

Geometric Freeze-Tag Problem

Sharareh Alipour

Tehran Institute for Advanced Studies, Khatam university
Tehran, Iran
sharareh.alipour@gmail.com

Mahdis Mirzaei

University of Tehran
Tehran, Iran
mahdis.mirzaei@ut.ac.ir

Kajal Baghestani

Sharif University of Technology
Tehran, Iran
kajal.baghestani83@sharif.edu

Soroush Sahraei

University of Tehran
Tehran, Iran
soroush.sahraei@ut.ac.ir

ABSTRACT

We study the Freeze-Tag Problem (FTP), introduced by Arkin et al. (SODA’02), where the goal is to wake up a group of n robots, starting from a single active robot. Our focus is on the geometric version of the problem, where robots are positioned in \mathbb{R}^d , and once activated, a robot can move at a constant speed to wake up others. The objective is to minimize the time it takes to activate the last robot, also known as the makespan.

We present new upper bounds for the l_1 and l_2 norms in \mathbb{R}^2 and \mathbb{R}^3 . For (\mathbb{R}^2, l_2) , we achieve a makespan of at most $5.4162r$, improving on the previous bound of $7.07r$ by Bonichon et al. (DISC’24). In (\mathbb{R}^3, l_1) , we establish an upper bound of $13r$, which leads to a bound of $22.52r$ for (\mathbb{R}^3, l_2) . Here, r denotes the maximum distance of a robot from the initially active robot under the given norm. To the best of our knowledge, these are the first known bounds for the makespan in \mathbb{R}^3 under these norms.

We also explore the FTP in (\mathbb{R}^3, l_2) for specific instances where robots are positioned on a boundary, providing further insights into practical scenarios.

KEYWORDS

Freeze-Tag Problem; Makespan; Approximation Algorithms; Robot Routing; Scheduling; Binary Trees; Distributed Computing

ACM Reference Format:

Sharareh Alipour, Kajal Baghestani, Mahdis Mirzaei, and Soroush Sahraei. 2025. Geometric Freeze-Tag Problem. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025)*, Detroit, Michigan, USA, May 19 – 23, 2025, IFAAMAS, 9 pages.

1 INTRODUCTION

The Freeze-Tag Problem (FTP), introduced by Arkin et al. [5], involves one active robot and n inactive robots in a metric space. Active robots can move at a constant speed, while inactive robots can only be activated when reached by an active one. The goal is to minimize the makespan, the total time needed to wake up all inactive robots.



This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

FTP has several applications in robotics. Related algorithmic problems have been studied for controlling swarms of robots to perform tasks such as environment exploration [2, 3, 9, 11, 12, 18, 19], robot formation [16, 17], and searching [18], as well as multi-robot formation in continuous and grid environments [10, 16, 17]. FTP also has applications in network design, including broadcast and IP multicast problems [4, 6, 14].

FTP is NP-Hard in high-dimensional metrics like centroid metrics [4] (based on weighted star n -vertex graphs) and unweighted graph metrics with a robot per node [6]. Subsequent research has extended this hardness result to constant-dimensional metric spaces, including Euclidean ones. A series of papers [1, 13, 15] proves that FTP is NP-Hard in (\mathbb{R}^3, l_p) for all $p \geq 1$, meaning it is NP-Hard in 3D with any l_p -norm. In 2D, FTP is known to be NP-Hard for (\mathbb{R}^2, l_2) , though the complexity for other norms remains open [1]. It is believed that FTP is also NP-Hard for (\mathbb{R}^2, l_1) [4].

In a geometric Freeze-Tag Problem (FTP) instance, the input consists of the positions of n inactive (asleep) robots and one active robot in \mathbb{R}^d , along with the distance norm l_p . The output is the makespan, a real number representing the minimum time needed to wake up all robots. Each active robot moves at a constant speed.

In \mathbb{R}^d with a norm η , the unit η -ball is the set of all points within a distance of one from the origin, where distance is measured by η , i.e., $\|v - u\|_\eta$ for points u and v . For the l_p norm, we write $\|u - v\|_p$, with $\|u - v\|_2$ representing the Euclidean distance, or the length of the line segment \overline{uv} . In the case of robots in (\mathbb{R}^d, l_p) , we assume all robots are inside the l_p -unit ball in \mathbb{R}^d , with the distance measured using the l_p norm and the initial active robot positioned at the origin. For simplicity, in \mathbb{R}^2 , we refer to the region as the unit η -disk. For example, the unit l_2 -disk is a regular disk, while the unit l_1 -disk is a square rotated by 45 degrees. In \mathbb{R}^3 , the unit l_2 -ball is the region enclosed by a sphere, while the unit l_1 -ball is shown in Figure 1.

According to Arkin et al. [5], FTP can be reformulated as finding a rooted spanning tree on a set of points that minimizes the weighted depth. The root (representing the awake robot) has one child, and all other nodes (representing the n sleeping robots) can have up to two children (see Figure 2). Each edge represents the distance between two points in the metric space. This tree is called a wake-up tree, and its weighted depth is the wake-up time.

We define $\gamma_n(\mathbb{R}^d, \eta)$ as the worst-case optimal wake-up time of a wake-up tree for any set of n sleeping robots located in the unit η -ball, rooted at the origin in \mathbb{R}^d . In other words, $\gamma_n(\mathbb{R}^d, \eta)$

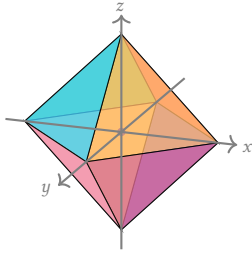


Figure 1: Representation of the unit l_1 -ball in \mathbb{R}^3

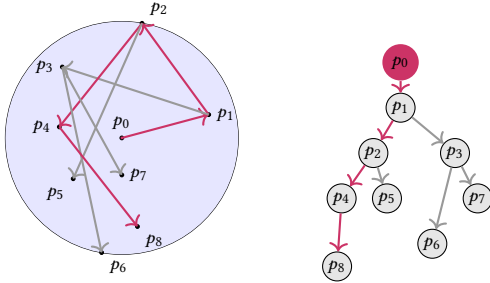


Figure 2: An FTP instance and its wake-up tree. The left diagram shows the positions and movements of the robots inside the l_2 -disk, while the right diagram displays the corresponding wake-up tree. Red arrows indicate the path from the root to a leaf.

represents the best possible upper bound for the makespan of n sleeping robots, with the awake robot placed at the origin in the unit η -ball of \mathbb{R}^d . The wake-up ratio w.r.t the η -norm is then defined as: $\gamma_{(\mathbb{R}^d, \eta)} = \max_{n \in \mathbb{N}} \gamma_n(\mathbb{R}^d, \eta)$.

Despite extensive research on the complexity of FTP, few results address the wake-up ratio. For (\mathbb{R}^2, l_1) , Bonichon et al. proved that the wake-up ratio is at most 5 and provided an example with a makespan of 5 [7]. This gives an upper bound of $5\sqrt{2}$ for the wake-up ratio in (\mathbb{R}^2, l_2) , improving the previous upper bound of 10.06 found in [20].

For any set of points, if the awake robot is at most distance r from all sleeping robots and we have an upper bound c for the makespan, scaling the unit ball with these positions allows constructing a wake-up tree with a makespan of at most $r \times c$. This gives a c -approximation factor algorithm for FTP, since r is a trivial lower bound on the makespan.

