Fast Two-Robot Disk Evacuation with Wireless Communication*

Ioannis Lamprou(✉) Russell Martin Sven Schewe

Abstract

In the fast evacuation problem, we study the path planning problem for two robots who want to minimize the worst-case evacuation time on the unit disk. The robots are initially placed at the center of the disk. In order to evacuate, they need to reach an unknown point, the exit, on the boundary of the disk. Once one of the robots finds the exit, it will instantaneously (using wireless communication) notify the other agent, who will make a beeline to it.

The problem has been studied for robots with the same speed [8]. We study a more general case where one robot has speed 1 and the other has speed $s \geq 1$. We provide optimal evacuation strategies in the case that $s \geq c_{2.75} \approx 2.75$ by showing matching upper and lower bounds on the worst-case evacuation time. For $1 \leq s < c_{2.75}$, we show (non-matching) upper and lower bounds on the evacuation time with a ratio less than 1.22. Moreover, we demonstrate that a different-speeds generalization of the two-robot search strategy from [8] is outperformed by our proposed strategies for any $s \geq c_{1.71} \approx 1.71$.

1 Introduction

Consider a pair of mobile robots in an environment represented by a circular disk of unit radius. The goal of the robots is to find an *exit*, i.e. a point at an unknown location on the boundary of the disk, and both move to this exit. The exit is only recognized when a robot visits it. The robots' aim is to accomplish this task as quickly as possible. This problem is referred to as the *evacuation problem*. The robots start at the center of the disk and can move with a speed not exceeding their maximum velocity (which may be different from one another). They can coordinate their actions in any manner they like, and can communicate wirelessly (instantaneously).

1.1 Related work

Evacuation belongs to the realm of distributed search problems, which have a long history in mathematics, computer science, and operations research, see, e.g. [3].

*The final publication is available at `http://link.springer.com/chapter/10.1007%2F978-3-662-53426-7_1`. Published version appears in Volume 9888 of the book series *Lecture Notes in Computer Science*.

Salient features in search problems include the *environment* (e.g. a geometric one or graph-based), *mobility* of the robots (how they are allowed to move), *perception* of and *interaction* with the environment, and their *computational* and *communication abilities*. Typical tasks include exploring and mapping an unknown environment, finding a (mobile or immobile) target (e.g. cops and robbers games [4] and pursuit-evasion games [17]; the "lost at sea" problem [12]; the cow-path problem and plane-searching problem [2],[5],[14],[15]), rendezvous or gathering of mobile agents [16], and evacuation [7, 8],[10]. (Note that we distinguish between the distributed version of evacuation problems involving a search for an unknown exit, and centralized versions, typically modeled as (dynamic) capacitated flow problems on graphs, where the exit is known.) A general survey of search and rendezvous problems can be found in [1]. Also related is the task of patrolling or monitoring, i.e. the periodic (re)visitation of (part of) the environment [6],[9],[18].

In most of these settings, the typical cost is the time required to finish the task (in a synchronous environment), or the total distance moved by the robots to finish it (in an asynchronous setting). (Patrolling has a different "cost", the time between consecutive visits to any point in the region, the so-called "idle time".)

A little explored feature of the robots is their *speed*. Most past work has focused on the case where all robots share the same (maximal) speed. Notable exceptions of which the authors are aware include [7], which considers the evacuation problem on the infinite line with robots with distinct maximal speeds, [9], which introduces a non-intuitive ring patrolling strategy using three robots with distinct maximal speeds, and [11],[13], where the rendezvous problem with different speeds in a cycle is studied. It is this feature, robots with different maximal speeds, that we explore in this paper.

The most relevant previous work is [8],[10], which explores the evacuation problem in the unit disk with two robots with identical speeds ($s = 1$).

1.2 Our results

We consider the evacuation problem in the unit disk using two robots with distinct maximal speeds (one with speed 1, the second with speed $s \geq 1$). The robots share a common clock and can communicate instantaneously when they have found the exit (wireless communication) and so can synchronize their behavior in the evacuation procedure. We assume that the robots can measure distances to an arbitrary precision (equivalently, they can measure time to an arbitrary precision), and can vary their speeds as they desire, up to their maximum speed.

We show that, even in the case of two robots, the analysis involved in finding (time) optimal evacuation strategies can become intricate, with strategies that depend on the ratio of the fast robot's to the slow robot's maximal speed.

For large s, we introduce an efficient and non-obvious search strategy, called the Half-Chord Strategy (Figure 1). We generalize a strategy from [8] for small s, the "Both-to-the-Same-Point Strategy" (BSP), where the two robots move to the same point on the boundary and then separately explore the boundary in clockwise and counterclockwise directions to find the exit. For values of $s \geq c_{1.86}$ (with $c_{1.86} \approx 1.856$), we show that BSP is not optimal by demonstrating that the Half-Chord Strategy is superior to it. Moreover, we improve on this with the Fast-Chord Strategy (Figure 4), which outperforms Half-Chord for $1.71 \approx c_{1.71} < s < c_{2.07} \approx 2.07$. We obtain optimality for all $s \geq c_{2.75} \approx 2.75$, in the wireless setting, as we demonstrate matching upper and lower bounds on the evacuation time. For $s \in (1, c_{2.75})$, we provide lower bounds on the evacuation time that do not match the bounds provided by the respective search strategies (BSP for $s < c_{1.71}$, Fast-Chord for $s \in [c_{1.71}, c_{2.07})$, and Half-Chord for $s \geq c_{2.07}$). The worst ratio between our upper and lower bound, 1.22, is realized for $s = c_{1.71}$.

