most potent one, but with this weaker adversary we still get a (weaker) lower bound) will wait for as long as there are three points A, B, C with $AB = BC = \alpha$ on the periphery such that at most one of them has been visited by a robot. Observe that this will be true for as long as less than $2\pi/3$ of the periphery has been explored; this will be done by the 2 robots after time at least $(2\pi/3)/2 = \pi/3$. Hence, after time at least $1 + \pi/3$ there are such points A, B, C with only one of these points visited by a robot. For convenience, we assume that robot 1 is the first to visit a point at time t and robot 2 visits a different point next at time $t + \varepsilon$ (if this doesn't happen, then the optimal algorithm would be behaving like the case of only one robot, which is clearly suboptimal for the adversary moves below). It will be apparent below that the lower bound becomes weaker for $\varepsilon = 0$, so that's what we will assume from now on. We distinguish the following cases:

Case 1 (Robot 1 at A, Robot 2 at C): If the adversary places $T \to B, E \to A$ or C, then recovery needs at least time $4\sin(\alpha/2)$ (if robot 1 or 2 respectively evacuates T by itself). If it places $T \to A, E \to C$, then recovery needs at least time $2\sin(\alpha)$ (a robot evacuates T by traversing AC). Any other placement of T, E by the adversary gives either the same or a worse (lower) bound, and, therefore, it's discarded. It is clear that, in this case, the adversary goes with the first option, for a lower bound of $4\sin(\alpha/2)$.

Case 2 (Robot 1 at A, Robot 2 at B): If the adversary places $T \to C, E \to A$, then recovery needs at least time $\min\{2\sin(\alpha/2) + 2\sin(\alpha), 4\sin(\alpha)\}$ (if robot 2 or 1 respectively evacuates T by itself). If it places $T \to C, E \to B$, then recovery needs at least time $\min\{4\sin(\alpha/2), 2\sin(\alpha) + 2\sin(\alpha/2)\}$ (if robot 2 or 1 respectively evacuates T by itself). Any other placement of T, E by the adversary gives either the same or a worse (lower) bound, and, therefore, it's discarded. It is clear that, in this case, the adversary goes with the option that maximizes the lower bound, for a lower bound of

$$\max\left\{ \begin{array}{l} \min\{2\sin(\alpha/2) + 2\sin(\alpha), 4\sin(\alpha)\}, \\ \min\{4\sin(\alpha/2), 2\sin(\alpha) + 2\sin(\alpha/2)\} \end{array} \right\}.$$

By taking the minimum of Cases 1,2 above, the lower bound of the theorem follows. □

5 Conclusion

In this paper we introduced a new problem on *searching and fetching* which we called *treasure-evacuation* from a unit disk. We studied two online variants of treasure-evacuation with two robots, based on different communication models. The main point of our approach was to propose distributed algorithms by a collaborative team of robots. Our main results demonstrate how robot communication capabilities affect the treasure evacuation time by contrasting face-to-face (information can be shared only if robots meet) and wireless (information is shared at any time) communication.

There are several open problems in addition to sharpening our bounds, and in particular to giving lower bounds that would effectively separate the two communication models. Other variations of our problem include the consideration of different 1) number of robots, 2) geometric domains (discrete or continuous), 3) robot starting positions, 4) number of hidden objects, 5) communication models, 6) robots' speeds, 7) a priori knowledge of the topology or partial information about the targets (e.g. a bound on the distance of the hidden items or no information at all), etc. In each case, the challenging task is to establish either tight bounds, or to separate closely related problems, e.g. the problem in which either the exact distance vs a bound on the distance of the hidden items is known.

When it comes to searching with multiple robots, our 2-robot algorithms can be easily extended to the n-robot case. Assuming that n is even (otherwise we ignore one robot) we split the robots into pairs, defining points in intervals of length $4\pi/n$ on the cycle, assigning each pair of robots to each such poin. Then, we let them run the corresponding 2-robot algorithm. Would that strategy be improvable? We anticipate that nearly optimal algorithms for small number of robots, e.g. for $n = 3, 4$, or any other variation of problem we consider will require new and significantly different algorithmic ideas than those we propose here, still in the same spirit.