2 OUR RESULTS

In this paper, we begin by focusing on FTP within the unit l_2 -disk and introduce two new wake-up strategies. Our main result is an improved makespan, reducing the previous bound from 7.07 to 5.4162, as established in [7]. Formally, we state the following theorem:

THEOREM 2.1. *A robot at the origin can wake up any set of n asleep robots in the unit l_2 -disk with a makespan of at most 5.4162.*

The proof of this theorem is provided in Section 3.1. Also, concurrently with our work, [8] reported a ratio of $4.63r$ for (\mathbb{R}^2, l_2) . Our first strategy is similar to their strategy; however, our second strategy is completely different. Next, we examine FTP in \mathbb{R}^3 and propose a new strategy for (\mathbb{R}^3, l_1) . We establish an upper bound of 13 for (\mathbb{R}^3, l_1) , which leads to an upper bound of $13\sqrt{3}$ for (\mathbb{R}^3, l_2) . To our knowledge, no previous bounds for the makespan in \mathbb{R}^3 have been provided. Formally, we state the following theorem:

THEOREM 2.2. *A robot at the origin can wake up any set of n asleep robots in the unit l_1 -ball in \mathbb{R}^3 with a makespan of at most 13.*

The proof of this theorem is presented in Section 4. Our approaches for \mathbb{R}^2 and \mathbb{R}^3 are fundamentally different. In \mathbb{R}^2 , we start at the origin and initially awaken robots within a disk of radius zero, since only the origin robot is awake. In each step, we expand the radius of this disk, waking up the robots that are closer to the origin, until the radius reaches 1. By the end, all robots inside the disk are awake. However, this method cannot be extended to \mathbb{R}^3 . Therefore, we employ a different strategy for \mathbb{R}^3 .

For \mathbb{R}^3 , when the number of robots is small, we solve the problem directly in a proper time. For larger numbers of robots, we divide the unit ball into smaller partitions and apply a recursive approach to solve the problem within each partition. This strategy allows us to handle a larger number of robots effectively.

Finally, we examine a version of FTP in (\mathbb{R}^3, l_2) , where the asleep robots are located on the boundary of the unit l_2 -ball. Using our approach for (\mathbb{R}^2, l_2) , we show that FTP in (\mathbb{R}^3, l_2) for asleep robots on the boundary can be solved with a makespan of at most 12.37. Formally, we have:

THEOREM 2.3. *A robot at the origin can wake up any set of n asleep robots on the boundary of the unit l_2 -ball in \mathbb{R}^3 with a makespan of at most 12.37.*

The proof of this theorem is deferred to Section 5.2. Our approach involves mapping the points on the boundary of the unit l_2 -ball in \mathbb{R}^3 to a l_2 -disk in \mathbb{R}^2 . By solving the problem in \mathbb{R}^2 , we obtain a wake-up strategy for \mathbb{R}^3 . One motivation for tackling the problem in this setting is our conjecture that the maximum makespan for n points is reached when the points lie on the boundary of the unit η -ball.

A similar wake-up strategy allows us to tackle another variant of FTP, which we call surface-FTP. In this version, the points are positioned on the boundary of a unit l_2 -ball in \mathbb{R}^3 , with the active robot also on the surface. The distance between two points v and u is defined as the geodesic distance, or the length of the shortest arc connecting them on the surface. This problem has practical applications in areas like communication and transportation on Earth's surface, where geodesic distance between points is important. Formally, we have:

THEOREM 2.4. *Given an instance of surface-FTP, the makespan is at most 11.65.*

This theorem is proven in Section 5.3.

3 FTP IN (\mathbb{R}^2, l_2)

3.1 The wake-up ratio is at most 5.4162 in (\mathbb{R}^2, l_2)

This section establishes an upper bound of 5.4162 for $\gamma_{(\mathbb{R}^2, l_2)}$. We present two key strategies for computing a wake-up tree and combine them to achieve an improved makespan.

3.1.1 Arc-Strategy. We introduce the Arc-Strategy as our initial approach. For an instance of FTP in (\mathbb{R}^2, l_2) , the Arc-Strategy is described as follows.

- The awake robot, p_0 , starts at the center of a unit l_2 -disk and moves to the position of the nearest asleep robot, p_1 , located at point A . At this stage, both p_0 and p_1 are active.
- Next, divide the disk into two halves by drawing a line through the center and point A . Each of the two active robots is responsible for waking up the asleep robots in one half of the disk. Their strategy is as follows: in a parallel step, p_0 activates the nearest asleep robot to the center in its half, denoted as p_2 , while p_1 activates the nearest asleep robot to the center in its half, denoted as p_3 .
- Now, p_0 and p_2 divide their half of the disk into two quarters. p_0 handles its quarter by activating the nearest robot to the center in its region, while p_2 does the same in its own quarter. Similarly, p_1 and p_3 each take responsibility for their respective quarters, activating the nearest robots to the center within their areas.
- In each step, when a robot p_i activates another robot p_j , where p_j is the nearest robot to the center within p_i 's assigned portion, p_i and p_j divide their region into two halves. Each robot then takes responsibility for its own half. If p_i is the only robot remaining in its portion, it stops and does nothing in subsequent rounds.

When the Arc-Strategy terminates, all the robots are awake. The following lemma provides the wake-up time for the Arc-Strategy.

LEMMA 3.1. *The Arc-Strategy provides an upper bound of 7.9651 for the wake-up ratio in (\mathbb{R}^2, l_2) .*

PROOF. At the end of the Arc-Strategy, all robots are awake. To determine the wake-up time of the Arc-Strategy, we need to calculate the time it takes to wake up all the robots. The Arc-Strategy produces a wake-up tree, and the wake-up time is the length of the longest path from the center to the leaves of this tree.

Without loss of generality, consider a path p_0, \dots, p_k , which starts at the root, p_0 , and ends at a leaf, p_k , in the wake-up tree. We now define the following variables: (see Figure 3).

- d_i = $\|p_{i-1} - p_i\|_2$
- r_i = $\|p_i - O\|_2$, where O is the center of unit l_2 -disk
- $C(O, r_i)$ = the circle with center O and radius r_i
- c_i = cross point of $\overline{p_i O}$ and $C(O, r_{i-1})$
- a_i = $\|p_i - c_i\|_2$
- b_i = $\|p_{i-1} - c_i\|_2$
- α_i = The angle between $\overline{O p_i}$ and $\overline{O p_{i+1}}$
- β_i = The angle between p_{i+1} and c_{i+1} and p_i

The total length of the path is expressed as:

$$\sum_{i=1}^k d_i = d_1 + d_2 + d_3 + \sum_{i=4}^k \sqrt{b_i^2 + a_i^2 - 2a_i b_i \cos \beta_{i-1}}$$

Since $\beta_i = \frac{\pi}{2} + \frac{\alpha_i}{2}$, the path length simplifies to

$$\sum_{i=1}^k d_i = d_1 + d_2 + d_3 + \sum_{i=4}^k \sqrt{b_i^2 + a_i^2 + 2a_i b_i \sin \frac{\alpha_{i-1}}{2}}$$

Given $b_{i+1} = 2r_i \sin \frac{\alpha_i}{2}$, the total path length becomes:

$$\sum_{i=1}^k d_i = d_1 + d_2 + d_3 + \sum_{i=4}^k \sqrt{a_i^2 + 4r_{i-1}^2 (\sin \frac{\alpha_{i-1}}{2})^2 + 4a_i r_{i-1} (\sin \frac{\alpha_{i-1}}{2})^2}$$