Section 2 contains a more formal definition of the problem we consider. Section 3 contains our upper bounds on the evacuation time, while Section 4 has our lower bounds. In the interests of space, parts of the proofs are omitted from this version, and we trust the reader to rely upon the supplied diagrams for the intuition of our results.

2 Problem Definition and Strategy Space

In this section, we formally define the problem in question. Furthermore, we provide a partition of the strategy space and some observations, which will be useful in the bounds to follow.

Definition 1 (The Fast Evacuation Problem). *Given a unit disk and two robots starting at its center (the former with maximum speed $s \geq 1$ and the latter with maximum speed 1), provide an algorithm such that* both *robots reach an unknown exit lying on a boundary point of the disk. The two robots, called* Fast *and* Slow*, are allowed to move within the entire unit disk, can only identify the exit when they stand on it, and can communicate wirelessly at any time.*

Definition 2. *An "evacuation strategy" is an algorithm on how each robot moves such that both robots have evacuated the disk at the end of its execution.*

The following remark is a direct consequence of the geometric environment in which this fast evacuation scenario takes place.

Remark 1. *In any evacuation strategy, when either robot discovers the exit, the optimal strategy of the other one immediately reduces to following a beeline to the exit.*

We now proceed with identifying key aspects of potential strategies.

Definition 3. *A "both-explore" strategy is a strategy for both robots to evacuate the disk, where (in the worst-case) both of them explore at least two distinct points on the boundary. We define the set of all both-explore strategies as BES.*

Definition 4. *A "fast-explores" strategy is a strategy where* only Fast *explores the boundary searching for the exit. Slow, eventually, only reaches the exit point and at any time it reaches no other point on the boundary of the disk. We define the set of all fast-explores strategies as FES.*

Definition 5. *A "slow-explores" strategy is a strategy where* only Slow *explores the boundary searching for the exit. Fast, eventually, only reaches the exit point and at any time it reaches no other point on the boundary of the disk. We define the set of all slow-explores strategies as SES.*

Notice that, for $s = 1$, if only one robot explores the boundary, we randomly assign such a strategy to FES or SES. Below, let ALL stand for the set of all evacuating strategies.

Proposition 2. *(BES, FES, SES) forms a partition of ALL.*

We remark that, when considering SES and FES strategies, it can become a burden to forcefully keep the non-exploring robot away from the boundary. E.g., if we only want Slow to explore in an SES strategy, the optimal behavior of Fast would be to mimic the behavior of Slow. For FES strategies with $s \leq 2$, it also proves to be most natural to allow Slow to move on the boundary, but to ignore it when Slow finds the exit first. For this reason we use FES and SES strategies in this sense. Alternatively, one could also let the non-exploring robot to move ε-close to the boundary.

We do not consider SES strategies in our analysis. An optimal SES strategy is obviously to go to the boundary and explore the boundary (counter)clockwise. The worst case time is $1 + 2\pi$.

3 Upper Bounds

3.1 The Half-Chord Strategy

The idea for this strategy stems from the proof of the FES lower bound to follow. The worst-case analysis is performed for $s \in [2, \infty)$. For the strategy details below, please refer to Figure 1. The center of the disk is denoted by O. Fast's trajectory is given with *double arrows*, while Slow's is given with *single arrows*. All angles and arcs are considered in counterclockwise order.

3.1.1 The Strategy.

Initially, both robots move in beelines with an angle of $\pi + 1/2$ between them until *Fast* reaches the boundary (i.e. for $\frac{1}{s}$ time). Let B be the first boundary point reached by *Fast*. From now on, *Fast's* strategy reduces to exploring the boundary. On the other hand, *Slow* continues on its beeline for another $\frac{1}{s}$ time until it reaches point C (where $|OC| = \frac{2}{s}$). Then, it takes an arc from C to M on the disk with radius $\frac{2}{s}$ centered at O (where M is the middle point of chord BA, where A is the point with arc distance $2\arccos\left(-\frac{2}{s}\right)$ from B). Finally, *Slow* traverses MB. Below, we provide a more structured and formal strategy definition.

Fast moves as follows until the exit is found:

- for $t \in \left[0, \frac{1}{s}\right]$: moves toward B and
- for $t \in \left(\frac{1}{s}, \frac{1+2\pi}{s}\right]$: traverses the boundary counterclockwise.

Slow moves as follows until the exit is found:

- Phase I: for $t \in \left[0, \frac{2}{s}\right]$ moves toward C,
- Phase II: for $t \in \left[\frac{2}{s}, \frac{1+2\arccos(-2/s)}{s}\right]$ moves toward M via $\widehat{CM}$ on disk $\left(O, \frac{2}{s}\right)$,

Table 1: Measurements for Half-Chord Strategy

$\lvert OC\rvert = \frac{2}{s}$	by choice
$\widehat{BA} = 2\arccos\left(-\frac{2}{s}\right)$	by choice
$\phi = \measuredangle BOC = \pi + 1/2$	by choice
$\lvert\widehat{CM}\rvert = \frac{1}{s}(2\arccos\left(-\frac{2}{s}\right) - 1)$	slow on M exactly when fast on A
$\theta = \measuredangle COM = \frac{s}{2}\lvert\widehat{CM}\rvert = \arccos\left(-\frac{2}{s}\right) - 1/2$	arc-to-angle
$\psi = \measuredangle MOB = 2\pi - \phi - \theta = \pi - \arccos\left(-\frac{2}{s}\right)$	sum of angles around O
$\lvert AB\rvert = 2\sin\left(2\arccos\left(-\frac{2}{s}\right)/2\right) = 2\sqrt{1-\frac{4}{s^2}}$	arc-to-chord computation
$\lvert AM\rvert = \lvert MB\rvert = \lvert AB\rvert/2 = \sqrt{1-\frac{4}{s^2}}$	since M is the middle of the chord
$\measuredangle OMB = \pi/2$	perpendicular bisector through center

- Phase III: for $t \in \left[\frac{1+2\arccos(-2/s)}{s}, \frac{1+2\pi}{s}\right]$ moves toward B via the MB segment.