References

[1] R. Ahlswede and I. Wegener. *Search problems*. Wiley-Interscience, 1987.

[2] S. Albers and M. R. Henzinger. Exploring unknown environments. *SIAM Journal on Computing*, 29(4):1164–1188, 2000.

[3] S. Albers, K. Kursawe, and S. Schuierer. Exploring unknown environments with obstacles. *Algorithmica*, 32(1):123–143, 2002.

[4] S. Alpern and S. Gal. *The theory of search games and rendezvous*. Springer, 2003.

[5] Steve Alpern. Find-and-fetch search on a tree. *Operations Research*, 59(5):1258–1268, 2011.

[6] R. Baeza Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Information and Computation*, 106(2):234–252, 1993.

[7] R. Baeza-Yates and R. Schott. Parallel searching in the plane. *Computational Geometry*, 5(3):143–154, 1995.

[8] Evangelos Bampas, Jurek Czyzowicz, Leszek Gasieniec, David Ilcinkas, Ralf Klasing, Tomasz Kociumaka, and Dominik Pajak. Linear search by a pair of distinct-speed robots. In Jukka Suomela, editor, *Structural Information and Communication Complexity - 23rd International Colloquium, SIROCCO 2016, Helsinki, Finland, July 19-21, 2016, Revised Selected Papers*, volume 9988 of *Lecture Notes in Computer Science*, pages 195–211, 2016.

[9] A. Beck. On the linear search problem. *Israel Journal of Mathematics*, 2(4):221–228, 1964.

[10] R. Bellman. An optimal search. *SIAM Review*, 5(3):274–274, 1963.

[11] P. Berman. On-line searching and navigation. In A. Fiat and G. J. Woeginger, editors, *Online Algorithms: The State of the Art*, pages 232–241. Springer, 1998.

[12] A. Bonato and R. Nowakowski. *The game of cops and robbers on graphs*. AMS, 2011.

[13] Sebastian Brandt, Felix Laufenberg, Yuezhou Lv, David Stolz, and Roger Wattenhofer. Collaboration without communication: Evacuating two robots from a disk. In Dimitris Fotakis, Aris Pagourtzis, and Vangelis Th. Paschos, editors, *Algorithms and Complexity: 10th International Conference, CIAC 2017, Athens, Greece, May 24-26, 2017, Proceedings*, pages 104–115, Cham, 2017. Springer International Publishing.

[14] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider. Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21(3):376–386, 2005.

[15] Marek Chrobak, Leszek Gasieniec, Thomas Gorry, and Russell Martin. Group search on the line. In Giuseppe F. Italiano, Tiziana Margaria-Steffen, Jaroslav Pokorný, Jean-Jacques Quisquater, and Roger Wattenhofer, editors, *SOFSEM*, volume 8939 of *Lecture Notes in Computer Science*, pages 164–176. Springer, 2015.

[16] H. Chuangpishit, K. Georgiou, and P. Sharma. Average case - worst case tradeoffs for evacuating 2 robots from the disk in the face-to-face model. In *14th International Symposium on Algorithms and Experiments for Wireless Sensor Networks, ALGOSENSORS'18*, volume 11410 of *Lecture Notes in Computer Science*, pages 62–82. Springer, 2018.

[17] T. H. Chung, G. A. Hollinger, and V. Isler. Search and pursuit-evasion in mobile robotics. *Autonomous robots*, 31(4):299–316, 2011.

[18] 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*, pages 122–136. Springer, Austin, Texas, 2014.

[19] J. Czyzowicz, K. Georgiou, E. Kranakis, L. Narayanan, J. Opatrny, and B. Vogtenhuber. Evacuating robots from a disc using face to face communication. In *CIAC 2015*, pages 140–152. Springer, Paris, France, 2015.

[20] J. Czyzowicz, E. Kranakis, D. Krizanc, L. Narayanan, J. Opatrny, and S. Shende. Wireless autonomous robot evacuation from equilateral triangles and squares. In *ADHOC-NOW 2015, Athens, Greece, June 29 - July 1, 2015, Proceedings*, pages 181–194, 2015.