We have,

$$\begin{aligned} & \sum_{i=4}^k \sqrt{a_i^2 + 4r_{i-1}^2 (\sin \frac{\alpha_{i-1}}{2})^2 + 4a_i r_{i-1} (\sin \frac{\alpha_{i-1}}{2})^2} \\ & \leq \sum_{i=4}^k \sqrt{a_i^2 + 4r_{i-1}^2 (\sin \frac{\alpha_{i-1}}{2})^2 + 4a_i r_{i-1} \sin \frac{\alpha_{i-1}}{2}} \\ & = \sum_{i=4}^k \sqrt{(a_i + 2r_{i-1} \sin \frac{\alpha_{i-1}}{2})^2} = \sum_{i=4}^k a_i + 2r_{i-1} \sin \frac{\alpha_{i-1}}{2} \end{aligned}$$

Now, we have the following inequality:

$$\sum_{i=1}^k d_i \leq d_1 + d_2 + d_3 + \sum_{i=4}^k a_i + 2 \sum_{i=4}^k r_{i-1} \sin \frac{\alpha_{i-1}}{2}$$

Note that $\sum_{i=4}^k a_i = 1 - r_3$ and $d_1 = r_1$ and $d_2 \leq r_1 + r_2$ and $d_3 \leq r_2 + r_3$ using triangle inequality so by replacing them and considering all $r_i \leq 1$ we have:

$$\sum_{i=1}^k d_i \leq r_1 + r_1 + r_2 + r_2 + r_3 + 1 - r_3 + 2 \sum_{i=3}^k \sin \frac{\alpha_i}{2}$$

Given that $\alpha_i \leq \frac{\pi}{2^{i-2}}$ for $i > 1$, we can write:

$$\sum_{i=1}^k d_i \leq 1 + 2r_1 + 2r_2 + 2 \left(\sin \frac{\pi}{4} + \sin \frac{\pi}{8} + \dots \right)$$

Using the fact that for $x \geq 0$, $\sin x \leq x$, we have:

$$\begin{aligned} \sum_{i=1}^k d_i & \leq 1 + 2r_1 + 2r_2 + 2 \left(\sin \frac{\pi}{4} + \sin \frac{\pi}{8} \right) + 2\pi \left(\frac{1}{16} + \dots + \frac{1}{2^n} \right) \\ \sum_{i=1}^k d_i & \leq 1 + 2 \left(\frac{\sqrt{2}}{2} + 0.3827 \right) + 2\pi \left(\frac{1}{16} + \dots \right) + 2r_1 + 2r_2 \\ \sum_{i=1}^k d_i & \leq 1 + (0.7654 + 1.4143) + \frac{\pi}{4} + 2r_1 + 2r_2 \\ \sum_{i=1}^k d_i & < 3.9651 + 2r_1 + 2r_2 \leq 7.9651 \end{aligned}$$

□

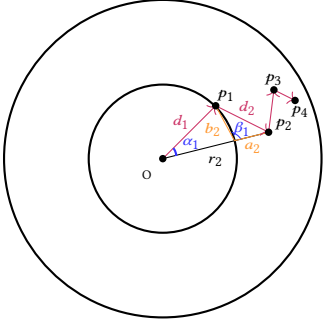


Figure 3: A path in the Arc-Strategy

Thus, using the Arc-Strategy, FTP in (\mathbb{R}^2, l_2) can be solved in at most $7.9651r$ time units, where r is the radius of the disk containing all the asleep robots, and the active robot is at the center.

The Arc-Strategy cannot be extended to \mathbb{R}^3 . In \mathbb{R}^2 , the strategy works by reducing the arc size each active robot covers, minimizing the distance d_i each robot travels to activate sleeping robots. This sum of distances converges efficiently in \mathbb{R}^2 , but does not converge in \mathbb{R}^3 . Thus, in the next section, we propose a different approach for \mathbb{R}^3 .

3.1.2 Ring-Strategy. We now introduce the Ring-Strategy. For a given center point, a ring is defined by its inner radius r_1 and outer radius r_2 , consisting of all points at a distance r , where $r_1 \leq r \leq r_2$, from the center. Suppose two awake robots, p_0 and p'_0 , are positioned at distance r_1 from the center. We focus on a ring with inner radius r_1 and outer radius 1, and outline a strategy to wake up the robots within the ring between r_1 and $r_2 = 1$. The Ring-Strategy is described as follows:

- Each of the two awake robots is responsible for waking up one half of the ring. The ring is split into two equal halves by drawing a diameter through $C(O, r_1)$ that passes through p_0 . Next, we explain the strategy for p_0 .
- Consider the projections of the points in p_0 's half of the ring onto the circle $C(O, r_1)$. The projection of a point p_j onto the circle $C(O, r)$ is the intersection of $\overline{Op_j}$ with the circle $C(O, r)$. Let p_1 be the point whose projection is closest to p_0 . Then, p_0 activates p_1 .
- In the next step, p_0 and p_1 divide their half of the ring into two smaller half-rings using the circle $C(O, \frac{r_1+1}{2})$. This creates one half-ring with an inner radius of r_1 and an outer radius of $\frac{r_1+1}{2}$, and another with an inner radius of $\frac{r_1+1}{2}$ and an outer radius of 1. Now, each of p_0 and p_1 is responsible for activating robots in their respective half-rings.
- As before, p_0 activates the point in its half-ring whose projection onto $C(O, r_1)$ is closest to it, while p_1 activates the point in its half-ring whose projection onto $C(O, \frac{r_1+1}{2})$ is closest to p_1 .
- In each subsequent step, the active point p_i in a half-ring with inner radius r_i and outer radius r_j identifies the point

in its half-ring whose projection onto $C(O, r_i)$ is closest and activates it. The half-ring is then split into two smaller half-rings: one with inner radius r_i and outer radius $\frac{r_i+r_j}{2}$, and the other with inner radius $\frac{r_i+r_j}{2}$ and outer radius r_j . At each step, the thickness of the half-ring is halved, with each robot responsible for waking up the robots in their respective half-ring.

- Meanwhile, similar to p_0 , p'_0 activates the robots in its half of the ring. By the end of this process, all robots within the ring are activated.

We now calculate the wake-up time for this strategy, specifically the time required to activate all the robots in a half-ring when the active robot is p_0 . Since there are two awake robots at the same initial position, p'_0 activates the robots in its own half in parallel. Therefore, the upper bound on the wake-up time is the same for both halves, and all robots will be awake within this time.

Let p_0, p_1, \dots, p_k represent a path in the wake-up tree of the Ring-Strategy, with the root at p_0 . We now define the following variables (see Figure 4):

$$\begin{aligned}
 d_i &= \|p_{i-1} - p_i\|_2 \\
 (r_i, \alpha_i) &= \text{polar coordinates of the location of } p_i \\
 a_i &= \text{the distance between } (r_{i-1}, \alpha_{i-1}) \text{ and } (r_i, \alpha_i) \\
 &\quad \text{i.e the distance between } p_i \text{ and image of } \\
 &\quad p_{i+1} \text{ on } C(O, r_i) \\
 b_i &= \text{the distance between } (r_{i-1}, \alpha_i) \text{ and } (r_i, \alpha_i) \\
 &\quad \text{i.e the distance between } p_i \text{ and image of } \\
 &\quad p_i \text{ on } C(O, r_{i-1})
 \end{aligned}$$

The total length of the path p_0, p_1, \dots, p_k is given by:

$$\sum_{i=1}^k d_i \leq \sum_{i=1}^k a_i + \sum_{i=1}^k b_i.$$

Since we split the half-ring associated with each active point at each step, we have $b_{i+1} \leq \frac{1-r_1}{2^{(i-1)}}$ for $i \geq 1$ and $\sum_{i=1}^k a_i \leq \pi$. Thus,

$$\sum_{i=1}^k d_i \leq \pi + b_1 + \sum_{i=2}^k \frac{1-r_1}{2^{(i-2)}}$$

$$\leq \pi + 1 - r_1 + 2(1 - r_1) = \pi + 3(1 - r_1)$$

Thus, we arrive at the following lemma.

