

Search-and-Fetch with 2 Robots on a Disk

Wireless and Face-to-Face Communication Models

Konstantinos Georgiou^{1,*}, George Karakostas^{2,*} and Evangelos Kranakis^{3,*}

¹Department of Mathematics, Ryerson University, 350 Victoria St., Toronto, ON, M5B 2K3, Canada

²Dept. of Computing & Software, McMaster University, 1280 Main St. West, Hamilton, Ontario L8S 4K1, Canada

³School of Computer Science, Carleton University, 1125 Col. By Dr., Ottawa, ON K1S 5B6, Canada

Keywords: Search & Fetch, Distributed Algorithm, Rendezvous, Exploration, Face-to-Face Model, Wireless Model.

Abstract: We introduce and study *treasure-evacuation* with 2 robots, a new problem on distributed *searching and fetching* related to well studied problems in searching, rendezvous and exploration. The problem is motivated by real-life search-and-rescue operations in areas of a disaster, where unmanned vehicles (robots) search for a victim (treasure) and subsequently bring (fetch) her to safety (the exit). One of the critical components in such operations is the communication protocol between the robots. We provide search algorithms and contrast two standard models, the face-to-face and the wireless model. Our main technical contribution pertains to the face-to-face model. More specifically, we demonstrate how robots can take advantage of some minimal information of the topology (i.e., the disk) in order to allow for information exchange without meeting. The result is a highly efficient distributed treasure-evacuation protocol which is minimally affected by the lack of distant communication.

1 INTRODUCTION

We introduce the study of a new 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 an interesting point (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, rather they only need to co-operate, possibly by sharing information, so as to learn the locations of the interesting points 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 (first the treasure, then the exit). This makes the problem inherently different in nature and more difficult than similarly looking results.

Finding an optimal algorithm turns out to be a challenging task even when the robots have some knowledge, e.g., the arc-distance α between the exit

and the treasure. 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 this knowledge 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.

Part of our contribution is that we demonstrate how robots can utilize the knowledge of the arc-distance α between the interesting points. We propose protocols that induce worst case evacuation time $1 + \pi - \alpha/2 + 3 \sin(\alpha/2)$ for the wireless model and $1 + \pi - \alpha + 4 \sin(\alpha/2)$ for the face-to-face model. 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. 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 $\alpha \in [2\pi/3, \pi]$.

*Research supported in part by NSERC.

1.1 Contributions

Rendezvous, treasure hunting and exploration have been subjects of extensive research in the broad area of distributed and online computing (see related work below). Challenges in each of these fundamental tasks arise from different computation, communication or knowledge limitations, with admittedly numerous variations. The novel distributed problem of treasure-evacuation that we introduce and study in this work combines in a complex way challenges from all these fundamental tasks. As such, progress towards solving generic treasure-evacuation-like problems will unavoidably touch on state-of-the-art techniques of achieving these tasks.

In treasure-evacuation, two stationary targets (a treasure and an exit that we call *interesting points*) are hidden on a specific domain. At first, robots need to (a) perform treasure hunting in this online environment. Interestingly, the knowledge of the location of one of these targets may or may not reveal the location of the other. In particular, the task of the robots does not end when both interesting points are located, rather, only when the treasure-holder learns (and finds) the location of the exit as well. Given communication limitations, the latter can be accomplished *efficiently* only if (b) a sufficient portion of the environment is explored before both interesting points are found, or (c) if information can be exchanged between robots. In case (b), robots “invest” in domain exploration in an attempt to expedite treasure evacuation once the interesting points are found. In case (c), robots may need to attempt to rendezvous (not very late in the time horizon) so that the distributed system becomes cognitive of the environment and consequently completes the given task. Clearly, in order for a distributed system to accomplish treasure evacuation efficiently, robots need to perform and balance all tasks (a),(b),(c) above, i.e to perform treasure hunting, while learning the environment either by exploration or by rendezvous. A unique feature of treasure-evacuation is that only the treasure has to be brought to the exit.