In Table 1, we shortly outline some core measurements on the emerging shape, e.g. angles and lengths, which will be useful in the proofs that follow. We now continue with some useful propositions.

Proposition 3. *Fast reaches A exactly when Slow reaches M.*

Proposition 4. *Fast explores the whole boundary before Slow reaches B.*

The aforementioned proposition, together with the fact that it takes $\frac{1+2\pi}{s}$ time for Fast to explore the whole boundary, provides us with the endtime for Phase III and the strategy in general.

The main result of this section follows from the combination of the upper bounds proved for Phases I, II, and III.

Theorem 5. *For any $s \geq 2$, the worst-case evacuation time of the Half-Chord strategy is at most $\frac{1+2\arccos\left(-\frac{2}{s}\right)}{s} + \sqrt{1-\frac{4}{s^2}}$.*

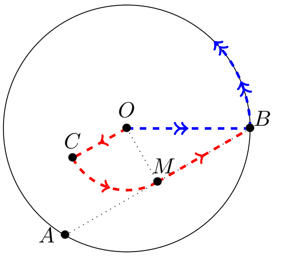


Figure 1: The Half-Chord Strategy (case $s = 4$)

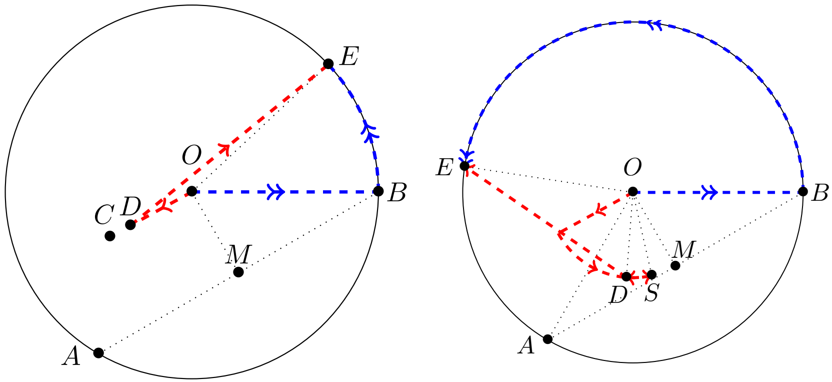
3.1.2 Phase I.

Lemma 6. *The Half-Chord evacuation strategy takes at most*

$$\frac{(1+2\arccos(-2/s))}{s} + \sqrt{1-\frac{4}{s^2}}$$

evacuation time, if the exit is found during Phase I.

Proof. We need only care about the time $t \in [1/s, 2/s]$, since for less time Slow has not yet reached the boundary. Imagine that the exit is discovered after $(1+a)/s$ time (for $a \in [0,1]$). For a visualization, the reader can refer to Figure 2a. Slow has covered $(1+a)/s$ distance on the OC segment, while Fast has explored an a part of $\widehat{BA}$. Slow now takes a segment from its current position (namely D) to the exit E. To compute $|DE|$ we use the law of cosines in $\triangle DOE$. Let $\omega = \measuredangle DOE$. In case $a \leq \frac{1}{2}$, $\omega \leq \pi$, and more accurately



(a) Exit during Phase I ($a = 0.75$)

(b) Exit during Phase II ($\tau = 0.3$)

Figure 2: Exit during Phases I & II (Examples for $s = 4$)

$\omega = a + \psi + \theta = \pi + a - \frac{1}{2}$. In case $a > \frac{1}{2}$, $\omega > \pi$, and more accurately $\omega = 2\pi - a - \psi - \theta$. Since $\cos(2\pi - x) = \cos(x)$, we can consider the two cases together. We compute,

$$|DE| = \sqrt{|OE|^2 + |OD|^2 - 2|OE||OD|\cos(\omega)} = \sqrt{1 + \tfrac{(1+a)^2}{s^2} + 2\tfrac{1+a}{s}\cos(1/2 - a)}$$

Overall, the worst-case evacuation time is given by

$$\max_{a \in [0,1]} \left\{ \frac{1+a}{s} + \sqrt{1 + \frac{(1+a)^2}{s^2} + 2\frac{1+a}{s}\cos(1/2 - a)} \right\}.$$

To conclude the proof, it suffices to observe that $\frac{2}{s} + \sqrt{1 + \frac{2^2}{s^2} + 2\frac{2}{s}}$ is an upper bound to the above quantity, since $a \le 1$ and $\cos(\cdot) \le 1$. Finally, $\frac{2}{s} + \sqrt{1 + \frac{2^2}{s^2} + 2\frac{2}{s}} \le \frac{1 + 2\arccos\left(-\frac{2}{s}\right)}{s} + \sqrt{1 - \frac{4}{s^2}}$ for any $s \ge 2$. □