LEMMA 3.2. *If in an instance of FTP all the robots are inside a ring with inner radius r_1 and outer radius of 1, and we have two awake robots in the boundary of $C(O, r_1)$, then Ring-Strategy activates all robots in $\pi + 3 - 3r_1$ time units.*

3.1.3 Our combined strategy for FTP in (\mathbb{R}^2, l_2) . We now propose a combined wake-up strategy based on the Arc-Strategy and Ring-Strategy. Given an instance of FTP in (\mathbb{R}^2, l_2) with an active robot at the center, let r_1 represent the distance from the center to the nearest asleep robot.

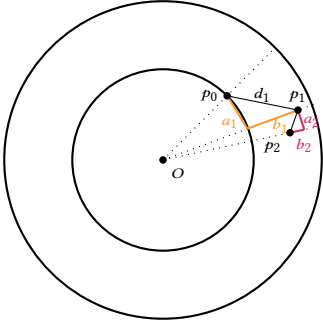


Figure 4: A path in the Ring-Strategy.

In the first step, p_0 moves toward p_1 , and now we have two awake robots. Divide the disk into two halves by drawing the diameter that passes through $\overline{Op_1}$. Now we have two active robots in the location of p_1 and each robot is responsible for one half.

Now, assume we want to wake the right half using p_0 . Let p_2 be the second nearest point to the center in the right half of the ring and r_2 represent its distance from the center. If $r_2 \leq 0.3627$, then using the Arc-Strategy, this half can be activated in $2r_1 + 2r_2 + 3.9651 \leq 4r_2 + 3.9651 \leq 5.4162$ time units, see the proof of Lemma 3.1. Otherwise, using the Ring-Strategy, p_0 moves $r_2 - r_1$ units in the direction of $\overrightarrow{Op_1}$. The half-ring on the right now has an inner radius of r_2 and an outer radius of 1. So, we can activate the robots in this half-ring in $r_1 + (r_2 - r_1) + 3 + \pi - 3r_2 < 6.1416 - 2 \times 0.3627 = 5.4162$ time units, see the proof of Lemma 3.2.

Similarly, let $p_{2'}$ be the second nearest point to the center in the left half of the disk. Again, based on the length of $r_{2'}$, we choose the best strategy for activating the robots in this half, which takes at most 5.4162 time units. Thus, the proof is complete.

3.2 Some examples

We show that for certain values of n , the wake-up ratio for n points is not achieved when the points are equally distributed on the unit circle by presenting specific instances and computing their makespan. Our first instance is a set of 5 asleep robots, the makespan of this instance is 3.530, see Figure 5. When five points are equally distributed on the cycle then the makespan is 3.351. Our second example is a set of 7 asleep robots, the makespan of this instance is 3.498, see Figure 5. When seven points are equally distributed on the cycle then the makespan is 3.431. The makespans of these instances are computed using a computer program by brute force search. Our code is publicly available online¹.

Furthermore, we generated over 10^5 random instances for different values of n and computed their makespans, leading to the following conjecture:

CONJECTURE 3.3. *The maximum makespan of n points is attained when they are on the boundary of the unit l_2 -disk.*

4 FTP IN (\mathbb{R}^3, l_1)

In this section, we study FTP in (\mathbb{R}^3, l_1) . The unit l_1 -ball in \mathbb{R}^3 , known as a cross-polytope, has all points on its surface exactly one

¹Makespan Calculator Code

| Point | θ | x | y |
|-------|----------|--------|--------|
| p_1 | 31.52 | 0.852 | 0.522 |
| p_2 | 101.98 | -0.207 | 0.978 |
| p_3 | 148.90 | -0.856 | 0.516 |
| p_4 | 195.69 | -0.962 | -0.270 |
| p_5 | 225.69 | -0.698 | -0.715 |
| p_6 | 271.63 | 0.028 | -0.999 |
| p_7 | 327.87 | 0.846 | -0.531 |

| Point | θ | x | y |
|-------|----------|--------|--------|
| p_1 | 17.37 | 0.954 | 0.298 |
| p_2 | 96.22 | -0.108 | 0.994 |
| p_3 | 174.42 | -0.995 | 0.097 |
| p_4 | 219.75 | -0.768 | -0.639 |
| p_5 | 299.27 | 0.488 | -0.872 |

Figure 5: Two instances of FTP in (\mathbb{R}^2, l_2) showing that the wake-up ratio is not attained for points equally distributed on the unit circle where θ is the angle from the positive x -axis in degree.

unit from the center, giving it a radius of 1. A cross-polytope with radius r is simply the unit ball scaled by r . Its diameter is $d = 2r$, meaning any two points within it are at most d units apart, and the distance from the center to any point is at most $r = \frac{d}{2}$.

In the FTP, the awake active robot starts at the center point $p_0 = (0, 0, 0)$. There are n asleep robots, each within at most one unit from p_0 in the l_1 norm. We begin by stating three lemmas.

LEMMA 4.1. *An awake robot at the center of a cross-polytope with radius r can wake any set of $n \leq 127$ asleep robots in $13r$ time units.*

PROOF. The initially awake robot, starting at the center of the cross-polytope, can reach any of the n sleeping robots in r time units. Once the first robot is awakened, both robots can each wake another sleeping robot in $d = 2r$ time units. This doubling process continues, with the number of awake robots doubling every d time units. After $6d = 12r$ time units, there will be at least $2 \times 2^6 = 128$ awake robots. Thus, using this strategy, up to 127 robots can be awakened in $13r$ time units. \square

LEMMA 4.2. *Given an awake robot at the center of a cross-polytope with radius r and $n \geq 128$ sleeping robots, seven of those robots can be awakened in $3r$ time units.*

PROOF. To prove this lemma, we first divide the cross-polytope with diameter d into six smaller cross-polytopes and eight pyramids (see Figure 6a). Each smaller cross-polytope has a diameter of $d' = \frac{d}{2} = r$ and a radius of $r' = \frac{r}{2}$. Their centers are located at $(\frac{1}{2}, 0, 0)$, $(-\frac{1}{2}, 0, 0)$, $(0, \frac{1}{2}, 0)$, $(0, -\frac{1}{2}, 0)$, $(0, 0, \frac{1}{2})$, and $(0, 0, -\frac{1}{2})$. The pyramids also have a diameter $d'' = \frac{d}{2} = r$, and three of their faces are shared with adjacent cross-polytopes, making each pyramid unique. As an example, the vertices of the pyramid in Figure 6b are $(0, 0, 0)$, $(\frac{1}{2}, \frac{1}{2}, 0)$, $(\frac{1}{2}, 0, \frac{1}{2})$, and $(0, \frac{1}{2}, \frac{1}{2})$. Since the pyramid is convex, its diameter equals the maximum distance between its vertices, $d'' = \frac{d}{2}$, which applies to all the pyramids.

