Freeze-Tag with Return

Nicolas Bonichon
LaBRI, University of Bordeaux, CNRS, Bordeaux INP, Talence, France

Cyril Gavoille
LaBRI, University of Bordeaux, CNRS, Bordeaux INP, Talence, France

Nicolas Hanusse
LaBRI, University of Bordeaux, CNRS, Bordeaux INP, Talence, France

Gabriel Le Bouder
LIP6, Sorbonne University, Paris, France

Taïssir Marcé
LaBRI, University of Bordeaux, CNRS, Bordeaux INP, Talence, France

Nils Morawietz
LaBRI, University of Bordeaux, CNRS, Bordeaux INP, Talence, France
Institute of Computer Science, Friedrich Schiller University Jena, Germany

Abstract

In the standard Freeze-Tag Problem (FTP), an initially awake robot (the *source*) is in charge of waking up a swarm of sleeping robots by moving towards them, given that all the awake robots can participate in the awakening process. The goal is to minimize the makespan to wake up all robots assuming they move at unit speed. In this paper we introduce the *Freeze-Tag-with-Return Problem* (FTRP) variant, where the robots must eventually return to their initial positions.

In the Euclidean plane with n sleeping robots lying on the unit disk centered at the initial position of the source, we show a non-trivial relationship between FTP and FTRP by proving that the difference between the optimal makespan of both problems never exceeds 1.959, and is at least 1.732 in the worst-case. We also present several upper and lower bounds on the optimal makespan. In particular, we show that if the sleeping robots are in convex positions, then the optimal makespan is at most $2 + 2\sqrt{2}$, which is achieved by some instances.

From an algorithmic point-of-view, we present single-exponential algorithms for general distance functions. In metric spaces, these algorithms are asymptotically optimal under the ETH, which we show via an NP-hardness reduction on unweighted graphs.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Freeze-Tag Problem, sleeping robots, metric spaces

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.43

Related Version *Full Version*: `https://arxiv.org/abs/2606.21985` [6]

Funding The authors are supported by the French ANR, project ANR-22-CE48-0001 (TEMPOGRAL).

1 Introduction

The Freeze-Tag Problem (FTP) was introduced by [3] and describes the following problem: given a set of n *sleeping* robots and *one initial awake* robot, the goal is to wake up all robots as fast as possible. In the context of energy-constrained autonomous robotics, the sleep state provides the necessary downtime for energy harvesting processes. What makes the problem interesting is that an awake robot must move to a sleeping robot to wake it up. Furthermore, once several robots are awake, they can collaborate to wake up the remaining sleeping robots.

The *makespan* is defined as the time required to wake up all robots. Robots are assumed to move at a constant speed and thus the time can also be expressed as the total length of a longest trajectory followed by a sequence of awake robots to wake up a sleeping robot.

In the following, we propose a new version of this problem, in which the robots, once awakened and after participating in the wake-up process, must return to their initial positions. The objective is now to minimize the time, still referred to as the *makespan*, required for all robots to be both awake and back to an initial position. We call this problem the Freeze-Tag-with-Return Problem (FTRP). We introduce two variants: in the first, called FTRPID, each robot has an *identifier* and must return exactly to its own initial position, while in the second, called FTRPA, the robots are *anonymous* and only need to end at some arbitrary initial position, provided that each initial position contains a robot at the end.

Our problem is motivated by scenarios where robots, once awakened and their wake-up task is completed, must come back to perform specific tasks at their initial positions: inspection of infrastructures or environment (fire detection), cleaning up an area, collecting or delivering objects like batteries, or performing mechanical maintenance. When robots are interchangeable, the FTRPA variant is enough, whereas when robots have specific roles or skills, the FTRPID variant is necessary.

Note that for any configuration of initial positions, the optimal makespan for the FTRP (both variants) is at most the makespan of the FTP plus the largest distance D between any two initial positions (to allow each robot to come back once awakened), and is at least the optimal makespan of the FTP plus the smallest distance between any two initial positions (since the robot awakening the last sleeping robot needs to go to another initial position). For the graph setting, the *overhead due to returns* can reach D. Take an unweighted star of 3 leaves containing sleeping robots and the initial awake robot located at the center. The makespan of FTP and FTRP for this instance are respectively 3 and 5. In this work we consider the question of whether these bounds for the overhead are tight.

State of the art. The FTP has received significant attention since its introduction [3]. Several complexity results have been obtained, showing that the problem is NP-hard, first for metrics induced by weighted star graphs and unweighted trees [4], then for Euclidean metrics in dimension 3 [13], ℓ_p metrics in dimension 3 for $p > 1$ [9], for the Euclidean metric in dimension 2 [1], for ℓ_1 metric in dimension 3 [14], and recently claimed for ℓ_1 metric in dimension 2 [15]. On the other hand, there is a known PTAS for FTP in $(\mathbb{R}^d, \ell_p)$ that runs in time $O(n \log n) + 2^{(d/\varepsilon)^{O(d)}}$ [4], subject to $\varepsilon \leqslant \varepsilon_d$, where ε_d depends on the dimension d.

A key parameter for an FTP instance is its *radius*, the maximum distance between the initial robot and any sleeping robot. Obviously, the radius is a lower bound for the optimal makespan. In the Euclidean plane, the worst-case optimal makespan for an instance of radius 1 with n sleeping robots is denoted by $\gamma(n)$. To warm up, we can observe that $\gamma(1) = 1$ and $\gamma(2) = \gamma(3) = 3$ (take sleeping robots being diametrically opposed). We call the *wake-up constant* $\overline{\gamma} = \sup_{n \in \mathbb{N}} \gamma(n)$. As the authors of [5] proved, $\overline{\gamma} \geqslant \gamma(4) = 1 + 2\sqrt{2} \approx 3.83$, and they have conjectured that this value is the worst possible, that is, $\overline{\gamma} = \gamma(4)$. Actually they have proposed a general conjecture claiming that the wake-up constant is reached with $n = 4$, in any norm. For instance, the conjectured wake-up constant for ℓ_p is $1 + 2^{1+\max\{1/p, 1-1/p\}}$, which holds for $p \in \{1, 2, \infty\}$. It turns out that $\gamma(n)$ is much smaller whenever n is large, as they showed that $\gamma(n) = 3 + O(1/\sqrt{n}\,)$.

Polynomial-time approximation algorithms have been proposed, leading to the following upper bound on the wake-up constant $\overline{\gamma}$: 57 [4], 10.07 [17], 7.07 [5], 4.63 [7], 4.31 [2]. In the case of the plane equipped with the ℓ_1 metric, a linear time algorithm has been proposed with a makespan bounded by 5 times the radius, which is best possible [5].

FTP has also been studied in online and distributed contexts. In the online context, sleeping robots appear as time progresses [8, 12]. In the distributed context, robots do not have knowledge of the other robots' positions. They need to explore regions with a bounded vision and also limited communication capabilities (two awake robots can only communicate if they are within a certain distance) [11].