The last observation gives some first evidence of the difficulty of solving treasure-evacuation, even when the domain is a disk. Can a robot choose a trajectory (maybe staying far from the exit) to help the treasure-holder expedite evacuation? If a robot discovers the treasure, is it a good strategy to become the treasure holder and greedily search for the exit? Should robots learn the environment by investing on exploration or on rendezvous and hence on message exchange? Finally, is it possible in a non-wireless environment for robots to exchange information without meeting? An efficient algorithm should somehow ad-

dress all these questions.

From the discussion above, it is not a surprise that plain vanilla algorithms cannot be efficient. Indeed, our algorithms (for both the wireless and face-to-face models) adapt their strategies, among others, with respect to the distance of the interesting points. On one hand, there are configurations where the evacuation protocols are simple and greedy-like. However, the reader can verify from our analyses that, had we followed such simplistic approach for all configurations, the evacuation time would have been much worse than our upper bounds. The simplest example of this kind will be transparent even in the analysis of the wireless model, in which robots can exchange information at will. To achieve our upper bound, robots choose different trajectories (still greedy-like) for various distances of the interesting points. Nevertheless, the analysis in this case is relatively simple.

Our main technical contribution pertains to the face-to-face model where robots can exchange information only if they meet. In particular, we explicitly exhibit distributed strategies that allow robots to exchange information even from distance. At a high level, and given that interesting points are located by robots at some (carefully defined) critical intervals in the time horizon, robots choose (occasionally) highly non-intuitive trajectories, not in order to locate the remaining interesting points, rather to potentially meet their fellow robots. The trajectories are carefully chosen so that a robot may deduce information, using an involved protocol, as to what the other robot has found, and hence learn the environment regardless of whether the rendezvous is realized or not. For the treasure-holder, this would result in learning the location of the exit. For the other robot(s), this would be an altruistic attempt to help the fellow treasure-holder. In particular, that could result in that the non treasure-holder never finds the exit, still expediting the treasure evacuation time. We note that once the trajectories are determined (which is the heart of our contribution) correctness and performance analysis is a matter of an exhaustive and technical case analysis. Interestingly, the efficiency of our algorithm for the face-to-face model is only slightly worse than the solution for the wireless case (but significantly better than the naive solution for the face-to-face model), indicating that lack of communication can be compensated by clever algorithms.

Finally, we complement our results by proving some lower bounds for treasure evacuation with 2 robots for the face-to-face model. That concludes the first attempt to study distributed problems of this kind, i.e. optimization treasure hunting problems where the distributed systems learn the online environment by a

combination of exploration and rendezvous, a feature which, to the best of our knowledge, is also novel.

1.2 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 (Ahlsweide and Wegener, 1987; Alpern and Gal, 2003; Stone, 1975)). 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 (Berman, 1998).

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 (Stone, 1975), game theoretic applications (Alpern and Gal, 2003), cops and robbers (Bonato and Nowakowski, 2011), classical pursuit and evasion (Nahin, 2012), search problems as related to group testing (Ahlsweide and Wegener, 1987), searching a graph (Koutsoupias et al., 1996), and many more. A survey of related search and pursuit evasion problems can be found in (Chung et al., 2011). 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, (Gluss, 1961; Isbell, 1967), the cow-path problem (Beck, 1964; Bellman, 1963), and with the plane searching problem (Baeza-Yates and Schott, 1995). 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 (Albers and Henzinger, 2000; Albers et al., 2002; Deng et al., 1991; Hoffmann et al., 2001). Coordinating the exploration of a team of robots is a main theme in the robotics community (Burgard et al., 2005; Thrun,

2001; Yamauchi, 1998) and often this is combined with the mapping of the terrain and the position of the robots within it (Kleinberg, 1994; Papadimitriou and Yannakakis, 1989).

Evacuation for grid polygons has been studied in (Fekete et al., 2010) 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 (Baeza Yates et al., 1993) and the recent (Chrobak et al., 2015)) 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 (Demaine et al., 2006) (on searching with turn costs), (Kao et al., 1996) (randomized algorithm for the cow-path problem), (Kao et al., 1998) (hybrid algorithms), and many more.