With $n \geq 128$ sleeping robots and the cross-polytope divided into 14 regions, at least one region will have at least $\lceil \frac{128}{14} \rceil = 10$ asleep robots. The initially awake robot at the center can wake one robot in this region in r time units. These two robots can each wake another in $d' = \frac{d}{2}$ time units, doubling the number of awake

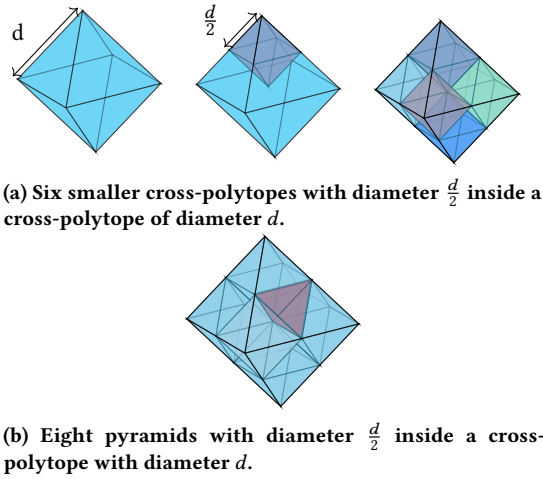


Figure 6: A cross-polytope partitioned into eight pyramids and six cross-polytopes with diameter $d' = \frac{d}{2}$.

robots every d' . Repeating this twice results in eight robots awake in $r + 2d' = 3r$ time units. \square

LEMMA 4.3. *A cross-polytope with diameter d can be fully covered by six smaller cross-polytopes, each with a diameter of $\frac{2d}{3}$.*

PROOF. Consider a cross-polytope with diameter d and radius $r = \frac{d}{2}$, centered at the origin. Now, take six smaller cross-polytopes, each with diameter $d' = \frac{2d}{3}$ and radius $r' = \frac{2r}{3}$, centered at $(\frac{r}{3}, 0, 0)$, $(-\frac{r}{3}, 0, 0)$, $(0, \frac{r}{3}, 0)$, $(0, -\frac{r}{3}, 0)$, $(0, 0, \frac{r}{3})$, and $(0, 0, -\frac{r}{3})$. See Figure 7.

Let $p = (x, y, z)$ be any point inside the larger cross-polytope, where $|x| + |y| + |z| \leq r$. We need to show that the distance from p to the center of one of the smaller cross-polytopes is at most $r' = \frac{2r}{3}$ in l_1 norm.

Without loss of generality, assume $|x| \geq |y|$ and $|x| \geq |z|$. We claim that if $x \geq 0$, p lies within the cross-polytope centered at $c_1 = (\frac{r}{3}, 0, 0)$, and if $x < 0$, it lies within the one centered at $c_2 = (-\frac{r}{3}, 0, 0)$. Assume $x \geq 0$. If $x \leq \frac{r}{3}$, then $|y| \leq |x| \leq \frac{r}{3}$ and $|z| \leq |x| \leq \frac{r}{3}$, and the distance is: $\|p - c_1\|_1 = |\frac{r}{3} - x| + |y| + |z| \leq \frac{r}{3} - x + 2|x| \leq \frac{2r}{3}$. If $x > \frac{r}{3}$, $\|p - c_1\|_1 = x - \frac{r}{3} + |y| + |z|$. Since $|y| + |z| \leq r - x$, it follows that $\|p - c_1\|_1 \leq x - \frac{r}{3} + r - x = \frac{2r}{3}$.

Similarly, if $x < 0$, we can show $\|p - c_2\|_1 \leq \frac{2r}{3}$. Thus, any point p inside the larger cross-polytope is contained in at least one of the six smaller cross-polytopes. \square

4.1 Our strategy for FTP in (\mathbb{R}^3, l_1)

Given a set of n asleep robots, our wake-up strategy is as follows: If $n \leq 127$, the awake robot at the origin can activate all robots within $13r$ time units, as shown in Lemma 4.1. If $n > 127$, by Lemma 4.2, we first wake up 7 robots in $3r$ time units. Then, we divide the cross-polytope into six smaller overlapping cross-polytopes, each with a diameter of $\frac{2d}{3}$, as described in Lemma 4.3. With eight robots awake, we assign six of them to one of the smaller cross-polytopes. Each robot reaches the center of its assigned cross-polytope in

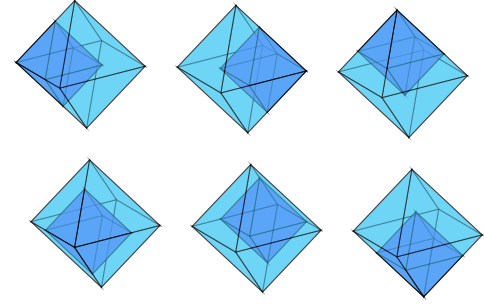


Figure 7: The cross-polytope with diameter d is covered by six overlapping cross-polytopes, each with a diameter of $\frac{2d}{3}$, fully covering all points of the larger cross-polytope.

at most $\frac{2d}{3}$ time units. Then, each robot wakes up the remaining robots in its section, taking at most $f(\frac{2d}{3})$ time units, where $f(d)$ is the time needed to wake all robots in a region of diameter d . If a robot belongs to multiple cross-polytopes, it can be assigned to any one of them. If a cross-polytope has fewer than 128 asleep robots, they can be woken up in $\frac{13d}{4}$ time units as per Lemma 4.1. Thus, the recursive formula for computing $f(d)$ is:

$$f(d) \leq \frac{3d}{2} + \frac{2d}{3} + f\left(\frac{2d}{3}\right) = \frac{13d}{6} + f\left(\frac{2d}{3}\right)$$

Which yields: $f(d) \leq \frac{13d}{6} \times \left(1 + \frac{2}{3} + \frac{4}{9} + \dots\right) = \frac{13d}{2}$. For the unit l_1 -ball where $d = 2$, the upper bound for the total time is 13 units.

5 WAKE-UP RATIO FOR ROBOTS ON THE BOUNDARY OF THE UNIT l_2 -BALL IN \mathbb{R}^3

In this section, we study FTP in (\mathbb{R}^3, l_2) for a special case where the asleep robots are on the boundary of the unit l_2 -ball.

We believe this version of FTP is significant, as we hypothesize the maximal makespan occurs when all asleep robots are on the boundary. This hypothesis is based on our observation that, in numerous random instances for various values of n in (\mathbb{R}^3, l_1) and (\mathbb{R}^3, l_2) , the maximum makespan was achieved when the asleep robots were on the boundary.

This scenario is also relevant to real-world applications like communication and transportation, where, similar to the unit ball in (\mathbb{R}^3, l_2) , key locations on Earth are often on its surface. This leads us to a variant called surface-FTP, where all robots, including the initially active one, are located on the surface of the l_2 -ball in \mathbb{R}^3 , and distances are measured using the geodesic (shortest arc) distance.

Our approach for these FTP versions is to project the points on the boundary of the unit l_2 -ball in \mathbb{R}^3 onto an l_2 -disk with radius $\frac{\pi}{2}$, ensuring the Euclidean distance between projected points is less than their original Euclidean distances on the boundary of l_2 -ball.