Table 1 Lower and upper bounds for the minimum makespan in the Euclidean plane.

	$\gamma(n)$	$\gamma_{\text{ID}}(n)$	**(our results)**
Lower Bound $n = 4$	$1 + 2\sqrt{2} \approx 3.823$	$2 + 2\sqrt{2} \approx 4.823$	[Theorem 1]
Lower Bound for any $n \geqslant 5$	3	4.672*	[Theorem 1]
UB (Convex case) for any n	$1 + 2\sqrt{2}$	$2 + 2\sqrt{2} \approx 4.823$	[Theorem 2]
Upper Bound n large	$3 + O(1/\sqrt{n}\,)$	$4.89 + O(1/\sqrt{n}\,)$	[Theorem 4]
Upper Bound for any n	4.31	$6.269 = 4.31 + 1.959$	[Theorem 5]

*We get a different bound for FTRPA: $\forall n \geqslant 5$, $\gamma_{\text{A}}(n) \geqslant 1 + \frac{1}{4}(7^{1/4} + 7^{3/4})\sqrt{6} \approx 4.631$.

Contributions. Our goal is to compute and minimize the makespan for the FTRP. As for FTP, $\overline{\gamma_{\text{ID}}} = \sup_{n\in\mathbb{N}} \gamma_{\text{ID}}(n)$ is the *wake-up and return constant* for the FTRPID and we use notation $\overline{\gamma_{\text{A}}}$ similarly for FTRPA. $\gamma_{\text{ID}}(n)$ is the worst-case makespan for n sleeping robots within a unit disk centered at the source in the FTRPID model.

Our contributions can be summarized as follows:

(1) In Section 3, we present several lower and upper bounds on the makespan of FTRPID and FTRPA in the Euclidean plane. In particular, we show that the overhead due to returns is at most 1.959, and thus never exceeds $D = 2$. We also remark that some distributions prove that the overhead must be at least $\sqrt{3} > 1.732$.

(2) In Section 4, we show that computing the optimal makespan is NP-hard in the (unweighted) graph model setting for both variants FTRPID and FTRPA.

(3) In Section 5, we present three $2^{O(n)}$-time algorithms to compute the optimal makespan for the FTP and both variants of the FTRP for any non-negative distance function, therefore including Euclidean and metric spaces and weighted graph instances.

More precisely, in the Euclidean plane, Theorem 1 states that $\overline{\gamma_{\text{ID}}}$ and $\overline{\gamma_{\text{A}}}$ are both at least $2 + 2\sqrt{2}$ and that this lower bound is achieved for $n = 4$ (see Corollary 3). We show in Theorem 2 that this is also an upper bound for any distribution of sleeping robots in convex positions. Even if there are some pairs of robots at a pairwise distance $D = 2$, surprisingly, we show in Theorem 5 that the optimal makespan of any instance of FTP and FTRPID have a difference that never exceeds 1.959.

Table 1 summarizes the different known bounds for $\gamma(n)$, and the new ones we get for $\gamma_{\text{ID}}(n)$. Note that each upper bound for FTRPID is also an upper bound for FTRPA and that each lower bound for FTRPA is also a lower bound for FTRPID. Although we exhibit some specific distributions having different optimal makespan for FTRPA and FTRPID, we are unable to prove a gap between the wake-up return constants $\overline{\gamma_{\text{A}}}$ and $\overline{\gamma_{\text{ID}}}$, which deal with worst-case distributions.

In Theorem 9, we show that even if we restrict the metric space to unweighted graphs of bounded degree, FTRPID and FTRPA are both NP-hard. Moreover, unless the ETH fails, no algorithm can solve any of these problems in time $2^{o(n)}$.

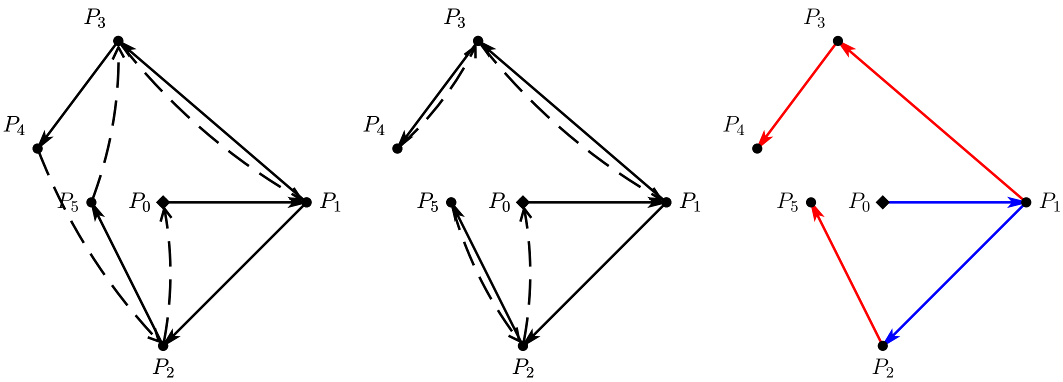
Complementarily, in Section 5, we provide algorithms returning the optimal makespan for any instance of FTP, FTRPID, and FTRPA in time $O(3^n n^2)$, $O(3^n n^3)$, and $O(9^n n^{3/2})$ respectively. These algorithms have been implemented, and used to verify our lower bounds for small n.

2 Preliminaries

In this section we give formal definitions of the key concepts introduced above. An instance is defined by a multi-set of points $\mathcal{P} = \{P_0, \dots, P_n\}$ within a metric space (X, d), each point P_i being associated with a robot r_i initially located at P_i. Depending on the context, P_r also stands for the initial position of robot r. We assume that r_0 at P_0 is the initially awake robot, and denote by $\mathcal{R}$ the set of all robots.

The solution of an FTP-instance can be encoded by a *wake-up tree* $\mathsf{T} = (\mathcal{P}, \mathcal{E}_\mathsf{T})$, a weighted tree rooted at P_0 that spans $\mathcal{P}$. The existence of an edge $(P_i, P_j) \in \mathcal{E}_\mathsf{T}$ means that robot r_j is woken up by an awake robot previously located at node P_i. In other words, r_j is woken up by r_i or by the robot r' that moved to P_i and woke up r_i. As a result, the root has at most one child, and any node has up to two children.

Arcs of $\mathcal{E}_\mathsf{T}$ are called *wake-up arcs* and are oriented from parent to child. The weight of arcs are given by the (metric) distance d between the positions of their endpoints, which can be zero since we allow co-located robots.



(a) **(b)** **(c)**

Figure 1 (a) A FTRPA wake-up strategy (self-loops on P_4 and P_5 are omitted). (b) A FTRPID wake-up strategy: r_0 wakes up $\{r_1, r_2\}$ and comes back, r_1 wakes up $\{r_3, r_4\}$ and comes back, r_2 wakes up r_5 and comes back. The robots r_3, r_4 and r_5 do not move. (c) The same FTRPID wake-up strategy is represented with a bicolored tree: the return arcs are between endpoints of maximal monochromatic sub-paths.