A setting similar to ours is presented in the recent works (Czyzowicz et al., 2014; Czyzowicz et al., 2016; Czyzowicz et al., 2015a; Czyzowicz et al., 2015b) 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 (Czyzowicz et al., 2014; Czyzowicz et al., 2016; Czyzowicz et al., 2015a) is a unit circle (while in (Czyzowicz et al., 2015b) 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.

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 (Jennings et al., 1997). A scenario similar to ours (but only for 1 robot) has been introduced by Alpern in (Alpern, 2011), where the domain was discrete (a tree) and the approach/analysis resembled that of standard search-type problems (Alpern and Gal, 2003). In contrast, our problem is of distributed nature, and our focus is to demonstrate how robots’ communication affects efficiency.

1.3 Problem and Model Motivation

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 vic-

tim (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.

From a technical perspective, our communication models are inspired by the recent works on evacuation problems (Czyzowicz et al., 2014; Czyzowicz et al., 2015a; Czyzowicz et al., 2015b). 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 interesting points. 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 (Georgiou et al., 2016b). In particular, the work of (Georgiou et al., 2016b) implies that establishing provably optimal evacuation protocols for 2-robots is a difficult task, even when the domain is the disk. Nevertheless, we view the 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 interesting points to be located, this kind of information suggests that the points lie anywhere on concentric 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.

In order to demonstrate that robots with primitive communication capabilities are in fact not much less powerful than in the wireless model, it is essential to assume that robots have some knowledge of the distance between the interesting points. The reader may also view this piece of advice as an algorithmic challenge in order to bypass the uncertainty regarding the locations of the interesting points. Notably, our algorithms adapt strategies as a function of the distance of the interesting points, 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 an interesting point reveals the location of the other α -arc distant away interesting point (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. Apart from that, partial knowledge of the input is also interesting due to the efficiency-information tradeoffs that are naturally induced by the problem, which is also a standard theme in competitive analysis, e.g. see (Hipke et al., 1999) and (Georgiou et al., 2016b).

Admittedly, the model we introduce is simple but natural, general, and complex enough to require non-standard algorithmic solutions. Most importantly, our model allows us to demonstrate in a relatively clean way a couple of novel algorithmic techniques for attacking challenging and newly introduced types of distributed problems. We anticipate that the ideas introduced in this work will initiate new research directions towards solving a family of problems that are not yet understood from a theoretical perspective.

1.4 Notation & Organization

A treasure and an exit are located at unknown positions on the perimeter of a unit-disk and at arc dis-

tance α (in what follows all distances will be arc-distances, unless specified otherwise). Robots 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 *interesting* points. For an interesting point 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 $B = \text{null}$ to denote the event that neither the treasure nor the exit is placed on B .

We focus on the following online variations 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. We give two algorithms: in the former case we prove a $1 + \pi - \alpha + 4 \sin(\alpha/2)$ and in the latter case a $1 + \pi - \alpha/2 + 3 \sin(\alpha/2)$ upper bound, resp., on the treasure evacuation time, where α is the arc distance between treasure and exit. Finally in Section 4 we provide a lower bound for treasure-evacuation with 2 robots in the F2F model.

Any omitted proofs, due to space limitations, can be found in the full version of the paper (Georgiou et al., 2016a).

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 al-

gorithm follows a greedy like approach, adapting its strategy as a function of the distance α of the interesting points. 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 an interesting point 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 1.

Noticeably, the performance analysis we give is tight, meaning that for every $\alpha \geq 0$, there are configurations (placements of the interesting points) for which the performance of the algorithm is exactly $1 + \pi - \alpha + 4 \sin(\alpha/2)$. Most importantly, the performance analysis makes explicit that 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 an interesting point $I = \{E, T\}$, say at time $x := \widehat{AI}$. We call B, C the points that are at clockwise and counter-clockwise arc-distance α from I respectively. Figure 1 depicts the interesting points encountered.

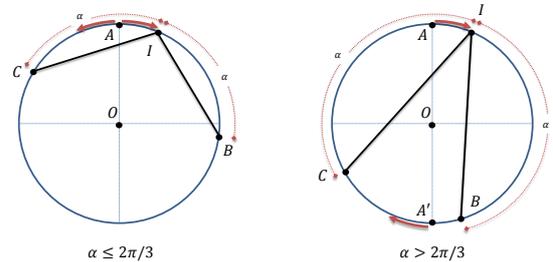


Figure 1: The points of interest for our Algorithm 1.