3.1.3 Phase II.

Lemma 7. *The Half-Chord evacuation strategy takes at most* $\frac{1+2\arccos\left(-\frac{2}{s}\right)}{s} + \sqrt{1 - \frac{4}{s^2}}$ *evacuation time, if the exit is found during Phase II.*

Proof. We prove that the worst-case placement for the exit is point A. Suppose the exit E is found at the time when Slow lies on point S and has not yet covered a τ part of $\widehat{CM}$. The corresponding central angle is $\frac{s\tau}{2}$, since $\widehat{CM}$ is an arc on $(O, \frac{2}{s})$. At the same time, Fast has not yet explored an $s\tau$ part of $\widehat{BA}$ with a corresponding central angle of size $s\tau$. Then, Slow can move backwards on the boundary of $(O, \frac{2}{s})$ for another τ distance to point D. Now, the central angle from D to M is $\frac{s\tau}{2} + \frac{s\tau}{2} = s\tau$ and matches the central angle between E and A. Thence, due to shifting by the same central angle, we get $\measuredangle EOD = \measuredangle EOA + \measuredangle AOD = \measuredangle DOM + \measuredangle AOD = \measuredangle AOM$. Moreover, since $|OD| = |OM| = \frac{2}{s}$ and $|OE| = |OA| = 1$, triangles $\triangle EOD$ and $\triangle AOM$ are congruent meaning that $|ED| = |AB|$. To sum up, if the exit is discovered τ time before Slow reaches M, it takes at most another $\tau + \sqrt{1 - \frac{4}{s^2}}$ time for it to reach it. At the same time, it would take $\tau + \sqrt{1-\frac{4}{s^2}}$ for it to reach A. Hence, exiting through A is the worst-case scenario and yields a total time of $\frac{1+2\arccos\left(-\frac{2}{s}\right)}{s} + \sqrt{1-\frac{4}{s^2}}$. □

3.1.4 Phase III.

Lemma 8. *The Half-Chord evacuation strategy takes at most* $\frac{1+2\arccos\left(-\frac{2}{s}\right)}{s} + \sqrt{1-\frac{4}{s^2}}$ *evacuation time, if the exit is found during Phase III.*

Proof. Since $\frac{1+2\arccos\left(-\frac{2}{s}\right)}{s}$ time has already passed at the beginning of Phase III, it suffices to show that at most $\sqrt{1-\frac{4}{s^2}}$ time goes by when the exit is discovered within $\widehat{AB}$.

Suppose that the exit is discovered τ time units after the beginning of Phase III. Then, Slow lies at C (Figure 3), τ distance away from M on the MB segment. On the other hand, Fast lies on E, an $s\tau$ distance away from A on $\widehat{AB}$.

Consider a disk with center C and radius $r = \sqrt{1-\frac{4}{s^2}} - \tau$. One can notice that (C, r) intersects $(O, 1)$ at two points: one of them is B and the other one is D, where D is included in $\widehat{AB}$, since $|AC| \geq r$ for any choice of $\tau \geq 0$. Moreover, we draw the chord DB and its middle point, say M'. Now, notice that OM' is perpendicular to DB, since DB is a chord of $(O, 1)$ and also that OM' passes through C, since DB is also a chord of (C, r). To conclude, we exhibit that E is included in $\widehat{DB}$. Equivalently, that $|\widehat{AE}| \geq |\widehat{AD}|$. We look into two cases.

First, that $\measuredangle AOD \leq \measuredangle AOM$. In this case, we compute $\measuredangle AOD = \measuredangle AOM - \measuredangle DOM = \measuredangle MOB - \measuredangle DOM = \measuredangle MOM' + \measuredangle M'OB - \measuredangle DOM = \measuredangle MOM' + \measuredangle DOM' - \measuredangle DOM = 2 \cdot \measuredangle MOM'$, since $\measuredangle AOM = \measuredangle MOB$ and $\measuredangle M'OB = \measuredangle DOM'$ from the fact that OM (OM') bisects AB (DB). Moreover, $\measuredangle DOM' - \measuredangle DOM = \measuredangle MOM'$. We compute $\measuredangle MOM' = \arctan(s\tau/2)$ by the right triangle $\triangle MOC$. Finally, $\measuredangle AOD = 2\arctan(s\tau/2) \leq s\tau = \measuredangle AOE$, since $\arctan(x) \leq x$ for $x \geq 0$.

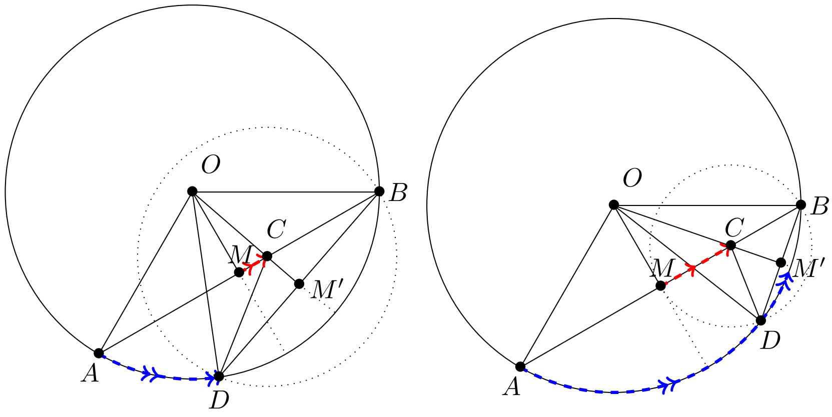
For the second case, $\measuredangle AOD > \measuredangle AOM$. Then, $\measuredangle AOD = \measuredangle AOM + \measuredangle MOD = \measuredangle MOB + \measuredangle MOD = \measuredangle MOM' + \measuredangle M'OB + \measuredangle MOD = \measuredangle MOM' + \measuredangle DOM' + \measuredangle MOD = 2 \cdot \measuredangle MOM'$, again by using the equalities deriving from bisecting the chords. The rest of the proof follows as before. □