Given a wake-up tree T, the *wake-up time* of a robot r_i (resp. position P_i) is denoted $\mathrm{wu}_\mathsf{T}(r_i)$ (resp. $\mathrm{wu}_\mathsf{T}(P_i)$) and corresponds to the weighted length of the path from P_0 to P_i in T. Thus the makespan of a wake-up strategy encoded by T is the weighted depth of T.

A solution to an instance $\mathcal{P}$ of FTRPA can be encoded by a tree being an extension of the wake-up tree duplicating the initial positions and adding the clones as leaves. For convenience, we represent a solution with a pair $(\mathsf{T}, \mathcal{E}_\mathrm{R})$ where $\mathsf{T} = (\mathcal{P}, \mathcal{E}_\mathsf{T})$ is a wake-up tree of $\mathcal{P}$ and $\mathcal{E}_\mathrm{R} \subseteq \mathcal{P}^2$ is a set of *return arcs*, such that each $P_i \in \mathcal{P}$ is the destination of exactly one return arc and the graph $(\mathcal{P}, \mathcal{E}_\mathsf{T} \cup \mathcal{E}_\mathrm{R})$ is an oriented Eulerian graph (i.e. each vertex has as many outgoing arcs as incoming arcs). Self-loops are allowed and correspond to a robot staying still during the process (see Figure 1a). The *completion time* of r_j is the quantity $\mathrm{co}(r_i) = \mathrm{wu}_{\mathsf{T}}(r_j) + d(P_j, P_i)$, where $(P_j, P_i) \in \mathcal{E}_{\mathrm{R}}$. The *makespan* of a pair $(\mathsf{T}, \mathcal{E}_{\mathrm{R}})$ is the maximum over all completion times.

A solution $(\mathsf{T}, \mathcal{E}_{\mathrm{R}})$ for an instance of FTRPA is also a solution for the same instance of FTRPID if each robot r_i returns to its initial position P_i after waking up the other robots. It is the case if each edge of the graph $(\mathcal{P}, \mathcal{E}_{\mathsf{T}} \cup \mathcal{E}_{\mathrm{R}})$ belongs to a unique cycle. The solution depicted on Figure 1b satisfies this property, while the one on Figure 1a does not.

Unlike for FTP or FTRPA where robots are indistinguishable, defining a strategy in ID context requires specifying each robot's *trajectory.* For $r_i \in \mathcal{R}$ we define $\mathsf{last}_{r_i} \in \mathcal{P}$ as the last position visited by r_i before returning to P_i. As an example, given a FTRPID solution encoded by a pair $(\mathsf{T}, \mathcal{E}_{\mathrm{R}})$, r_i's trajectory is the unique cycle of $(\mathcal{P}, \mathcal{E}_{\mathsf{T}} \cup \mathcal{E}_{\mathrm{R}})$ containing the return arc that ends at P_i. This arc is $(\mathsf{last}_{r_i}, P_i)$ and is called *r_i's return arc.*

Conversely, given a trajectory for each robot (seen as a sequence of points of $\mathcal{P}$), it is immediate to get the corresponding wake-up tree, where each trajectory (minus the return arc) will still appear as a branch of that tree. There is a one-to-one correspondence between robots and trajectories by introducing a bicoloring of the tree arcs (as depicted on Figure 1c). Most strategies depicted in the present article are represented using bicolored trees. The return arc of a robot r_i always goes from the leaf of the monochromatic path rooted in P_i, which is last_{r_i}, to P_i.

3 Lower and Upper Bounds for Euclidean Plane

3.1 Lower bounds

As soon as $n \geqslant 3$, getting a lower bound on $\gamma_{\mathrm{ID}}(n)$ and $\gamma_{\mathrm{A}}(n)$ is not straightforward. We only give here some hints to explain how we get our bounds. Informally, the construction of the lower bounds is based on regular convex configurations for $n = 3$ and $n = 4$ and on properties of the wake-up trees (number of leaves and the minimum completion time for the robots reaching the leaves).

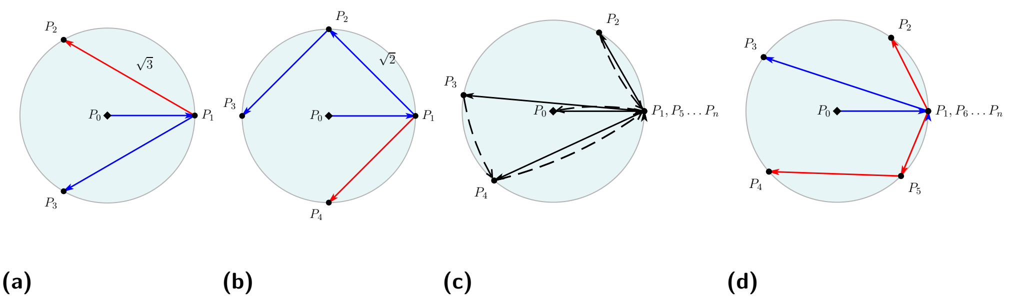


Figure 2 (a) and (b): Worst distributions for $n = 3$ and $n = 4$, and their optimal wake-up trees for FTRPID. (c): Asymptotic lower bound for FTRPA. (d): Asymptotic lower bound for FTRPID. In these two last configurations, there are respectively $n - 3$, $n - 4$ robots at P_1.

▶ **Theorem 1** (Lower bounds). *The following inequalities hold true.*

- $\gamma_{\mathrm{ID}}(3) \geqslant \gamma_{\mathrm{A}}(3) \geqslant 1 + 2\sqrt{3} \approx 4.46$,
- $\gamma_{\mathrm{ID}}(4) \geqslant \gamma_{\mathrm{A}}(4) \geqslant 2 + 2\sqrt{2} \approx 4.824$,
- *For any* $n \geqslant 5$, $\gamma_{\mathrm{A}}(n) \geqslant 1 + \frac{1}{4}(7^{1/4} + 7^{3/4})\sqrt{6} > 4.631$ *and* $\gamma_{\mathrm{ID}}(n) > 4.672$.