The description of Algorithm 1 is from the perspective of the robot that finds first an interesting point, that we always assume is R_2 . Next we assume that the finding of any interesting point is instanta-

neously transmitted and received by the two robots. Also, if at any moment, the positions of the interesting points 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.

Algorithm 1: Wireless Algorithm.

- Step 1.** If $\alpha \leq 2\pi/3$, then the two robots move together to an arbitrary point on the ring and start moving 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 interesting point discovered by R_2 , at time $x := \widehat{AI}$. Let B, C be the points that are at clockwise and counter-clockwise arc-distance α from I respectively.
- Step 3.** If $x \geq \alpha/2$ then robots learn that the other interesting point is in B , else R_2 moves to B , R_1 moves to C .
- Step 4.** Evacuate
-

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 is exactly as promised by Theorem 2.1. Moreover, we may assume that $\alpha > 0$ as otherwise the problem is solved when one interesting point is found.

Note that our algorithm performs differently when $\alpha \leq 2\pi/3$ and when $\alpha > 2\pi/3$. Let $x := \widehat{AI}$ be the time that R_2 first discovers interesting point I . Then it must be that $x \leq \alpha/2$ and $x \leq \pi - \alpha$ for the cases $\alpha \leq 2\pi/3$ and $\alpha > 2\pi/3$ respectively (see also Figure 1). 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 an interesting point.

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 an

interesting point first) and the two interesting points.

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

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

It is clear now that Lemmata 2.2, 2.3 imply 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 F2F 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_{ff} 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 promised by the theorem above. It should be intuitive that when the distance of the interesting points 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 interesting points and fetch the treasure to the exit without coordination (which is hindered anyways due to lack of communication). Following a worst case analysis, it is easy to see that as long as α does not exceed a special threshold, call it α_0 , 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 interesting points are not located in certain parts of the disk, potentially revealing their actual location. In fact, this recipe 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 interesting points are further than $2\pi/3$ apart. Intuitively, in such a case there is always uncertainty as to where the interesting points are located, even when one of them is discovered. At the same time, the interesting points, hence the robots, might be already far apart when some or both interesting points 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 interesting point.

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 $\widehat{DC} = \widehat{CI} = \widehat{IB} = \alpha$ (see Figure 2). 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.

According to our algorithm, R_2 starts moving from point A till it reaches an interesting point I at

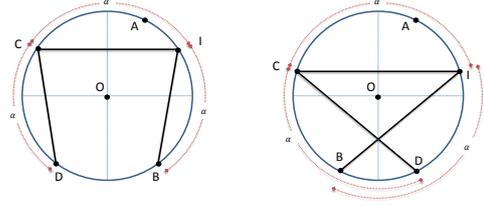


Figure 2: The points of interest from the perspective of R_2 , when $\alpha \leq 2\pi/3$ on the left, and when $\alpha \geq 2\pi/3$ on the right.

time $x := \widehat{AI}$. 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’’.

$\mathcal{A}_1(\mathbf{x})$ (Figure 3 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 3 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 $B = \text{null}$, 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).

$\mathcal{A}_2(\mathbf{x})$ (Figure 3 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 3 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 $B = \text{null}$, then go to the centre O and halt.

$\mathcal{A}_3(\mathbf{x})$ (Figure 3 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 $\widehat{IK} = y$. 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 3 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 $B = \text{null}$, 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 .

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 interesting points 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 $\bar{x}(\alpha) := 3\alpha/2 - \pi - \sin(\alpha/2) + 2\sin(\alpha)$, which we abbreviate by \bar{x} whenever α is clear from the context. By Lemma Aa, $\alpha_0 \approx 1.22353$ is the unique root of $\bar{x}(\alpha) = 0$, while \bar{x} is positive for all $\alpha \in (\alpha_0, \pi)$ and negative for all $\alpha \in [0, \alpha_0)$.

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

Algorithm 2: Non-Wireless Algorithm.

Step 1. Starting from A , move clockwise until an interesting point I is found at time $x := \widehat{AI}$.