3.2 The Half-Chord Strategy for $1 \leq s \leq 2$

We first observe that, for $s = 2$, the name "Half-Chord" is slightly misleading, as the points A, B, and M coincide. The time needed for $s = 2$ is, as shown in Theorem 5, $\frac{1+2\pi}{s}$. Note also that the Half-Chord strategy is a BES strategy for $s = 2$.

For $s < 2$, Slow can simply move even slower, namely with speed $\frac{s}{2}$. Using the same paths as for $s = 2$, this provides the same upper bound of $\frac{1+2\pi}{s}$.

Theorem 9. *For $1 \leq s \leq 2$, the (generalized) Half-Chord strategy leads to a $\frac{1+2\pi}{s}$ evacuation time.*



(a) First case ($\tau = \frac{1}{5}$)

(b) Second case ($\tau = \frac{1}{2}$)

Figure 3: Exit during Phase III (when $s = 4$; exit E lies at the end of Fast's arrow)

3.3 The Both-to-the-Same-Point Strategy

This BES strategy follows the same key idea presented in [8] where it is proven to be optimal for $s = 1$.

3.3.1 The Strategy.

In the *Both-to-the-Same-Point Strategy* (shortly *BSP* strategy), initially both robots set out toward the same boundary point moving in a beeline. Once they arrive there, they move to opposite directions along the boundary. This goes on, until the exit has been found by either robot or the robots meet each other on the boundary. We restrict the analysis of BSP for $s \in [1, 2]$, since for $s > c_{1.71}$ this strategy becomes non-dominant.

Theorem 10. *The BSP strategy requires evacuation time at most*

$$1 + 2\sqrt{1 - \frac{1}{(s+1)^2}} + \frac{2\arccos(-\frac{1}{s+1}) - s + 1}{s + 1}$$

when $s \in [1, 2]$.

3.4 The Fast-Chord Strategy

In the Half-Chord strategy for $s = 2$, we observe that the final point reached after Phase I, i.e. point C, lies on the disk boundary. Thence, after that, Slow explores $\widehat{CB}$, but so does Fast (since by its strategy it explores the whole boundary). This seems like an unnecessary double-exploring of this part of the boundary. Thus, we propose a new strategy, where Fast reaches C as usual, but then traverses the CB chord, instead of $\widehat{CB}$. Furthermore, we could vary the position of C, in order for Fast to reach B (for the second time) exactly when Slow reaches D (a point before B) and so get Fast to explore some part of the boundary in clockwise fashion as well. In this case, Slow does not traverse the whole $\widehat{CB}$. Let us now describe more formally this *Fast-Chord* family of strategies. All arcs are considered in *counterclockwise* fashion unless otherwise stated. Below, let $|\widehat{BA}| = s-1$, $x_1 = |\widehat{AC}|$, $x_2 = |CB|$, $x_3 = |\widehat{DB}|$ and $y = |\widehat{CB}|$; see Figure 4.

3.4.1 The Strategy.

Fast moves as follows until the exit is found:

- for $t \in \left[0, \frac{1}{s}\right]$ moves toward B,
- Phase I: for $t \in \left(\frac{1}{s}, 1\right]$ traverses $\widehat{BA}$,
- Phase IIa: for $t \in \left(1, 1+\frac{x_1}{s}\right]$ traverses $\widehat{AC}$,
- Phase IIb: for $t \in \left(1+\frac{x_1}{s}, 1+\frac{x_1+x_2}{s}\right]$ traverses CB and
- Phase IIc: for $t \in \left(1+\frac{x_1+x_2}{s}, 1+\frac{x_1+x_2}{s}+\frac{x_3}{s+1}\right]$ moves toward D (clockwise) till it meets Slow.

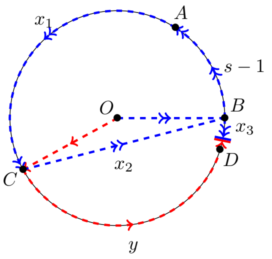


Figure 4: The Fast-Chord Family of Strategies

Slow moves as follows until the exit is found:

- for $t \in [0, 1]$ moves toward C,
- for $t \in (1, 1+y]$ traverses $\widehat{CD}$,
- for $t \in \left(1+y, 1+y+\frac{x_3}{s+1}\right]$ traverses $\widehat{DB}$ till it meets Fast.