5.1 Mapping

We begin by cutting the unit l_2 -ball in \mathbb{R}^3 into two hemispheres and mapping one hemisphere to a disk. For this disk, we generate a wake-up tree using the method from Section 3.1. The wake-up tree

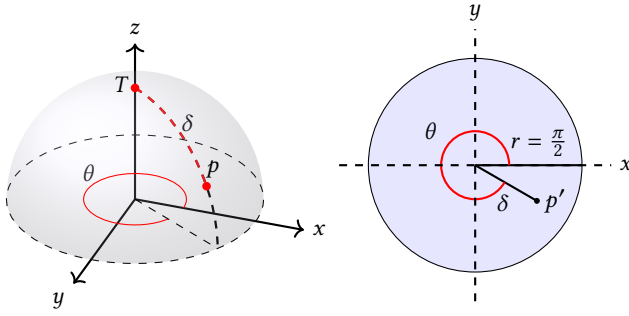


Figure 8: Every point on the surface of a unit hemisphere can be represented by δ and θ . The point $p = (0.433, 0.75, 0.5)$ is represented by $\theta = -30^\circ = 330^\circ$ and $\delta = 1.047$. The mapping of p onto the unit l_2 -disk results in p' with coordinates (δ, θ) in polar coordinates.

is then extended to coordinate the wake-up process for the robots on the boundary. We explain our mapping in the following.

Consider the upper hemisphere of a unit l_2 -ball with $T = (0, 0, 1)$. Any point $p = (x, y, z)$ on the surface of the hemisphere can be uniquely represented by the pair (δ, θ) , where $\delta = \arccos(z)$ is the geodesic distance from T to p (the shortest path along the hemisphere), and θ is the angle of p 's projection onto the xy -plane relative to the positive x -axis. This is illustrated in Figure 8.

Mapping \mathcal{M} : We define a mapping \mathcal{M} that maps each point p with pair (δ, θ) on the hemisphere to a point $p' = (\delta, \theta)$ in a disk with radius $\frac{\pi}{2}$ using polar coordinates. In this mapping, $T = (0, 0, 1)$ is mapped to $(0, 0)$ in the l_2 -disk, with $\theta \in [0, 2\pi)$ and $\delta \in [0, \frac{\pi}{2}]$. The boundary of the hemisphere is mapped to a disk of radius $\frac{\pi}{2}$, as shown in Figure 8.

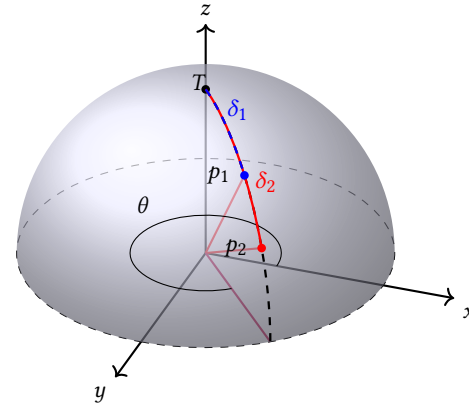
We now prove that for any two points p_1 and p_2 on the hemisphere, the Euclidean distance between them is less than or equal to the Euclidean distance between their mapped points $p'_1 = \mathcal{M}(p_1)$ and $p'_2 = \mathcal{M}(p_2)$. This is formally stated in the following lemma.

LEMMA 5.1. *The mapping \mathcal{M} ensures that for any two points p_1 and p_2 on the hemisphere, mapped to $p'_1 = \mathcal{M}(p_1)$ and $p'_2 = \mathcal{M}(p_2)$ on the disk, $\|p_1 - p_2\|_2 \leq \|p'_1 - p'_2\|_2$.*

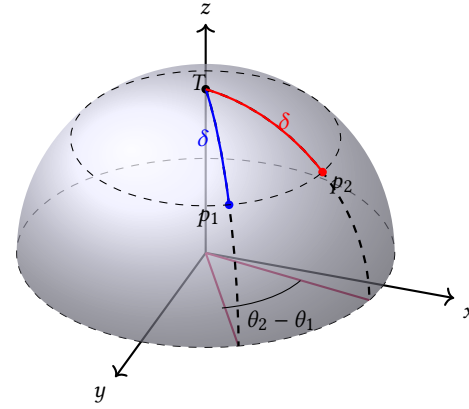
PROOF. We have three cases for $p_1 = (\theta_1, \delta_1)$ and $p_2 = (\theta_2, \delta_2)$.

Case 1: $\theta_1 = \theta_2 = \theta$, (see Figure 9a). In this case, the geodesic distance between p_1 and p_2 is given by $|\delta_1 - \delta_2|$, and $\|p'_1 - p'_2\|_2$ is also equal to $|\delta_1 - \delta_2|$. Since p_1 and p_2 lie on the same meridian, $\|p_1 - p_2\|_2$ is equal to $2 \sin\left(\frac{|\delta_1 - \delta_2|}{2}\right)$. Using the inequality $x \leq \sin(x)$ for $x \in [0, \frac{\pi}{2}]$, we get $\|p_1 - p_2\|_2 \leq \|p'_1 - p'_2\|_2$.

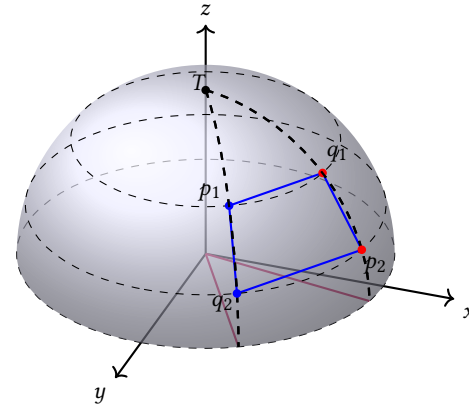
Case 2: $\delta_1 = \delta_2 = \delta$, (see Figure 9b). Consider the circle passing through p_1 and p_2 that is parallel to the xy -plane, with its center on the z -axis. The angle between p_1 , the center of the circle, and p_2 is $|\theta_1 - \theta_2|$, and the radius of the circle is $\sin(\delta)$. Consequently, the geodesic distance between p_1 and p_2 is $|\theta_1 - \theta_2| \sin(\delta)$, while $\|p_1 - p_2\|_2$ is $2 \sin\left(\frac{|\theta_1 - \theta_2|}{2}\right) \sin(\delta)$, and $\|p'_1 - p'_2\|_2$ is $2 \sin\left(\frac{|\theta_1 - \theta_2|}{2}\right)$. Therefore, we have $\|p_1 - p_2\|_2 \leq \|p'_1 - p'_2\|_2$.



(a) In this case, $\theta_1 = \theta_2 = \theta$, and δ_1 and δ_2 represent the geodesic distances of p_1 and p_2 from the point $T = (0, 0, 1)$.



(b) Case 2: In this case, $\delta_1 = \delta_2 = \delta$, where δ is the geodesic distance of both p_1 and p_2 from $T = (0, 0, 1)$.



(c) Case 3: In this case, $\theta_1 \neq \theta_2$ and $\delta_1 \neq \delta_2$. The geodesic distance of p_1 and q_2 from $T = (0, 0, 1)$ is the same, and the geodesic distance of q_1 and p_2 from T is also the same. A plane passes through the points p_1, p_2, q_1 , and q_2 , forming a trapezoid. The Euclidean distance between p_1 and p_2 is the length of the diameter of this trapezoid.