[21] Jurek Czyzowicz, Stefan Dobrev, Konstantinos Georgiou, Evangelos Kranakis, and Fraser MacQuarrie. Evacuating two robots from multiple unknown exits in a circle. *Theoretical Computer Science*, 709:20–30, 2018.

[22] Jurek Czyzowicz, Konstantinos Georgiou, Ryan Killick, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, Jaroslav Opatrny, and Sunil Shende. Priority evacuation from a disk using mobile robots. In *International Colloquium on Structural Information and Communication Complexity*, pages 392–407. Springer, 2018.

[23] Jurek Czyzowicz, Konstantinos Georgiou, Ryan Killick, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, Jaroslav Opatrny, and Sunil M. Shende. God save the queen. In Hiro Ito, Stefano Leonardi, Linda Pagli, and Giuseppe Prencipe, editors, *9th International Conference on Fun with Algorithms, FUN 2018, June 13-15, 2018, La Maddalena, Italy*, volume 100 of *LIPIcs*, pages 16:1–16:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[24] E. D. Demaine, S. P. Fekete, and S. Gal. Online searching with turn cost. *Theoretical Computer Science*, 361(2):342–355, 2006.

[25] X. Deng, T. Kameda, and C. Papadimitriou. How to learn an unknown environment. In *FOCS*, pages 298–303. IEEE, 1991.

[26] S. Fekete, C. Gray, and A. Kröller. Evacuation of rectilinear polygons. In *Combinatorial Optimization and Applications*, pages 21–30. Springer, 2010.

[27] Konstantinos Georgiou, George Karakostas, and Evangelos Kranakis. Search-and-fetch with one robot on a disk. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 80–94. Springer, 2016.

[28] B. Gluss. An alternative solution to the lost at sea problem. *Naval Research Logistics Quarterly*, 8(1):117–122, 1961.

[29] Christoph Hipke, Christian Icking, Rolf Klein, and Elmar Langetepe. How to find a point on a line within a fixed distance. *Discrete Appl. Math.*, 93(1):67–73, 1999.

[30] F. Hoffmann, C. Icking, R. Klein, and K. Kriegel. The polygon exploration problem. *SIAM Journal on Computing*, 31(2):577–600, 2001.

[31] J. R. Isbell. Pursuit around a hole. *Naval Research Logistics Quarterly*, 14(4):569–571, 1967.

[32] J. S. Jennings, G. Whelan, and W. F. Evans. Cooperative search and rescue with a team of mobile robots. In *ICAR*, pages 193–200. IEEE, 1997.

[33] M.-Y. Kao, Y. Ma, M. Sipser, and Y. Yin. Optimal constructions of hybrid algorithms. *J. Algorithms*, 29(1):142–164, 1998.

[34] 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.

[35] J. Kleinberg. On-line search in a simple polygon. In *SODA*, page 8. SIAM, 1994.

[36] E. Koutsoupias, C. Papadimitriou, and M. Yannakakis. Searching a fixed graph. In *ICALP 96*, pages 280–289. Springer, 1996.

[37] Ioannis Lamprou, Russell Martin, and Sven Schewe. Fast two-robot disk evacuation with wireless communication. In Cyril Gavoille and David Ilcinkas, editors, *Distributed Computing: 30th International Symposium, DISC 2016, Paris, France, September 27-29, 2016. Proceedings*, pages 1–15. Springer Berlin Heidelberg, 2016.

[38] P. Nahin. *Chases and Escapes: The Mathematics of Pursuit and Evasion*. Princeton University Press, 2012.

[39] C. H Papadimitriou and M. Yannakakis. Shortest paths without a map. In *ICALP*, pages 610–620. Springer, 1989.

[40] L. Stone. *Theory of optimal search*. Academic Press New York, 1975.

[41] S. Thrun. A probabilistic on-line mapping algorithm for teams of mobile robots. *The International Journal of Robotics Research*, 20(5):335–363, 2001.

[42] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on Autonomous agents*, pages 47–53. ACM, 1998.