The following system of equations describes the relationship between the variable distances:

$$\begin{cases} x_1 + y + x_3 + s - 1 & = 2\pi \quad \text{(I)} \\ x_2 & = 2\sin\left(\frac{x_3+y}{2}\right) \quad \text{(II)} \\ x_1 + x_2 & = s \cdot y \quad \text{(III)} \end{cases}$$

Equation (I) suggests how the disk boundary is partitioned. Equation (II) suggests that x_2 is the chord of an arc with length x_3+y. Equation (III) suggests that Fast traverses x_1 and x_2 at the same time as slow traverses y. That is, since Fast lies on A exactly when Slow lies on C, then Fast arrives at B (for the second time) exactly when Slow arrives at D. The latter happens at time $1+y = 1+\frac{x_1+x_2}{s}$. The remaining x_3 part of the boundary can be explored in time $\frac{x_3}{s+1}$, since both robots explore it concurrently until they meet. Hence, within $\frac{x_3}{s+1}$ time, they can explore a distance equal to $s \cdot \frac{x_3}{s+1} + \frac{x_3}{s+1} = (s+1) \cdot \frac{x_3}{s+1} = x_3$. All variables are non-negative representing distance.

The idea behind this paradigm is to try different values for x_3 and then solve the above system to extract x_1, x_2 and y. Nonetheless, due to the $\sin(\cdot)$ function in equation (II), it is not possible to obtain a symbolic solution. Thence, we hereby provide bounds computed *numerically*. For any value of s, we iterate over all possible x_3 values and then solve the above system numerically. For each x_3 value and for each exploration phase, we use a small time step and compute the worst-case evacuation time. Then, we can select the x_3 value which minimizes this worst-case time. All this numerical work is implemented in Matlab. We iterate over x_3 in the interval $[0, 2\pi - s + 1]$. The upper bound for x_3 stems from the case $x_1 = y = 0$. Indeed, notice that, for $s = 1$, Fast-Chord is exactly BSP when we set $x_1 = y = 0$. For the time parameter, namely t, we iterate in the interval $\left[0, 1 + \frac{x_1+x_2}{s} + \frac{x_3}{s+1}\right]$. Finally, we use a parametric representation of the disk (where the center O lies on coordinates $(0,0)$) to calculate the distance between the two robots.

By studying the numerical bounds we obtain via the Fast-Chord method, we state the following result, in comparison to the other two strategies studied in this paper.

Theorem 11. *Fast-Chord performs better than (Generalized) Half-Chord for $s \in (c_{1.71}, c_{2.07})$. It also performs better than Both-to-the-Same-Point for $s \geq c_{1.71}$.*

4 Lower Bounds

The main tool behind our lower bounds is the following lemma from [8].

Lemma 12 (Lemma 5 [8]). *Consider a boundary of a disk whose subset of total length $u + \epsilon > 0$ has not been explored for some $\epsilon > 0$ and $\pi \geq u > 0$. Then there exist two unexplored boundary points between which the distance along the boundary is at least u.*

4.1 Fast Explores

Lemma 13. *Any FES-strategy takes at least*

- $\frac{1+2\pi}{s}$ *time for any $s \in [1,2]$ and*
- $\frac{1+2\arccos\left(-\frac{2}{s}\right)}{s} + \sqrt{1-\frac{4}{s^2}}$ *time for any $s \geq 2$.*

Proof. For any s, Fast needs at least $\frac{1+2\pi}{s}$ time to explore the whole boundary. We now show a better bound for $s \geq 2$. At time $\frac{1+a}{s}$ (where $a \geq 0$), Fast has explored at most an a part of the boundary. Then, if we consider the time $\frac{1+a-\epsilon}{s}$ (where $\epsilon > 0$), a $2\pi - (a - \epsilon) = 2\pi - a + \epsilon$ subset of the boundary has not yet been explored. We bound $a \in [\pi, 2\pi)$ such that $0 < 2\pi - a \leq \pi$ holds. We now apply Lemma 12 with $u = 2\pi - a$ and ϵ. Thence, there exist two unexplored boundary points between which the distance along the boundary is at least u. Let us now consider the perpendicular bisector of the chord connecting these two points. Depending on which side of the bisector Slow lies, an adversary may place the exit on the boundary point lying at the opposite side. The best case for Slow is to lie exactly on the point of the bisection. That is, Slow will have to cover a distance of at least $\frac{2\sin\left(\frac{u}{2}\right)}{2} = \sin\left(\frac{a}{2}\right)$, where $2\sin\left(\frac{u}{2}\right)$ is the chord length. In this case, the overall evacuation time is equal to $\frac{1+a}{s} + \sin\left(\frac{a}{2}\right)$ and for the best lower bound we compute $\max\limits_{\pi \leq a < 2\pi} \left\{\frac{1+a}{s} + \sin\left(\frac{a}{2}\right)\right\}$. □

4.2 Both Explore

The following lower bound is a result of applying Lemma 12 to obtain a generalization of the lower bound proved in [8]. The proof considers a timestep when both robots have explored some part of the boundary and lie on the opposite ends of a long chord. Then, an adversary acts according to his best interests. He either places the exit on the end opposite Fast or in the end being farthest to Slow; the latter leading to a chord bisection argument similar to the one used in Lemma 13.

Lemma 14. *Any BES-strategy takes at least*

- $1+\frac{2}{s}\sqrt{1-\frac{s^2}{(s+1)^2}}+\frac{-s+2\arccos\left(-\frac{s}{s+1}\right)+1}{s+1}$ *time for* $s\in[1,2)$*,*
- $1+\sqrt{1-\frac{4}{(s+1)^2}}+\frac{-s+2\arccos\left(-\frac{2}{s+1}\right)+1}{s+1}$ *for* $s\in[2,c_{4.84}]$ *(where* $c_{4.84}\approx 4.8406$*) and*
- $1+\sin\left(\frac{s-1}{2}\right)$ *time for* $s\in(c_{4.84},2\pi+1)$*.*

The above lower bound loses its value as s grows. Hence, there is a need to capture a lower bound for the case where Slow has not explored any part of the boundary yet. This is possible, since we can apply an FES lower bound idea when s is big enough.