Case 3: $\theta_1 \neq \theta_2$ and $\delta_1 \neq \delta_2$, (see Figure 9c). Let $q_1 = (\delta_1, \theta_2)$ and $q_2 = (\delta_2, \theta_1)$. Along with p_1 and p_2 , these four points form an isosceles trapezoid. And the length of the diagonal of this isosceles trapezoid is equal to $\|p_1 - p_2\|_2$. The first case implies, $\|p_1 - q_2\|_2 = \|p_2 - q_1\|_2 = 2 \sin(\frac{|\delta_2 - \delta_1|}{2})$ and the second case implies $\|p_1 - q_1\|_2 = \|p_2 - q_2\|_2 = 2 \sin(\delta_1) \sin(\frac{|\theta_1 - \theta_2|}{2})$ and $\|p_2 - q_2\|_2 = 2 \sin(\delta_2) \sin(\frac{|\theta_1 - \theta_2|}{2})$. Thus,

$$\|p_1 - p_2\|_2 = \sqrt{4 \sin^2(\frac{|\delta_1 - \delta_2|}{2}) + 4 \sin(\delta_1) \sin(\delta_2) \sin^2(\frac{|\theta_1 - \theta_2|}{2})}.$$

On the other hand,

$$\|p'_1 - p'_2\|_2 = \sqrt{\delta_1^2 + \delta_2^2 - 2\delta_1\delta_2 \cos(|\theta_1 - \theta_2|)}.$$

Now, we show that $\|p_1 - p_2\|_2 \leq \|p'_1 - p'_2\|_2$. Consequently we need to show,

$$\sin^2(\frac{|\delta_1 - \delta_2|}{2}) + \sin(\delta_1) \sin(\delta_2) \sin^2(\frac{|\theta_1 - \theta_2|}{2}) \leq (\frac{\delta_1}{2} - \frac{\delta_2}{2})^2 + \frac{\delta_1\delta_2}{2} - \delta_1\delta_2 \frac{\cos(|\theta_1 - \theta_2|)}{2}$$

$\sin(x) \leq x$ implies that $\sin^2(\frac{|\delta_1 - \delta_2|}{2}) \leq (\frac{\delta_1}{2} - \frac{\delta_2}{2})^2$. It remains to prove that:

$$\sin(\delta_1) \sin(\delta_2) \sin^2(\frac{|\theta_1 - \theta_2|}{2}) \leq (\delta_1\delta_2)(\frac{1}{2} - \frac{\cos(|\theta_1 - \theta_2|)}{2}).$$

Since $\sin(\delta_1) \sin(\delta_2) \leq \delta_1\delta_2$ and using the identity $2 \sin^2(\theta) = 1 - \cos(2\theta)$, the inequality holds, completing the proof. \square

5.2 Our strategy, when asleep robots are on the boundary on the unit l_2 -ball

For an instance of FTP in (\mathbb{R}^3, l_2) , where the asleep robots are on the boundary and the initially active robot is at the center of the ball, we propose the following strategy. We split the sphere into two hemispheres. Without loss of generality, assume $T = p_1 = (0, 0, 1)$, and that the hemisphere containing p_1 has $n_1 \geq \frac{n}{2}$ robots. We move p_0 toward p_1 in 1 time unit. Then, using the mapping \mathcal{M} , we map the robots on the boundary of the hemisphere containing p_1 onto an l_2 -disk with radius $\frac{\pi}{2}$. We have n_1 asleep robots and one awake robot at the origin of the disk. Using the Arc-Strategy from Lemma 3.1, we can wake all the robots in the disk in $\frac{\pi}{2} \times 3.9651 + 2r_1 + 2r_2$ time units, where r_1 is the distance of the nearest point to the origin and r_2 depends on the path and we have $r_2 \leq 1$.

. Since both p_0 and p_1 are at the origin, $r_1 = 0$, and all robots on the disk can be woken in $\frac{\pi}{2} \times 5.9651$ time units.

Thus, by Lemma 5.1 and the Arc-Strategy, after 1 time unit (for moving p_0 toward p_1) plus $\frac{\pi}{2} \times 5.9651$ time units, all robots in the upper hemisphere are awake, giving us $n_1 + 1$ awake robots. Since $n_1 \geq \frac{n}{2}$, each awake robot on the boundary can wake a robot in the lower hemisphere within 2 time units. Therefore, all robots can be woken in $3 + \frac{\pi}{2} \times 5.9651 \approx 12.37$ time units.

5.3 Our strategy for surface-FTP

We apply the mapping approach to solve the surface-FTP. Points on the hemisphere's boundary are mapped onto a disk with radius $\frac{\pi}{2}$ using the mapping \mathcal{M} . In Lemma 5.1, we proved that the Euclidean distance between two points p_1 and p_2 on the hemisphere is:

$$\|p_1 - p_2\|_2 = \sqrt{4 \sin^2(\frac{|\delta_1 - \delta_2|}{2}) + 4 \sin(\delta_1) \sin(\delta_2) \sin^2(\frac{|\theta_1 - \theta_2|}{2})}.$$

So, the geodesic distance between them is:

$$2 \arcsin(\sqrt{\sin^2(\frac{|\delta_1 - \delta_2|}{2}) + \sin(\delta_1) \sin(\delta_2) \sin^2(\frac{|\theta_1 - \theta_2|}{2})})$$

On the other hand, the Euclidean distance between their mapping $\mathcal{M}(p_1) = p'_1$ and $\mathcal{M}(p_2) = p'_2$ is:

$$\|p'_1 - p'_2\|_2 = \sqrt{\delta_1^2 + \delta_2^2 - 2\delta_1\delta_2 \cos(|\theta_1 - \theta_2|)}.$$

We claim that the geodesic distance between points p_1 and p_2 is less than or equal to the Euclidean distance between the corresponding points p'_1 and p'_2 . To support this, we conducted a computational analysis, calculating distances for all combinations of $\delta_1 = \frac{\pi}{2}\epsilon \times i$, $\delta_2 = \frac{\pi}{2}\epsilon \times j$, and $\theta_1 - \theta_2 = \pi\epsilon \times k$, with i, j, k ranging from 0 to $\frac{1}{\epsilon}$. The code, available online ², was run with $\epsilon = 0.001$, and in all cases, the geodesic distance between p_1 and p_2 was less than or equal to $\|p'_1 - p'_2\|_2$. ³ Thus, the wake-up time for robots on the hemisphere is less than or equal to the wake-up time for robots on the disk. This means that a wake-up tree for robots on the disk also works for robots on the hemisphere, with a makespan that is no greater.

Now we focus on surface-FTP. Initially, we have an awake robot, p_0 , on the surface of a sphere, and this robot must wake n asleep robots. The distances between them are geodesic distances on the surface. We divide the sphere into two halves, with p_0 located at the top of the upper half.