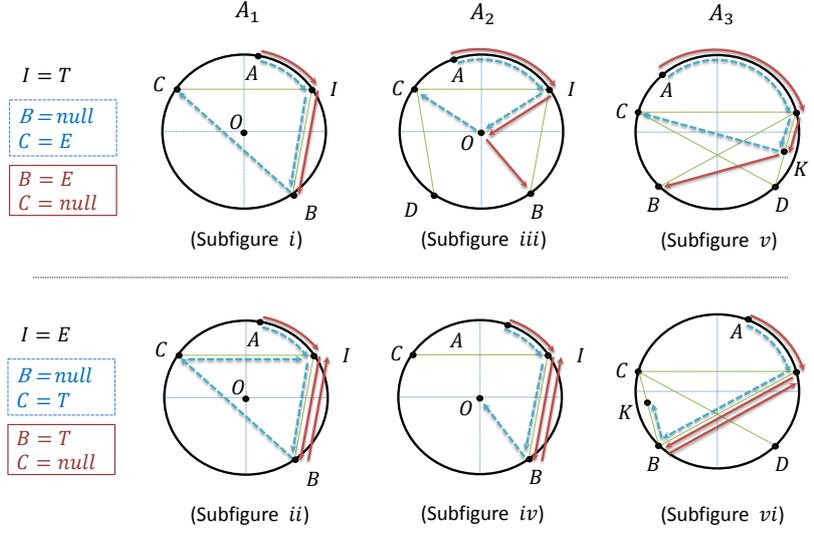
Step 2. Proceed according to the following cases:

- If $\alpha > 2\pi/3$ and $I = T$ and $\alpha > x \geq \alpha - \bar{x}$, then run $\mathcal{A}_3(x)$.
 - If $\alpha > 2\pi/3$ and $I = E$ and $x \leq \bar{x}$, then run $\mathcal{A}_3(x)$.
 - If $\alpha_0 \leq \alpha \leq 2\pi/3$ and $I = T$ and $\alpha > x \geq \alpha - \bar{x}$, then run $\mathcal{A}_2(x)$.
 - If $\alpha_0 \leq \alpha \leq 2\pi/3$ and $I = E$ and $x \leq \bar{x}$, then run $\mathcal{A}_2(x)$.
 - In all other cases, run $\mathcal{A}_1(x)$.
-

Proof. First, it is easy to see that the treasure is always picked up. Indeed, if the first interesting point I that is discovered (by any robot) is the treasure, then the claim is trivially true. If the first interesting point 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 3).

If R_2 is following subroutine \mathcal{A}_2 , then it must be that $\alpha_0 \leq \alpha \leq 2\pi/3$, that $\alpha - \bar{x} \leq x < \alpha$, and that it has not found any other interesting point before (by Lemma Aa we have $\alpha - \bar{x} < \alpha$ and $\bar{x} > 0$ for all $\alpha > \alpha_0$). Figure 3 subcase iii depicts exactly 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 $\alpha - x \leq \bar{x}$ 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 3 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 .


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

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 $\alpha - \bar{x} \leq x < \alpha$, and that it has not found any other interesting point before. Figure 3 subcase v depicts this scenario. Note that the exit could be either in C or in D .

If the exit is in C , then $\alpha - x \leq \bar{x}$, 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 3 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 in 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 $\widehat{AI} + \overline{IK} = \widehat{CA} + \widehat{CD} + (\widehat{DI} - \overline{IK})$, 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 $x = \alpha - \bar{x}$, 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 in 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 interesting point at time $x \geq \alpha$ after moving on the perimeter of the disk, then that robot can also deduce where the other interesting point 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 an interesting point $I \in \{E, T\}$ 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 an interesting point $I \in \{E, T\}$. Let also B be the other interesting point $\{E, T\} \setminus I$. 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 2 as reference (although A is depicted in the interior of the arc CI , we will not use that $\widehat{AI} \leq \alpha$).

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

$$\min \{x + 2 \sin(\alpha/2), 2\pi - \alpha - x\} + 2 \sin(\alpha/2) \leq \pi - \alpha/2 + 3 \sin(\alpha/2) \quad (\text{by Lemma Ab})$$

It is easy to see 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 an interesting point, arrives at I at time $x := \widehat{AI} < \alpha$. A reference for the analysis below is Figure 3 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 both lemmata below, the reader may consult Figures 2 and 3.