Lemma 15. *Any BES-strategy takes at least*

- $1+\sin\left(\frac{s-1}{2}\right)$ *time for* $s\in(\pi+1,c_{4.97})$*, where* $c_{4.97}\approx 4.9699$*, and*
- $\frac{1+2\arccos\left(-\frac{2}{s}\right)}{s}+\sqrt{1-\frac{4}{s^2}}$ *time for* $s\geq c_{4.97}$*.*

4.3 An Improvement for Both Explore

We now obtain numerical values for a stronger BES lower bound by performing a more complex analysis on the *Original BES* lower bound proof given in Lemma 14. The main idea behind the improvement is to provide a better bound for the subcase when the adversary places the exit on the farthest endpoint from Slow's current position. Apparently, the best play for Slow is to lie exactly on the midpoint of the chord with the unexplored endpoints. Nevertheless, in order for Slow to be there, it needs to spend some of its time, originally destined for exploration, within the disk interior. We hereby examine the best possible scenario for Slow in terms of its distance from the midpoint following the above reasoning. Let us refer to this lower bound as *Improved BES*.

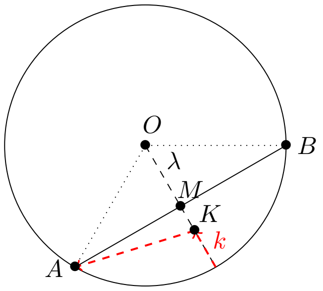


Figure 5: An Improved BES Lower Bound

Lemma 16. *Improved BES is greater or equal to Original BES for any* $s\geq 1$*.*

Proof. At time $1+y$, where $y \geq 0$ is a variable, Fast has explored at most an $s-1+sy$ part of the boundary and Slow has explored at most a y part of the boundary. Now suppose that Slow has spent k time, where $k \in [0, y]$, *not exploring* the boundary, i.e. moving within the disk interior.

Notice that it takes $1 + \frac{2\pi - s + 1}{s+1}$ time for the whole perimeter to be explored, when both robots are only exploring after timestep 1. Thence, we upper-bound $y \leq \frac{2\pi - s + 1}{s+1}$. To lower-bound y, we restrict the unexplored part $u = 2\pi - s + 1 - (s+1)y + k \leq \pi$. That is, we get $y \geq \max\{\frac{\pi - s + 1 + k}{s+1}, 0\}$. Moreover, $u > 0$ is already covered by the aforementioned upper bound.

Now, we are ready to apply Lemma 12: There exist two unexplored points (say A, B) with arc distance $\geq 2\pi - s + 1 - (s+1)y + k$, which implies that the chord between them has length at least $2\sin\left(\frac{2\pi - s + 1 - (s+1)y + k}{2}\right) = 2\sin\left(\frac{s - 1 + (s+1)y - k}{2}\right)$. An adversary could place the exit on any of the two endpoints. If Slow reaches an endpoint first (case I), then the exit is placed on the other side, such that Slow has to traverse the chord. If Fast reaches an endpoint first, then the exit is placed either on the other side (case II), meaning that Fast has to traverse the chord, or on the endpoint that lies the farthest from Slow's current position (case III), meaning that Slow has to traverse at least half the chord. We assume that both the robots and the adversary behave optimally. Hence, the robots will always avoid case I.

Let us now examine more carefully what happens in case III. For a depiction of the proof, see Figure 5. The ideal location for Slow is to lie exactly on the chord midpoint, say M. Nevertheless, this may not be possible due to it only spending k time within the disk interior. Let us consider the minimum distance from the chord midpoint to the boundary. This is exactly $1-\lambda$, where $\lambda = |OM|$ is the distance from the midpoint to the center of the disk. Notice that OM intesects AB *perpendicularly*, since M is the midpoint of chord AB. Using the Pythagorean theorem in $\triangle AMO$, we get $\lambda = \sqrt{1 - \sin^2\left(\frac{s-1+(s+1)y-k}{2}\right)}$. If we consider the case when $1 - \lambda > k$, then the ideal position for Slow is to lie k distance away from the boundary and on the extension of OM (i.e. on point K). From there, Slow can take a beeline to the exit, yielding a $\sqrt{\sin^2\left(\frac{s-1+(s+1)y-k}{2}\right) + (1-\lambda-k)^2}$ distance again by the Pythagorean theorem, now in $\triangle AMK$.

To conclude, Slow will try to minimize this beeline distance over k, while the adversary will select a case between II and III that maximizes the total distance. Overall, the optimization problem reduces to computing:

$$\max_{y \in [y_{min}, y_{max})} \left\{ 1 + y + \max \left\{ \begin{array}{l} \min\limits_{k \in [0,y]} \frac{2}{s} \sin\left(\frac{s-1+(s+1)y-k}{2}\right), \\ \min\limits_{k \in [0,y]} \sqrt{\sin^2\left(\frac{s-1+(s+1)y-k}{2}\right) + \max\left\{1-\lambda-k, 0\right\}^2} \end{array} \right\} \right\} \tag{1}$$

Note that the above bound matches the original one for $1 - \lambda < k$.