It turns out that these lower bounds are tight for $n = 3$ and $n = 4$ (as we show in the next section). For $n \geqslant 5$, a new phenomenon appears: the lower bounds (and the underlying distributions of robots) are not the same for $\gamma_{\text{ID}}(n)$ and $\gamma_{\text{A}}(n)$. Moreover, the worst distribution of robots is no longer regular. For instance, to lower bound $\gamma_{\text{A}}(n)$ (see Figure 2c) whenever $n \geqslant 5$, we spread the n sleeping robots on 4 sites: $n-3$ sleeping robots are located on the unit circle with the same angle $\alpha_1 = 0$, i.e., with $P_1 = P_5 = \cdots = P_n$, whereas the angles of the three remaining robots at P_2, P_3, P_4 are respectively

$$\alpha_2 = 2\arctan\left(\frac{7^{1/4}\sqrt{6}}{3\sqrt{7}-1}\right), \quad \alpha_3 = \alpha_2 + 2\arctan\left(\frac{7^{1/4}\sqrt{6}}{(\sqrt{7}-1)(4\sqrt{7}-6)}\right), \quad \alpha_4 = \alpha_3 + \alpha_2.$$

In this configuration, regardless of the value of n, at most 2 robots positioned at P_1 need to move to wake the other robots. It is therefore sufficient to analyze the case $n = 5$, which is done by evaluating a finite number of wake-up trees. See the full version [6] for the full proof of Theorem 1.

3.2 Upper bounds

3.2.1 The convex case

In this section we focus on the case where sleeping robots are in a convex configuration within a unit disk. We provide a dedicated algorithm, called Convex, and we show that it achieves a makespan of $2 + 2\sqrt{2}$ for the FTRPid.

▶ **Theorem 2.** *For any unit-disk instance*[1] $\mathcal{P}$ *where the sleeping robots are in convex position, the makespan of FTRPid for* $\mathcal{P}$ *is at most* $2 + 2\sqrt{2}$.

Assuming that positions can be ordered cyclically with respect to their angular distances to a fixed point, we sketch Algorithm Convex (see Figure 3) used for the proof of Theorem 2: (1) First, we show that the worst distribution is obtained whenever the sleeping robots lie on the boundary of the unit circle. (2) We then define a simple algorithm, called Linear, to wake up *consecutive* robots according to a cyclic ordering: informally, robot r_i located at position P_i wakes up r_{i+1} and returns to P_i. (3) Given a specific geometric position Q, we partition the circle into 3 arcs $(P_1, Q), (Q, P_k)$ and (P_{k+1}, P_1) with respect to the maximum empty cone of angle β between points P_k and P_{k+1}. Point P_1 is diametrically opposed to the widest empty cone of measure β. Robot r_0 wakes up the extreme robots r_1, r_2 and r_j and returns to P_0. (4) In each arc, Algorithm Linear is applied. The analysis is based on a precise analysis of this algorithm and the position of Q.

As a corollary of Theorem 2, we can get upper bounds for $\gamma_{\text{ID}}(n)$ and $\gamma_{\text{A}}(n)$ for $n \in \{3, 4\}$ by adding a specific algorithm for the non-convex distributions for $n = 4$ (for a proof, see the full version [6]). According to Theorem 1, these upper bounds are best possible.

▶ **Corollary 3.** *It holds that* $\gamma_{\text{A}}(3) = \gamma_{\text{ID}}(3) = 1 + 2\sqrt{3}$ *and* $\gamma_{\text{A}}(4) = \gamma_{\text{ID}}(4) = 2 + 2\sqrt{2}$.

3.2.2 Asymptotic upper bound

We adapt the FTP algorithm from [5, Theorem 2] (cone+return+cone). In this version, the disk is divided into $\sqrt{n}$ cones of equal angle. First, we wake up the robots in the densest cone. This cone contains at least $\sqrt{n}$ robots. For each remaining cone, we use one awake robot.

---

[1] That is a distribution of robots in the unit disk of $(\mathbb{R}^2, \ell_2)$ where the initial awake robot is at the origin.

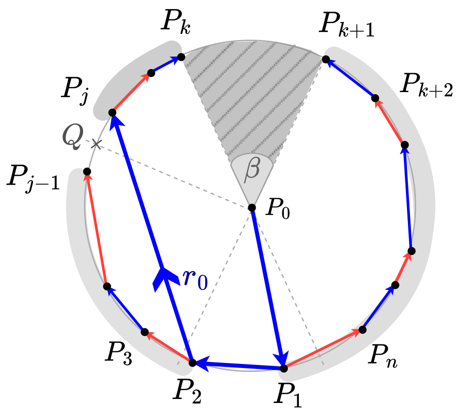


Figure 3 Bi-colored wake-up tree resulting from Algorithm Convex in the proof of Theorem 2. Highlighted arcs are woken up using Algorithm Linear.

This algorithm is called SplitConeStrategy. It runs in $O(n \log n)$ time and produces a makespan of at most $3 + O(1/\sqrt{n}\,)$ for the FTP problem. Unfortunately, for the FTRP problem, the makespan of this algorithm can reach $5 + O(1/\sqrt{n}\,)$ (for example, for n points equally distributed on the unit circle). Indeed, the robot that must wake up the cone opposite to its initial position might have a return arc of length up to 2.

Our new algorithm relies on two ideas. The first idea is to use the initial robot to wake up the cone C_b opposite to the initial cone and assign it a larger opening $\alpha_b \approx 0.9439$. The second idea is to use a second robot if C_b contains another robot at a distance of at most $\rho_c \approx 0.9445$ from P_0. With these two ingredients, we can reduce the makespan to $4.89 + O(1/\sqrt{n}\,)$.

▶ **Theorem 4.** *For any unit-disk instance with n sleeping robots, we can build in time $O(n \log n)$ an FTRPId solution with a makespan of $4.89 + O(1/\sqrt{n}\,)$.*

3.2.3 General upper bound

In this section, we present the general ideas of the proof of Theorem 5. The detailed proofs are deferred to the full version [6].

▶ **Theorem 5.** *Let t be the makespan of a solution of the FTP for a unit-disk instance $\mathcal{P}$. There exists a solution of makespan t_R for the FTRPId on $\mathcal{P}$ such that $t_R < t + 1.9583$. In particular, for every $n \in \mathbb{N}$, $\gamma_{\text{ID}}(n) < \gamma(n) + 1.9583$.*

We observe that the constant 1.9583 in Theorem 5 cannot be replaced by any constant less or equal to $\sqrt{3} \approx 1.732$. Indeed, consider the worst-case distribution for $\gamma_{\text{ID}}(3)$, where the sleeping 3 robots are pairwise at distance $\sqrt{3}$ (cf. Figure 2a). Clearly, such distribution has a solution for FTP with makespan $t \leqslant 1 + \sqrt{3}$, whereas the makespan of any solution for FTRPId or FTRPA must be at least $t_R \geqslant 1 + 2\sqrt{3} \geqslant t + \sqrt{3}$ (cf. Theorem 1).

Let us sketch the proof of Theorem 5. We show that there is some $\varepsilon > 0$ such that, for each distribution $\mathcal{P}$ of robots, for which there is a wake-up tree T with makespan t solving FTP on $\mathcal{P}$, there is also a wake-up tree and return arcs for FTRPId with makespan at most $t + 2 - \varepsilon$. For a specific value of ε, we show that 0.0417 can be chosen, but most of our arguments apply for possibly way larger values of ε. However, from the previous discussion such ε must be less than $2 - \sqrt{3} \approx 0.2679$.