Lemma 3.4. *Let x be the time some robot is the first to reach an interesting point $I \in \{E, T\}$ 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 .

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

If R_2 runs \mathcal{A}_2 , then it must be that $\alpha - \bar{x} \leq x < \alpha$ and $\alpha \leq 2\pi/3$, and the robot goes to the centre in order to learn where the exit is (see Figure 3 iii). Independently of where the exit is, and by Lemma 3.2, R_2 makes the right decision and evacuates in time $1 + \max_{\alpha - \bar{x} \leq x < \alpha} \{x, \alpha - x + 2 \sin(\alpha/2)\} + 1 \leq \max\{\alpha, \bar{x} + 2 \sin(\alpha/2)\} + 2$ which, by Lemma Ac, 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 interesting point 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 $\alpha - \bar{x} \leq x < \alpha$ and $\alpha > 2\pi/3$. Then the trajectory of R_2 is as in Figure 3 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 $\widehat{AI} + \overline{IK} + \max\{\overline{KB}, \overline{KC}\}$, where $\overline{IK} = y$ (see definition of \mathcal{A}_3). Since as we have proved, K lies always in chord ID , and since $\widehat{DB} = 3\alpha - 2\pi$ 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

$$\begin{aligned} x + y + 2 \sin(\alpha/2) &= \alpha/2 + 3 \sin(\alpha/2) + \sin(\alpha) \\ &\leq \pi - \alpha/2 + 3 \sin(\alpha/2) \quad (\text{by Lemma Ad}) \end{aligned}$$

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

($I = T, B = \text{null}, C = E$): Since I is found first, we must have $x \leq \alpha/2$, hence both robots run \mathcal{A}_1 , see Figure 3 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 $\alpha \in [0, \pi]$.

($I = E, B = T, C = \text{null}$): If R_2 is the first to find the treasure, then this case is depicted in Figure 3 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 $I = T, B = E, C = \text{null}$ 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 $x \geq \bar{x}$, while the cost is always $\alpha - x + 2 \sin(\alpha/2) + 2 \sin(\alpha)$ (the case is depicted in Figure 3 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 Ah is at most $\pi - \alpha/2 + 3 \sin(\alpha/2)$, for all $\alpha \in [0, \alpha_0]$. If $x \geq \bar{x}$, then the cost would be at most $\alpha - \bar{x} + 2 \sin(\alpha/2) + 2 \sin(\alpha) = \pi - \alpha/2 + 3 \sin(\alpha/2)$. ■

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.

3.3 Extension to n Robots

We can easily extend our 2-robot algorithms to the n -robot case (when n is even, otherwise we ignore one robot) by splitting the robots into pairs, defining points in intervals of length $4\pi/n$ on the cycle, assigning each pair of robots to each such point, and letting them run the corresponding 2-robot algorithm.

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 $0 \leq \alpha < 2\pi/3$, 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 interesting points.

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. These include problems on 1) the number of robots, 2) other geometric domains (discrete or continuous), 3) differing robot starting positions, 4) multiple treasures and exits, 5) limited range wireless communication, 6) robots with different speeds, 6) different a priori knowledge of the topology or partial information about the targets, etc. In particular, 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

- Ahlsweide, R. and Wegener, I. (1987). *Search problems*. Wiley-Interscience.
- Albers, S. and Henzinger, M. R. (2000). Exploring unknown environments. *SIAM Journal on Computing*, 29(4):1164–1188.
- Albers, S., Kursawe, K., and Schuierer, S. (2002). Exploring unknown environments with obstacles. *Algorithmica*, 32(1):123–143.
- Alpern, S. (2011). Find-and-fetch search on a tree. *Operations Research*, 59(5):1258–1268.
- Alpern, S. and Gal, S. (2003). *The theory of search games and rendezvous*. Springer.
- Baeza Yates, R., Culberson, J., and Rawlins, G. (1993). Searching in the plane. *Information and Computation*, 106(2):234–252.
- Baeza-Yates, R. and Schott, R. (1995). Parallel searching in the plane. *Computational Geometry*, 5(3):143–154.
- Beck, A. (1964). On the linear search problem. *Israel Journal of Mathematics*, 2(4):221–228.
- Bellman, R. (1963). An optimal search. *SIAM Review*, 5(3):274–274.
- Berman, P. (1998). On-line searching and navigation. In Fiat, A. and Woeginger, G. J., editors, *Online Algorithms: The State of the Art*, pages 232–241. Springer.
- Bonato, A. and Nowakowski, R. (2011). *The game of cops and robbers on graphs*. AMS.
- Burgard, W., Moors, M., Stachniss, C., and Schneider, F. E. (2005). Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21(3):376–386.