Last but not least, we need also consider the case where the adversary chooses to place the exit on the last boundary point to be explored. In the current setting, it takes at least $\frac{u}{s+1} = \frac{2\pi - s + 1 - (s+1)y + k}{s+1}$ extra time for both robots to explore the rest of the boundary, since Fast explores $s\frac{u}{s+1}$ while Slow explores $\frac{u}{s+1}$ for a total distance of u. Overall, we are looking to compute $\max\limits_{y \in [y_{min}, y_{max})} \left\{ 1 + y + \frac{2\pi - s + 1 - (s+1)y}{s+1} \right\}$, since Slow wishes to minimize k. Due to the inherent complexity of the optimization problem (1), we compute *numerical* bounds. The computational work is done in Matlab, where we iterate over feasible values of y and k. The resulting bounds show that, for all $s \in [1, 2\pi+1)$, this lower bound is greater or equal to the lower bound given in Lemma 14 with $k = 0$ *always* selected as the minimizer. □

5 Comparison and Future Work

Regarding the lower bounds, for each value of s we select the minimum (weakest) lower bound between the (maximum) BES and FES ones as our overall lower bound. We see that Improved BES is strictly stronger than Original BES for any $s \geq c_{1.71} \approx 1.71$. Moreover, Improved BES is stronger than the FES lower bound for $s \geq c_{2.75} \approx 2.75$.

As far as the upper bounds are concerned, we notice that Half-Chord outperforms BSP for any $s \geq c_{1.86} \approx 1.856$. Besides, Fast-Chord outperforms BSP for any $s \geq c_{1.71} \approx 1.71$. Finally, Fast-Chord outperforms Half-Chord for any $s \leq c_{2.07} \approx 2.072$. That is, the introduction of Fast-Chord yields a better upper bound for any $s \in [c_{1.71}, c_{2.07}]$.

By comparing upper and lower bounds, we see that Half-Chord is optimal for $s \geq c_{2.75}$, since the matching FES lower bound is the weakest in this interval. On the other hand, for $s < c_{2.75}$ the ratio between the bounds is at most 1.22 (maximized when $s = c_{1.71}$), where the strategy changes from BSP to Fast-Chord. The best strategy to use is BSP when $s < c_{1.71}$, Fast-Chord when $c_{1.71} < s < c_{2.07}$ and Half-Chord for $s \geq c_{2.07}$.

Optimality for the case $1 < s < c_{2.75}$ remains open. Regarding further work, one could consider a more-than-two-robots evacuation scenario. Moreover, the non-wireless case for two-robots fast evacuation seems to be quite challenging-given that exact optimality is complex to obtain even for $s = 1$ ([10]).

References

[1] S. Alpern and S. Gal. *The Theory of Search Games and Rendezvous.* Int. Series in Operations Research and Management Science, number 55, Kluwer Academic Publishers, 2002.

[2] R.A. Baeza-Yates, J.C. Culberson, and G.J.E. Rawlins. Searching in the plane. *Inform. and Comput.* **106** (1993), pp. 234–252.

[3] A. Beck. On the linear search problem. *Naval Res. Logist.* **2** (1964), pp. 221–228.

[4] A. Bonato and R.J. Nowakowski. *The Game of Cops and Robbers on Graphs.* American Mathematical Society, 2011.

[5] P. Bose, J.-L. De Carufel, and S. Durocher. Revisting the problem of searching a line. *ESA 2013*, pp. 205–216.

[6] Y. Chevaleyre. Theoretical analysis of the multi-agent patrolling problem. *IAT 2004*, pp. 302–308.

[7] M. Chrobak, L. Gąsieniec, T. Gorry, and R. Martin. Group search on the line. *SOFSEM 2015*, pp. 164–176.

[8] J. Czyzowicz, L. Gąsieniec, T. Gorry, E. Kranakis, R. Martin, and D. Pająk, Evacuating robots from an unknown exit located on the boundary of a disc, *DISC 2014*, LNCS 8784, pp. 122-136, Springer, 2014.

[9] J. Czyzowicz, L. Gąsieniec, A. Kosowski, and E. Kranakis. Boundary patrolling by mobile agents with distinct maximal speeds. *ESA 2011*, pp. 701–712.

[10] J. Czyzowicz, K. Georgiou, E. Kranakis, L. Narayanan, J. Opatrny, and B. Vogtenhuber, Evacuating robots from a disk using face-to-face communication, *CIAC 2015*, pp. 140–152.

[11] O. Feinerman, A. Korman, S. Kutten and Y. Rodeh, Fast Rendezvous on a cycle by agents with different speeds, *ICDCN 2014*, LNCS, vol. 8314, pp. 1-13, Springer, 2014.

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

[13] E. Huus and E. Kranakis, Rendezvous of Many Agents with Different Speeds in a Cycle, *ADHOC-NOW 2015*, LNCS 9143, pp. 195-209, Springer, 2015.

[14] A. Jeż, and J. Łopuzański. On the two-dimensional cow search problem. *Information Processing Letters* **131** (2009), pp. 543–547.

[15] M.-Y. Kao, J.H. Reif, and S.R. Tate. Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. *Inform. and Comput.* **131** (1996), pp. 63–80.

[16] E. Kranakis, D. Krizanc, and S. Rajsbaum. Mobile agent rendezvous: A survey. *SIROCCO 2006*, pp. 1–9.

[17] T.D. Parsons. Pursuit-evasion in a graph, in *Theory and Applications of Graphs: Proc., Michigan May 11–15*, pp. 426–441, 1976.

[18] V. Yanovski, I.A Wagner, and A.M. Bruckstein. A distributed ant algorithm for efficiently patrolling a network. *Algorithmica* **37** (2003), pp. 165–186.