Starting with a wake-up tree T for $\mathcal{P}$, we can consider any solution $(\mathsf{T}, \mathcal{E}_{\mathrm{R}})$ for FTRPID. For each robot $r \in \mathcal{R}$, its completion time is $\mathrm{wu}_{\mathsf{T}}(\mathsf{last}_r) + |\mathsf{last}_r P_r|$. Let (r, r') be a pair of robots such that $P_{r'}$ is an ancestor of P_r in T. We say that $(P_r, P_{r'})$ is an *ε-good pair*, if $\mathrm{wu}_{\mathsf{T}}(P_r) + |P_r P_{r'}| \leqslant t + 2 - \varepsilon$. Otherwise, we call $(P_r, P_{r'})$ an *ε-bad pair*. Similarly, we call a return arc (last_r, P_r) *ε-good* if (last_r, P_r) is an ε-good pair, and *ε-bad* otherwise. We denote by $\mathcal{B} \subset \mathcal{R}$ the set of all robots r such that (last_r, P_r) is an ε-bad return arc.

The overall goal of the proof is to build on the top of T a wake-up tree T' for FTP, and a set of return arcs $\mathcal{E}'_{\mathrm{R}}$ such that there are no ε-bad return arcs in $(\mathsf{T}', \mathcal{E}'_{\mathrm{R}})$. As it is necessary to alter T in order to remove its ε-bad return arcs, T' is possibly different of T. For the construction, it is relevant to start with a solution $(\mathsf{T}, \mathcal{E}_{\mathrm{R}})$ that minimizes the number of ε-bad return arcs among all solutions for FTRPID based on T. Given a set of return arcs $\mathcal{E}_{\mathrm{R}}$, we can define the ε-cost of $\mathcal{E}_{\mathrm{R}}$ as the sequence $(b_k)_{k \in \mathbb{N}}$ where b_k is the number of robots $r \in \mathcal{B}$ such that the path from P_{r_0} to P_r in T has k edges. In the following, we consider that $\mathcal{E}_{\mathrm{R}}$ minimizes the number of ε-bad return arcs and, among all such solutions, has the lexicographic largest ε-cost.

Construction of T'. The first part of the construction is disconnecting branches of T as low as possible, so that there are no ε-bad return arcs in the remaining wake-up tree. The resulting wake-up tree does not completely span $\mathcal{P}$, meaning we will have to complete the tree later. Informally, for every $r \in \mathcal{B}$, we let r follow the wake-up tree as long as its return does not induce an ε-bad return arc, and then come back to its initial position. For every $r \in \mathcal{B}$, let G_r be the closest ancestor of last_r in T for which (G_r, P_r) is an ε-good pair. Such an ancestor exists since (P_r, P_r) is guaranteed to be a ε-good pair. Moreover, let B_r be the child of G_r leading to last_r, and let $\mathsf{T}_{\mathsf{B}_r}$ be the subtree of T rooted in B_r. We denote by $\mathcal{L}_r$ the set of the robots with initial positions in $\mathsf{T}_{\mathsf{B}_r}$. The first operation is disconnecting $\mathcal{L}_r$ from T and adding the ε-good return arc (G_r, P_r) to $\mathcal{E}'_{\mathrm{R}}$.

To state our results about the robots in $\mathcal{L}_r$, we define a *crown of angle α and width w* as the region containing all points between distance $1 - w$ and 1 from P_0 and whose angular coordinates are within a cone of angle α centered at P_0. Moreover, we call a crown of width ε and of angle 2π an *ε-crown*. Our result is based on the following technical lemma about makespan of crowns, that we believe interesting on its own.

▶ **Lemma 6.** *Let C be a crown of angle α and width w. Assuming that the initial awake robot is anywhere inside C, there exists a strategy for solving FTRPID in C with a makespan $M(C) \leqslant 2\alpha + w \cdot (4 + \varphi)$ where $\varphi = (1 + \sqrt{5})/2$ is the Golden ratio.*

The key lemma about robots in $\mathcal{L}_r$ is the following.

▶ **Lemma 7.** *Let (last_r, P_r) be an ε-bad return arc in T and let $x \in \mathcal{L}_r$. Then, P_x is on the ε-crown and has distance at most ε to some leaf of the subtree of T rooted in B_r.*

After the previously mentioned trimming operation, robots of $\mathcal{L} = \bigcup_{r \in \mathcal{B}} \mathcal{L}_r$ are disconnected from the wake-up tree. The second step is to find a strategy for the awakening of these robots. Lemma 7 guarantees that for each $r \in \mathcal{B}$, the robots of $\mathcal{L}_r$ are located within an ε-crown. Now, for every $r \in \mathcal{B}$, we will designate some robot $r' \in \mathcal{R} \backslash \mathcal{L}$ that was not disconnected during the first operation, and such that $P_{r'}$ is within the ε-crown. This robot r', after being awakened, wakes up all the robots of $\mathcal{L}_r$ using a dedicated FTP algorithm. Since the returns are confined to a small sub-crown of width ε, this step will not significantly increase the makespan as stated in Lemma 6.

More precisely, given $r \in \mathcal{B}$, we first define $r^\star$ and its initial position A_r, under the assumption that $\varepsilon \in (0,1)$. Let A_r, called the *best ancestor of* r, be the closest ancestor of P_r in T such that the arc $(\mathsf{last}_{r^\star}, \mathsf{A}_r)$ is an ε-good return arc and $r^\star$ wakes up r or a robot located at an ancestor of P_r. Now we can define r' as being the robot initially located at $\mathsf{last}_{r^\star}$. Note that $r', r^\star$ and A_r always exist (cf. the full version [6]).

It is possible that several distinct robots of $\mathcal{B}$ share the same best ancestor. Thus we denote by $\mathcal{B}_{r^\star}$ the subset of robots in $\mathcal{B}$ whose best ancestor is also A_r.

What is left to do is to show that $r' \in \mathcal{R}\backslash\mathcal{L}$ and that the positions of r' and robots in $\bigcup_{r'' \in \mathcal{B}_{r^\star}} \mathcal{L}_{r''}$ are close to each other. Roughly, the latter is done by exhibiting one position, denoted $\mathsf{W}_{r^\star}$, that is far enough from both the position of r' and robots of $\bigcup_{r'' \in \mathcal{B}_{r^\star}} \mathcal{L}_{r''}$, to guarantee that they lie in the same area of the unit disk. See Figure 4.