A Trigonometric Inequalities

Lemma A.1. *a) There exists some* $\alpha_0 \in (0, \pi)$ *such that* $3\alpha/2 - \pi - \sin(\alpha/2) + 2\sin(\alpha)$ *is positive for all* $\alpha \in (\alpha_0, \pi)$*, and negative for all* $\alpha \in [0, \alpha_0)$*. In particular,* $\alpha_0 \approx 1.22353$*.*

b) $\min\{x + 2\sin(\alpha/2), 2\pi - \alpha - x\} + 2\sin(\alpha/2) \leq \pi - \alpha/2 + 3\sin(\alpha/2)\,,\ \forall \alpha \in [0, \pi]$.

c) $\max\{\alpha, \overline{x} + 2\sin(\alpha/2)\} + 2 \leq \pi - \alpha/2 + 3\sin(\alpha/2)$ *for all* $\alpha \in [0, 2\pi/3]$.

d) $\alpha + \sin(\alpha) \leq \pi$ *for all* $\alpha \in [0, \pi]$.

e) $\alpha/2 + 2\sin(\alpha) \leq \pi - \alpha + 2\sin(\alpha/2)\,,\ \forall \alpha \in [0, 2\pi/3]$. .

f) $\max_{0 \leq x \leq \pi - \alpha}\{\sin(\pi/2 - \alpha/2 - x)\} \leq \sin(\alpha/2)\,,\ \forall \alpha \in [2\pi/3, \pi]$

g) $\max_{0 \leq x \leq \alpha/2}\{x + 2\sin(\alpha/2 - x)\} + 2\sin(\alpha) \leq \pi - \alpha + 4\sin(\alpha/2)\,,\ \forall \alpha \in [0, 2\pi/3]$.

h) $\max_{0\leq x\leq \pi-\alpha}\{x+2\sin\left(\pi/2-\alpha/2-x\right)\} \leq \pi-\alpha+2\sin\left(\alpha/2\right), \ \forall \alpha \in [2\pi/3,\pi]$

i) $\sin\left(\alpha\right) \leq \sin\left(\alpha/2\right), \ \forall \alpha \in [0,2\pi/3]$, *and*
$\sin\left(\alpha\right) \geq \sin\left(\alpha/2\right), \ \forall \alpha \in [2\pi/3,\pi]$.

Proof of A.1a We observe that

$$\begin{aligned}\frac{\partial}{\partial\alpha}\overline{x}(\alpha) =& \frac{\partial}{\partial\alpha}\left(3\alpha/2-\pi-\sin\left(\alpha/2\right)+2\sin\left(\alpha\right)\right)\\ =&3/2-\cos\left(\alpha\right)+\cos\left(\alpha/2\right).\end{aligned}$$

Observe that the above quantity remains positive for $\alpha < 2\pi/3$, while it is negative for $\alpha > 2\pi/3$. Since $\overline{x}(0) < 0$ and $\overline{x}(2\pi/3) > 0$, it follows that there is a unique root $\alpha_0 \in (0,2\pi/3)$ (which numerically can be estimated to $\alpha_0 \approx 1.22353$). Finally, we see that $\overline{x}(\pi) = \pi-1 > 0$, so $\overline{x}(\alpha)$ remains positive for $\alpha \in [2\pi/3,\pi]$.

Proof of A.1b We observe that $\min\left\{x+2\sin\left(\alpha/2\right), 2\pi-\alpha-x\right\}$ attains its maximum when $x+2\sin\left(\alpha/2\right) = 2\pi-\alpha-x$, in which case its value becomes $\pi-\alpha/2+3\sin\left(\alpha/2\right)$.

Proof of A.1c First we claim that $3\alpha/2-2\sin\left(\alpha/2\right) \leq \pi-2$ for $\alpha \leq 2\pi/3$. This is because $\frac{\partial}{\partial\alpha}\left(3\alpha/2+2\sin\left(\alpha/2\right)\right) = 3/2+\cos\left(\alpha/2\right) > 0$, hence $3\alpha/2-2\sin\left(\alpha/2\right) \leq \pi-\sqrt{3}/2 \leq \pi-2$. This claim immediately shows that $\alpha+2 \leq \pi-\alpha/2+3\sin\left(\alpha/2\right)$ for all $\alpha \in [0,2\pi/3]$.