- We move p_0 to the nearest point, p_1 , in ρ_1 time units, where ρ_1 is the geodesic distance between p_0 and p_1 . Now, with two awake robots, p_1 handles robots in the upper hemisphere, while p_0 handles the robots in the lower hemisphere.
- The strategy for waking up the robots in the upper hemisphere: Using the mapping \mathcal{M} , we map the points on the upper hemisphere onto a disk with radius $\frac{\pi}{2}$, placing the awake robot at $p_1 = (\rho_1, \theta_1)$. We then move p_1 to the center of the disk. The remaining robots on the disk can be awakened in $\frac{\pi}{2} \times 5.4162$ time units using Theorem 2.1. Therefore, all robots on the upper hemisphere are awakened in at most $2\rho_1 + \frac{\pi}{2} \times 5.4162$ time units. Since $\rho_1 \leq \frac{\pi}{2}$, the makespan for the upper hemisphere is at most 11.65 time units.
- The strategy for waking up the robots in the lower hemisphere: First, p_0 moves to p_1 in ρ_1 time units, then to p'_1 in $\pi - \rho_1 - \rho'_1$ time units, where p'_1 is the nearest point to the lowest point in the hemisphere. Using the mapping \mathcal{M} , we map the lower hemisphere onto a disk with radius $\frac{\pi}{2}$. Now, with two awake robots, p_0 and p'_1 , within ρ'_1 of the disk's center, the remaining robots can be awakened in $\frac{\pi}{2} \times 5.4162$ time units using Theorem 2.1. The total makespan for the lower hemisphere is $\pi - \rho'_1 + \frac{\pi}{2} \times 5.4162 \leq 11.65$.

Thus, the wake-up ratio for surface-FTP is at most 11.65.

²Geodesic to Euclidean Distance Ratio Calculator Code

³A detailed computation of the derivatives of the distance functions can be found in our code.

REFERENCES

- [1] Zachary Abel, Hugo A Akitaya, and J Yu. 2017. Freeze tag awakening in 2d is NP-hard. In *27th Fall Workshop on Computational Geometry*. 105–107.
- [2] Susanne Albers and Monika Rauch Henzinger. 1997. Exploring Unknown Environments. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4–6, 1997*, Frank Thomson Leighton and Peter W. Shor (Eds.). ACM, 416–425. <https://doi.org/10.1145/258533.258630>
- [3] Susanne Albers, Klaus Kursawe, and Sven Schuierer. 2002. Exploring Unknown Environments with Obstacles. *Algorithmica* 32, 1 (2002), 123–143. <https://doi.org/10.1007/S00453-001-0067-X>
- [4] Esther M Arkin, Michael A Bender, Sándor P Fekete, Joseph SB Mitchell, and Martin Skutella. 2006. The freeze-tag problem: how to wake up a swarm of robots. *Algorithmica* 46 (2006), 193–221.
- [5] Esther M. Arkin, Michael A. Bender, Sándor P. Fekete, Joseph S. B. Mitchell, and Martin Skutella. 2002. The freeze-tag problem: how to wake up a swarm of robots. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6–8, 2002, San Francisco, CA, USA*, David Eppstein (Ed.). ACM/SIAM, 568–577. <http://dl.acm.org/citation.cfm?id=545381.545457>
- [6] Esther M. Arkin, Michael A. Bender, and Dongdong Ge. 2003. Improved approximation algorithms for the freeze-tag problem. In *SPAA 2003: Proceedings of the Fifteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, June 7–9, 2003, San Diego, California, USA (part of FCRC 2003)*, Arnold L. Rosenberg and Friedhelm Meyer auf der Heide (Eds.). ACM, 295–303. <https://doi.org/10.1145/777412.777465>
- [7] Nicolas Bonichon, Arnaud Casteigts, Cyril Gavoille, and Nicolas Hanusse. 2024. Freeze-Tag in L_1 Has Wake-Up Time Five with Linear Complexity. In *38th International Symposium on Distributed Computing (DISC 2024) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 319)*, Dan Alistarh (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 9:1–9:16. <https://doi.org/10.4230/LIPIcs.DISC.2024.9>
- [8] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and Saeed Odak. 2024. Euclidean Freeze Tag Problem. In *The Canadian Conference on Computational Geometry (CCCG 2024)*, St. Catharines, Canada. <https://hal.science/hal-04803161>
- [9] Alfred M Bruckstein, Colin I Mallows, and Israel A Wagner. 1997. Probabilistic pursuits on the grid. *The American mathematical monthly* 104, 4 (1997), 323–343.
- [10] Adrian Dumitrescu, Ichiro Suzuki, and Masafumi Yamashita. 2002. High Speed Formations of Reconfigurable Modular Robotic Systems. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, May 11–15, 2002, Washington, DC, USA*. IEEE, 123–128. <https://doi.org/10.1109/ROBOT.2002.1013349>
- [11] Douglas W. Gage. 2000. Minimum-resource distributed navigation and mapping. In *Mobile Robots XV and Telemanipulator and Telepresence Technologies VII, Boston, MA, USA, November 5, 2000 (SPIE Proceedings, Vol. 4195)*, Howie Choset, Douglas W. Gage, and Matthew R. Stein (Eds.). SPIE, 96–103. <https://doi.org/10.1117/12.417293>
- [12] Christian Icking, Thomas Kamphans, Rolf Klein, and Elmar Langetepe. 2000. Exploring an Unknown Cellular Environment. In *EuroCG*. 140–143.
- [13] Matthew P Johnson. 2017. Easier hardness for 3D freeze-tag. *Proc. 27th Fall Worksh. Comp. Geom.(FWCG)* (2017).
- [14] Jochen Könnemann, Asaf Levin, and Amitabh Sinha. 2005. Approximating the Degree-Bounded Minimum Diameter Spanning Tree Problem. *Algorithmica* 41, 2 (2005), 117–129. <https://doi.org/10.1007/S00453-004-1121-2>
- [15] Lehlilton Lelis Chaves Pedrosa and Lucas de Oliveira Silva. 2023. Freeze-Tag is NP-hard in 3D with L_1 distance. In *Proceedings of the XII Latin-American Algorithms, Graphs and Optimization Symposium, LAGOS 2023, Huatulco, Mexico, September 18–22, 2023 (Procedia Computer Science, Vol. 223)*, Cristina G. Fernandes and Sergio Rajsbaum (Eds.). Elsevier, 360–366. <https://doi.org/10.1016/J.PROCS.2023.08.248>
- [16] Kazuo Sugihara and Ichiro Suzuki. 1996. Distributed algorithms for formation of geometric patterns with many mobile robots. *J. Field Robotics* 13, 3 (1996), 127–139. [https://doi.org/10.1002/\(SICI\)1097-4563\(199603\)13:3<3C127::AID-ROB1%3E3.0.CO;2-U](https://doi.org/10.1002/(SICI)1097-4563(199603)13:3<3C127::AID-ROB1%3E3.0.CO;2-U)
- [17] Ichiro Suzuki and Masafumi Yamashita. 1999. Distributed Anonymous Mobile Robots: Formation of Geometric Patterns. *SIAM J. Comput.* 28, 4 (1999), 1347–1363. <https://doi.org/10.1137/S009753979628292X>
- [18] Israel A. Wagner, Michael Lindenbaum, and Alfred M. Bruckstein. 1998. Efficiently Searching a Graph by a Smell-Oriented Vertex Process. *Ann. Math. Artif. Intell.* 24, 1–4 (1998), 211–223. <https://doi.org/10.1023/A:1018957401093>
- [19] Israel A. Wagner, Michael Lindenbaum, and Alfred M. Bruckstein. 1999. Distributed covering by ant-robots using evaporating traces. *IEEE Trans. Robotics Autom.* 15, 5 (1999), 918–933. <https://doi.org/10.1109/70.795795>
- [20] Ehsan Najafi Yazdi, Alireza Bagheri, Zahra Moezkarimi, and Hamidreza Keshavarz. 2015. An $O(1)$ -approximation algorithm for the 2-dimensional geometric freeze-tag problem. *Inf. Process. Lett.* 115, 6–8 (2015), 618–622. <https://doi.org/10.1016/J.IPL.2015.02.011>