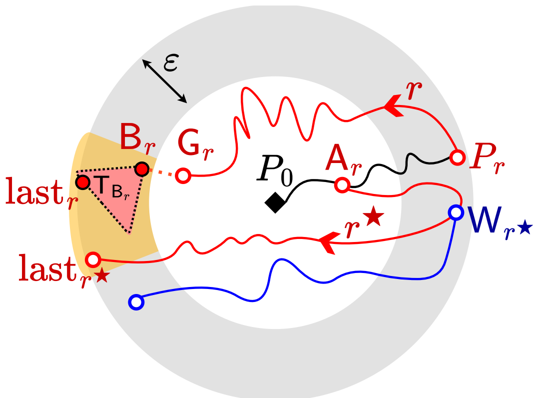


Figure 4 Robots r, $r^\star$ and r_{W} are initially located at positions P_r, A_r and $\mathsf{W}_{r^\star}$. Robots of $\mathcal{L}_r$ within the highlighted sub-crown of angle α_ε (given by Lemma 8) are awakened from a robot located at $\mathsf{last}_{r^\star}$. Positions $\mathsf{last}_{r^\star}$ and leaves of $\mathsf{T}_{\mathsf{B}_r}$ are at distance at least $2-\varepsilon$ from position $\mathsf{W}_{r^\star}$.

▶ **Lemma 8.** *If $\varepsilon \in (0,1/2)$, then there is a crown of width ε and angle $\alpha_\varepsilon = 2\arcsin\left(\frac{\varepsilon}{2(1-\varepsilon)}\right) + 4\arccos\left(1-\frac{\varepsilon}{2}\right)$ that contains $\mathsf{last}_{r^\star}$ and the initial position of each robot in $\bigcup_{r'' \in \mathcal{B}_{r^\star}} \mathcal{L}_{r''}$.*

Now, the makespan of robots in $\mathcal{R}\backslash\mathcal{L}$ is at most $t+2-\varepsilon$ since by construction there are no ε-bad return arcs. On the other hand, Lemma 6 and Lemma 8 guarantee that the makespan of robots in $\mathcal{L}$ is at most $t + 2\alpha_\varepsilon + (4+\varphi)\varepsilon$. By setting $\varepsilon = 0.0417$, we obtain a makespan dominated by $t + 1.9583$ in both situations, which completes the proof of Theorem 5.

4 NP-Hardness for Unweighted Graphs

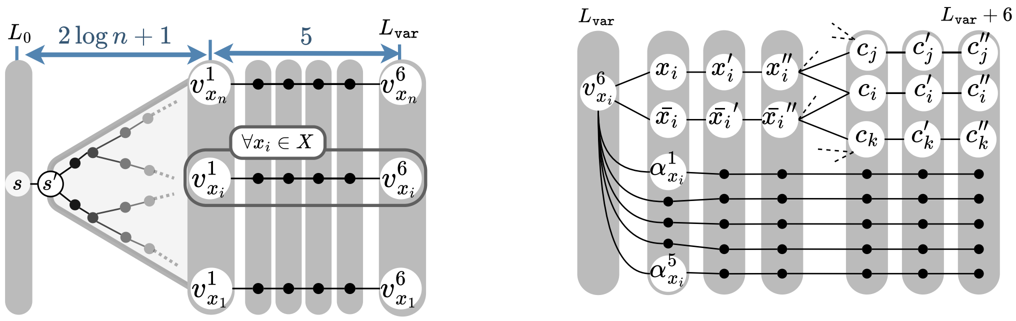
In this section, we study the two variants of FTRP on graphs. The space is defined by a graph G with non-negative weights on the edges. Each vertex of the graph initially contains one robot and the robots move along edges. A robot sleeping on a vertex v is woken up as soon as an awake robot reaches v.

▶ **Theorem 9.** *FTRPId and FTRPA are NP-hard on unweighted graphs of constant maximum degree and diameter $O(\log n)$. Moreover, unless the ETH fails, neither of these problems on graphs can be solved in $2^{o(n+m)}$ time, where n and m denote the number of vertices and edges, respectively.*

We reduce from a variant of 3-SAT where we assume that each literal occurs at most twice. This variant is NP-complete and cannot be solved in $2^{o(|F|)}$ time, unless the ETH fails, as the reduction by Tovey [16] uses a linear space reduction. This property allows us to describe a construction of bounded degree and where the number of vertices is linear in the size of the formula F.

Let F be an instance of this 3-SAT variant with variable set X. For technical reasons, we add for each variable $x \in X$ the clause $(x \vee \overline{x})$ to the formula. We denote by F' the resulting formula, and by C the clauses of F'. It is clear that F is satisfiable if and only if F' is satisfiable. Note that each variable occurs at most three times positively and at most three times negatively in F'.

Construction. We now define the unweighted graph $G = (V, E)$ for our instance of FTRP. Let $n = |X|$. For the sake of simplicity, we assume that n is a power of 2.



(a) Set-up gadget. **(b)** Variables and clauses gadget.

Figure 5 An example for the gadget constructions.

Set-up gadget. We start with a vertex s at which the source robot is initially located, and the graph obtained from subdividing each edge of a balanced binary tree rooted at a vertex s' with n leaves. The leaves are denoted by a_x for each $x \in X$.[2] We connect s to s', where log Next, for each $x \in X$ we add a path $(a_x =: v^1_x, v^2_x, v^3_x, v^4_x, v^5_x, v^6_x =: v_x)$. We get a binary tree rooted at s having n leaves that has height $2\log n + 6$. See Figure 5a.

Variable gadget. For each variable $x \in X$, we add two paths (x, x', x'') and $(\overline{x}, \overline{x}', \overline{x}'')$, and we connect x and $\overline{x}$ to v_x. These two paths will encode the truth assignment of x depending on which of the robots on x or $\overline{x}$ is woken up first. For each $i \in [1, 5]$, connect to v_x a path on 6 vertices whose first vertex (the one adjacent to v_x) is denoted by α^i_x and whose last vertex is denoted by ω^i_x. The distance from s to any ω^i_x is exactly $2\log n + 12$.

Clause gadget. For each clause $c \in C$, add a path (c, c', c''). For each literal x $(\overline{x})$ occurring in clause c, there is an edge connecting c and x'' $(\overline{x}'')$. Since each clause contains at least one literal, for $c \in C$, the distance from s to c'' is $2\log n + 12$. See Figure 5b.

Notations. We define the *depth* of G, $d := 2\log n + 12$, which is the eccentricity of s. Let $0 \leqslant i \neq j \leqslant d$. We denote by L_i the *i-th layer* of the graph, formally $L_i := \{v \in V \mid |sv| = i\}$ where $|sv|$ denotes the length of a shortest path from s to v. We say that a vertex $u \in L_i$ is an *ancestor* of $v \in L_j$, if the distance from u to v is $j - i$. Note that this is the case if there is a shortest path from s to v that contains u. Furthermore, note

[2] Note that this implies that each leave a_x has distance exactly $2 \cdot \log n + 1$ from s, where the log function is in base two.

that $L_{\mathsf{var}} = \{v_x \mid x \in X\}$, where $\mathsf{var} := d - 6 = 2\log n + 6$. Moreover, note that the number of vertices of the graph is linear in the size of the formula F' and thus linear in the size of the formula F.

Proof summary. The correctness of the reduction is shown by proving the following.