Now we show that $\overline{x}+2\sin\left(\alpha/2\right)+2 \leq \pi-\alpha/2+3\sin\left(\alpha/2\right)$ for all $\alpha \in [0,2\pi/3]$. For this it suffices to check that $\alpha+\sin\left(\alpha\right)-\sin\left(\alpha/2\right) \leq \pi-1$. To that end we see that $\frac{\partial}{\partial\alpha}\left(\alpha+\sin\left(\alpha\right)-\sin\left(\alpha/2\right)\right) = 1+\cos\left(\alpha\right)-\cos\left(\alpha/2\right)/2 \geq 0$ for all $\alpha \leq 2\pi/3$. Hence $\alpha+\sin\left(\alpha\right)-\sin\left(\alpha/2\right) \leq 2\pi/3 \leq 2\pi/3+\sqrt{3}/2-\sqrt{3}/2 \leq \pi-1$ as wanted.

Proof of A.1d We see that $\frac{\partial}{\partial\alpha}\left(\alpha+\sin\left(\alpha\right)\right) = 1+\cos\left(\alpha\right) \geq 0$, for all $\alpha \in [0,\pi]$. hence, $\alpha+\sin\left(\alpha\right) \leq \pi+\sin\left(\pi\right) = \pi$.

Proof of A.1e We observe that

$$\begin{aligned}&\frac{\partial}{\partial\alpha}\left(3\alpha/2+2\sin\left(\alpha\right)-2\sin\left(\alpha/2\right)\right)\\ =&3/2+2\cos\left(\alpha\right)-\cos\left(\alpha/2\right).\end{aligned}$$

From the monotonicity of cosine in $[0,2\pi/3]$, we see that the above derivative preserves non negative sign when $\alpha \in [0,2\pi/3]$. Hence, the maximum of

$$3\alpha/2+2\sin\left(\alpha\right)-2\sin\left(\alpha/2\right) \leq \pi$$

is attained when $\alpha = 2\pi/3$, and its value is π as wanted.

Proof of A.1f We have that

$$\begin{aligned}&\max_{0\leq x\leq \pi-\alpha}\left\{\sin\left(\pi/2-\alpha/2-x\right)\right\}\\ =&\max_{0\leq x\leq \pi-\alpha}\left\{\cos\left(\alpha/2+x\right)\right\}\\ \leq&\cos\left(\alpha/2\right),\end{aligned}$$

since cosine is monotonically decreasing in $[0,\pi]$. But also for all $\alpha \in [2\pi/3,\pi]$ we have that $\cos\left(\alpha/2\right) \leq \sin\left(\alpha/2\right)$, concluding what we need.

Proof of A.1g We have that

$$\begin{aligned} &\max_{0\leq x\leq \alpha/2}\{x+2\sin\left(\alpha/2-x\right)+2\sin\left(\alpha\right)\} \\ \leq&\alpha/2+\sin\left(\alpha/2\right)+\sin\left(\alpha\right) \end{aligned}$$

where the first inequality is true due to the monotonicity of $x, \sin\left(\alpha/2-x\right)$ w.r.t. $x \leq \alpha/2$ and for all $\alpha \in [0, 2\pi/3]$, and the last inequality since again $\alpha \leq 2\pi/3$. The claim now follows from Lemma A.1e.

Proof of A.1h Follows immediately since $x \leq \pi - \alpha$, and by Lemma A.1f.

Proof of A.1i Observe that $\sin\left(\alpha/2\right) - \sin\left(\alpha\right)$ is convex in $\alpha \in [0, 2\pi/3]$, so it attains its maximum either at $\alpha = 0$ or at $\alpha = 2\pi/3$. In both cases, its value is 0. Also, $\sin\left(\alpha/2\right) - \sin\left(\alpha\right)$ is monotonically increasing for $\alpha > 2\pi/3$, implying what was promised.