Chrobak, M., Gasieniec, L., T., G., and Martin, R. (2015). Group search on the line. In *SOFSEM 2015*. Springer.

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

Czyzowicz, J., Gasieniec, L., Gorry, T., Kranakis, E., Martin, R., and Pajak, D. (2014). Evacuating robots from an unknown exit located on the perimeter of a disc. In *DISC 2014*, pages 122–136. Springer, Austin, Texas.

Czyzowicz, J., Georgiou, K., Dobrev, S., Kranakis, E., and MacQuarrie, F. (2016). Evacuating two robots from multiple unknown exits in a circle. In *ICDCN 2016*.

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

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

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

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

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

Georgiou, K., Karakostas, G., and Kranakis, E. (2016a). Search-and-fetch with 2 robots on a disk: Wireless and face-to-face communication models. *CoRR*, abs/1611.10208.

Georgiou, K., Karakostas, G., and Kranakis, E. (2016b). Search-and-fetch with one robot on a disk. In *12th International Symposium on Algorithms and Experiments for Wireless Sensor Networks*.

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

Hipke, C., Icking, C., Klein, R., and Langetepe, E. (1999). How to find a point on a line within a fixed distance. *Discrete Appl. Math.*, 93(1):67–73.

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

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

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

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

Kao, M.-Y., Reif, J. H., and Tate, S. R. (1996). Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. *Information and Computation*, 131(1):63–79.

Kleinberg, J. (1994). On-line search in a simple polygon. In *SODA*, page 8. SIAM.

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

Nahin, P. (2012). *Chases and Escapes: The Mathematics of Pursuit and Evasion*. Princeton University Press.

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

Stone, L. (1975). *Theory of optimal search*. Academic Press New York.

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

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

APPENDIX

- Lemma A.** 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) $\max\{\alpha, \bar{x} + 2\sin(\alpha/2)\} + 2 \leq \pi - \alpha/2 + 3\sin(\alpha/2)$ for all $\alpha \in [0, 2\pi/3]$.
- d) $\alpha + \sin(\alpha) \leq \pi$ for all $\alpha \in [0, \pi]$.
- e) $\alpha - \sin(\alpha/2) + 2\sin(\alpha) \leq \pi$ for all $\alpha \in [0, \pi]$.
- f) $\alpha/2 + 2\sin(\alpha) \leq \pi - \alpha + 2\sin(\alpha/2), \forall \alpha \in [0, 2\pi/3]$.
- g) $\max_{0 \leq x \leq \pi - \alpha} \{\sin(\pi/2 - \alpha/2 - x)\} \leq \sin(\alpha/2), \forall \alpha \in [2\pi/3, \pi]$
- h) $\max_{0 \leq x \leq \alpha/2} \{x + 2\sin(\alpha/2 - x)\} + 2\sin(\alpha) \leq \pi - \alpha + 4\sin(\alpha/2), \forall \alpha \in [0, 2\pi/3]$.
- i) $\max_{0 \leq x \leq \pi - \alpha} \{x + 2\sin(\pi/2 - \alpha/2 - x)\} \leq \pi - \alpha + 2\sin(\alpha/2), \forall \alpha \in [2\pi/3, \pi]$
- j) $\sin(\alpha) \leq \sin(\alpha/2), \forall \alpha \in [0, 2\pi/3]$, and $\sin(\alpha) \geq \sin(\alpha/2), \forall \alpha \in [2\pi/3, \pi]$.