▶ **Lemma 10.** *If F' is satisfiable, then there is a FTRPID solution of makespan $d + 1$. Moreover, if there is a FTRPA solution of makespan $d + 1$, then F' is satisfiable.*

As each wake-up tree for FTRPID is also a solution for FTRPA, this then shows that both problems are NP-hard. Moreover, no wake-up tree with any kind of return can have a makespan smaller than $d + 1$.

Sketch of the proof. (See the full version [6] for full proof of Lemma 10.) Our arguments rely on the fact that robots in layer L_d have to be woken up in a time that is exactly d, which is their distance from the root vertex. This strongly constrains the robot's trajectories, as almost every robot must follow a shortest path. The graph structure also ensures that only a fixed number of robots can reach some critical layers, as the number of positions to return to within a fixed radius is bounded.

The first $2\log n + 1$ layers are designed in a way, such that all robots that are initially located on these layers can be woken up and return to their initial position at time at most $2\log n + 2 < d + 1$, while ensuring that robots of the vertices a_x with $x \in X$ are all woken up at time $2\log n + 1$. These robots then follow their respective shortest paths to the variable gadget, that is, to the vertex v_x for each $x \in X$. The robots woken up on these paths follow toward the variable gadget. At time var, we will then have 6 awake robots located at each vertex v_x. For each $x \in X$, five of these robots wake up the robots on the vertices α_x^i with $i \in [1, 5]$, whereas the sixth robot can only wake up the robot on one of the vertices x or $\overline{x}$ at time $\mathsf{var} + 1$. The set of literals for which the respective vertex is woken up at time $\mathsf{var} + 1$ then models a truth assignment. Moreover, this truth assignment satisfies formula F' if and only if the overall makespan with return is at most $d + 1$. See Figure 6 for an example of such wake-up strategy represented by a bi-colored wake-up tree.

For the anonymous return (FTRPA), this wake-up strategy is in fact necessary: To obtain a makespan with return of $d + 1$, each robot initially located at a vertex of layer L_d needs to be woken up at time d. This holds in particular for the robot on vertex c'' for each clause $c \in C$. Based on the structure of the clause and variable gadgets, this ensures that at time $\mathsf{var} + 1$ the robot on at least one literal of c is woken up, as otherwise, no robot can reach c'' at a time prior to $d + 1$. The only remaining obstacle is then to ensure that for each variable only one robot on one of its literals is woken up at time $\mathsf{var} + 1$. Intuitively, this holds due to two facts: Firstly, for each variable x, at least one of the robots located on a literal of x needs to be woken up at time $\mathsf{var} + 1$, as $(x \vee \overline{x})$ is a clause of F'. Secondly, no more than $6n$ robots can wake up robots on vertices of layer $L_{\mathsf{var}+1}$ at time $\mathsf{var} + 1$, as otherwise, not all of these robots can return to pairwise distinct positions in the remaining time until the end of the makespan of $d + 1$. This is essentially due to the fact that the layers L_ℓ with $\ell \in [\mathsf{var} - 5, \mathsf{var}]$ together only contain $6n$ vertices, and each vertex needs to contain exactly one robot at time $d + 1 = \mathsf{var} + 7$. By the fact that $5n$ of these robots are necessary to wake up the robots initially located on the vertices of $\{\alpha_x^i \mid x \in X, \in [1, 5]\}$, for only one literal per variable can the initially located robot be woken up at time $\mathsf{var} + 1$.

▶ **Corollary 11.** *FTRPID and FTRPA are NP-hard in metric spaces and cannot be solved in $2^{o(n)}$ time, unless the ETH fails.*

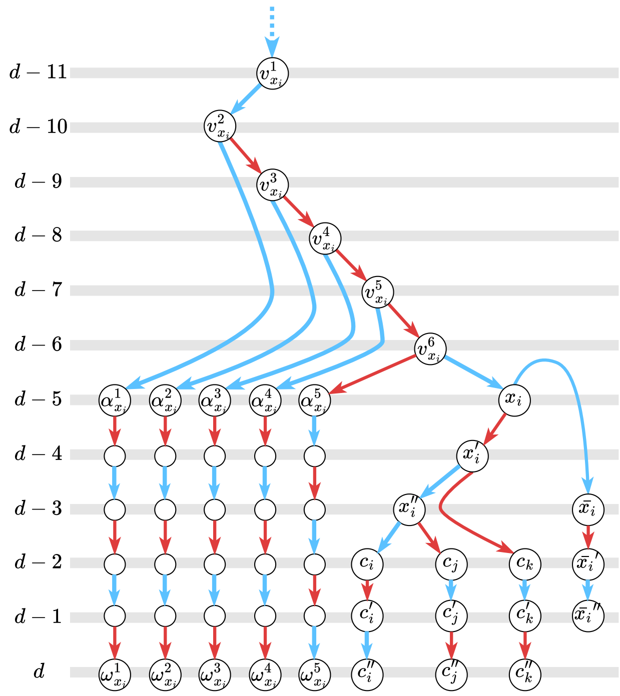


Figure 6 Bi-colored wake-up tree for the variable and clause gadget. Curved arcs to hints the path taken in the graph.

5 Optimal Algorithms

In this part we develop algorithms for FTP and its variants. They return an optimal solution, that is, the optimal makespan, assuming any non-negative distance function between the initial positions of the robots. In particular, they apply to metric spaces as well as weighted graphs, even if the triangle inequality does not hold.

It is clear that any brute-force algorithm needs to consider at least $n!$ solutions, that is at least as many solutions as the path-TSP[3] and the classical TSP, since these latter problems are valid solutions for the FTP and FTRP. Actually, this is even worse when considering all possible wake-up trees (that can be seen as labeled and binary trees), as this number raises up to $2^{\Omega(n)} \cdot n!$. In the following, we assume that the given distance function can be evaluated in constant time.

▶ **Theorem 12.** *For FTP, FTRPID, and FTRPA with any non-negative distance function, there are deterministic algorithms returning the optimal solution in time complexity $O(3^n \cdot n^2)$, $O(3^n \cdot n^3)$, and $O(9^n \cdot n^{3/2})$ respectively, with space complexity $O(2^n \cdot n)$, $O(2^n \cdot n^2)$, and $O(4^n \cdot n)$ respectively, where n is the number of sleeping robots.*

We stress that from our hardness results in Theorem 9 and under the ETH, no $2^{o(n)}$-time algorithms can exist for these problems. Here, we sketch our approach, details and proofs being available in the full version [6]. These algorithms have been implemented[4] and have contributed in determining some lower bounds of Theorem 1.

[3] In the *Path Traveling Salesperson Problem*, one asks for a path of minimum length connecting all given points.

[4] For the FTP version, a code is available in [10], and for the two variants with returns, FTRPID and FTRPA, their codes are available on demand.

Our algorithms are inspired by the famous Held–Karp's algorithm, independently discovered by Bellman in the 1960s. We use a recursive approach which for FTP is actually quite close to the original Bellman–Held–Karp's formulation of the path-TSP sub-problem decomposition of TSP. Then, we use a dynamic programming with exponential space. For the implementation of FTP and FTRPID, we can use a relatively simple pair of tables of respectively $2^n \times n$ and $2^n \times n^2$ cells, where each cell is assumed to be large enough to store a distance. Whereas for FTRPA, the most complicated variant, we analyze a standard memoization technique which leads, to a roughly 9^n-time algorithm.

6 Conclusion

We introduced two versions of the Freeze-Tag-with-Return Problem, a natural variant of the Freeze-Tag Problem. We showed that both versions are NP-hard when the underlying space is an unweighted undirected graph. However, it remains open whether the problems are NP-hard in other metric spaces, like $(\mathbb{R}^d, \ell_p)$ spaces ($d = 2$ and $p = 2$, the Euclidean plane).

Regarding the presented lower and upper bounds for the optimal makespan, we conjecture that they are not optimal for large instances. We found a distribution of positions (for any $n \geqslant 5$) with an optimal makespan of 4.67. We still have a gap between this lower bound and the two upper bounds of $4.89 + O(1/\sqrt{n}\,)$ for arbitrary distribution and of $2 + 2\sqrt{2} \approx 4.83$ for convex distribution.

In general, we conjecture that the optimal makespan of worst-case instances of both introduced problems, i.e., the wake-up return constants, never exceeds $2 + 2\sqrt{2}$. In other words:

▶ **Conjecture 13.** $\overline{\gamma_{ID}} = \overline{\gamma_{A}} = 2 + 2\sqrt{2}$.

It is also interesting to further analyze the connection between the worst-case makespan between FTP and FTRP for any fixed distribution of sleeping robots $\mathcal{P}$. We gave examples, where this difference can be larger than 1.732 but showed that it never exceeds 1.959. The latter was shown via an algorithm that takes any solution for FTP for the set $\mathcal{P}$ and computes a solution for FTRP for the set $\mathcal{P}$ where the makespan does only increase by at most 1.959. There are two main directions to improve in this area: (1) reduce the difference between upper and lower bounds and (2) obtain an efficient and constructive algorithm to achieve a solution for FTRP from a solution for FTP with an overhead of less than 2.

Finally, the definition of the FTRP might inspire the introduction of other variants of FTP, like for example, the task where all robots have to gather at the same point after the wake-up process is finished.

References

1 Zachary Abel, Hugo A. Akitaya, and Jingjin Yu. Freeze tag awakening in 2D is NP-hard. In *27th Annual Fall Workshop on Computational Geometry (FWCG)*, November 2017.

2 Sharareh Alipour, Arash Ahadi, and Kajal Baghestani. Improved wake-up time for euclidean freeze-tag problem. In *37th Canadian Conference on Computational Geometry (CCCG)*, pages 331–339, August 2025.

3 Esther M. Arkin, Michael A. Bender, Sándor P. Fekete, Joseph S.B. Mitchell, and Martin Skutella. The freeze-tag problem: How to wake up a swarm of robots. In *13th Symposium on Discrete Algorithms (SODA)*, pages 568–577. ACM-SIAM, 2002. URL: http://dl.acm.org/citation.cfm?id=545381.545457.

4 Esther M. Arkin, Michael A. Bender, Sándor P. Fekete, Joseph S.B. Mitchell, and Martin Skutella. The freeze-tag problem: How to wake up a swarm of robots. *Algorithmica*, 46:193–221, 2006. doi:10.1007/s00453-006-1206-1.

5 Nicolas Bonichon, Arnaud Casteigts, Cyril Gavoille, and Nicolas Hanusse. Freeze-tag in L_1 has wake-up time five with linear complexity. In Dan Alistarh, editor, *38th International Symposium on Distributed Computing (DISC)*, volume 319 of *LIPIcs*, pages 9:1–9:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, October 2024. doi:10.4230/LIPIcs.DISC.2024.9.

6 Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, Gabriel Le Bouder, Taïssir Marcé, and Nils Morawietz. Freeze-tag with return. Technical Report 2606.21985v1 [cs.DS], arXiv, June 2026. doi:10.48550/arXiv.2606.21985.

7 Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and Saeed Odak. Euclidean freeze-tag problem on plane. In *36th Canadian Conference on Computational Geometry (CCCG)*, pages 199–205, July 2024.

8 Josh Brunner and Julian Wellman. An optimal algorithm for online freeze-tag. In *10th International Conference Fun with Algorithms (FUN)*, volume 157 of *LIPIcs*, pages 8:1–11. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, September 2020. doi:10.4230/LIPIcs.FUN.2021.8.

9 Erik D. Demaine and Mikhail Rudoy. Freeze tag is hard in 3D. In *27th Annual Fall Workshop on Computational Geometry (FWCG)*, November 2017.

10 Cyril Gavoille. An optimal algorithm for the Freeze-Tag Problem in 2D. Software, LaBRI, Université Bordeaux, March 2022. URL: https://dept-info.labri.fr/~gavoille/soft/.

11 Cyril Gavoille, Nicolas Hanusse, Gabriel Le Bouder, and Taïssir Marcé. Distributed freeze tag: a sustainable solution to discover and wake-up a robot swarm. Technical Report 2503.22521v1 [cs.DS], arXiv, March 2025. doi:10.48550/arXiv.2503.22521.

12 Mikael Hammar, Bengt J. Nilsson, and Mia Persson. The online freeze-tag problem. In *7th Latin American Symposium on Theoretical Informatics (LATIN)*, volume 3887 of *Lecture Notes in Computer Science*, pages 569–579. Springer, March 2006. doi:10.1007/11682462_53.

13 Matthew Johnson. Easier hardness for 3D freeze-tag. In *27th Annual Fall Workshop on Computational Geometry (FWCG)*, November 2017.

14 Lehilton Lelis Chaves Pedrosa and Lucas de Oliveira Silva. Freeze-tag is NP-hard in 3D with L_1 distance. In *12th Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS)*, volume 223:C, pages 360–366. Procedia Computer Science, September 2023. doi:10.1016/j.procs.2023.08.248.

15 Lehiton Lelis Chaves Pedrosa and Lucas de Oliveira Silva. Freeze-tag is NP-hard in 2D with L_1 distance. Technical Report 2509.14357v1 [cs.CG], arXiv, September 2025. arXiv:2509.14357v1.

16 Craig A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85–89, April 1984. doi:10.1016/0166-218X(84)90081-7.

17 Ehsan Najafi Yazdia, Alireza Bagheri, Zahra Moezkarimia, and Hamidreza Keshavarz. An $O(1)$-approximation algorithm for the 2-dimensional geometric freeze-tag problem. *Information Processing Letters*, 115(6-8):618–622, June 2015. doi:10.1016/j.ipl.2015.02.011.