

# Freeze-Tag in $L_1$ has Wake-up Time Five with Linear Complexity

N. Bonichon



A. Casteigts



C. Gavoille



N. Hanusse



université  
de BORDEAUX



LaBRI

38th International Symposium on Distributed Computing (DISC)  
Madrid, Spain – October 1st, 2024

# Context

A robot in charge of awaking a team of asleep robots.



## Subject to:

- awaking by contacts (need a move to meet)
- awaked robots can help to awake others
- robots move in the Euclidean plane
- constant velocity moves

## Goal:

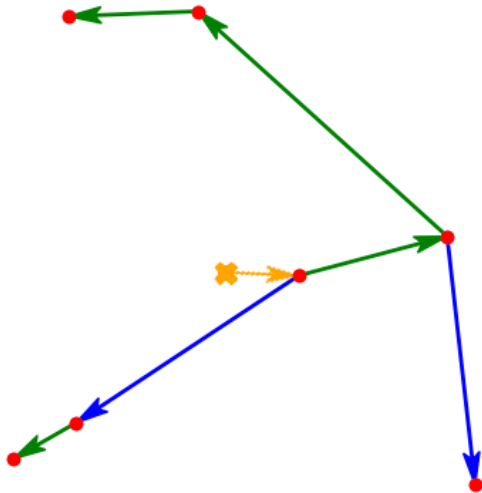
- to minimize the **makespan**, the wake-up time of the last robot
- with a schedule computable efficiently

[[demo](#)]

#pts = 8  
diam = 1.890  
ecc = 1.000

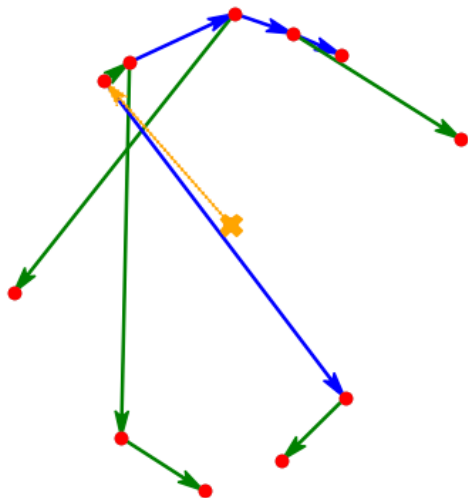


optimal = 2.110  
depth = 2.110  
#pts = 8  
diam = 1.890  
ecc = 1.000  
time = 0.009s





convex pts  
#pts = 12  
diam = 1.783  
ecc = 1.000  
time = 1.372s  
seed = 3906



optimal = 2.594

depth = 2.594

#pts = 12

diam = 1.783

ecc = 1.000

seed = 3906

The screenshot shows a web browser window with the URL <https://topp.openproblem.net>. The page title is "The Open Problems Project". It is edited by Erik D. Demaine, Joseph S. B. Mitchell, and Joseph O'Rourke. The page includes an introduction, a numerical list of all problems, and a categorized list of all problems. The categorized list includes sections for arrangements, art galleries, and coloring.

# The Open Problems Project

edited by [Erik D. Demaine](#), [Joseph S. B. Mitchell](#), [Joseph O'Rourke](#)

## Introduction

This project originally aimed to record important open problems of interest to researchers in computational geometry and related fields. It commenced in 2001 with the publication of thirty problems in Computational Geometry Column 42 [MO01] (see [Problems 1–30](#)), and then grew to over 75 problems.

While we are no longer encouraging new problem submissions, we strongly encourage updates to existing problems, especially when those problems have been solved (completely or partially). Updates should be done in the form of a Github Pull Request; see the [Github repository](#).

Each problem is assigned a unique number for citation purposes. Problem numbers also indicate the order in which the problems were entered. Each problem is classified as belonging to one or more categories.

The problems are also available as a single [PDF](#) file.

## Numerical List of All Problems

To begin navigating through the open problems, you may select from a category of interest below, or view [a list of all problems sorted numerically](#).

## Categorized List of All Problems

Below, each category lists the problems that are classified under that category. Note that each problem may be classified under several categories.

**arrangements:**

- [3-Colorability of Arrangements of Great Circles \(Problem 44\)](#)

**art galleries:**

- [Vertex  \$\pi\$ -Floodlights \(Problem 23\)](#)

**coloring:**

The screenshot shows a web browser window with the address bar containing <https://topp.openproblem.net/p35>. The page title is "TOPP: Problem 35: Freeze-Tag: Opti...". The browser's address bar also shows a search engine "Rechercher". The page content includes a blue header "The Open Problems Project", a "Next" link to "Problem 36: Inplace Convex Hull of a Simple Polygonal Chain", and a "Previous" link to "Problem 34: Extending Pseudosegment Arrangements by Subdivision". The main heading is "Problem 35: Freeze-Tag: Optimal Strategies for Awakening a Swarm of Robots". Below this is a "Statement" section with a paragraph describing the optimization problem of waking up robots. It also includes an "Origin" section with a citation "[ABF+02]" and a "Status/Conjectures" section with a paragraph stating the NP-hardness conjecture.

## The Open Problems Project

Next: [Problem 36: Inplace Convex Hull of a Simple Polygonal Chain](#)

Previous: [Problem 34: Extending Pseudosegment Arrangements by Subdivision](#)

# Problem 35: Freeze-Tag: Optimal Strategies for Awakening a Swarm of Robots

### Statement

An optimization problem that naturally arises in the study of “swarm robotics” is to wake up a set of “asleep” robots, starting with only one “awake” robot. One robot can only awaken another when they are in the same location. As soon as a robot is awake, it may assist in waking up other robots. The goal is to compute an optimal *awakening schedule* such that all robots are awake by time  $t^*$ , for the smallest possible value of  $t^*$  (the optimal *makespan*). The  $n$  robots are initially at  $n$  points of a metric space. The problem is equivalent to finding a spanning tree with maximum out-degree two that minimizes the radius from a fixed source.

Is it NP-hard to determine an optimal awakening schedule for robots in the Euclidean (or  $L_1$ ) plane? In more general metric spaces, can one obtain an approximation algorithm with better than  $O(\log n)$  performance ratio?

### Origin

[ABF+02]

### Status/Conjectures

[ABF+02] conjecture that the freeze-tag problem is NP-hard in the Euclidean (or  $L_1$ ) plane. (They show it to be NP-complete in star metrics.)

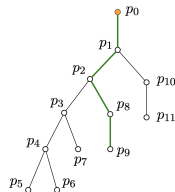
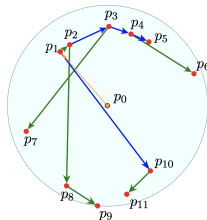


# Problem statement

## FREEZE-TAG PROBLEM (FTP)

**Input:** a source  $s \in \mathbb{R}^2$   
a set  $P \subseteq \mathbb{R}^2$  of  $n$  points

**Output:** a *wake-up tree* for  $(s, P)$   
with minimum *makespan*

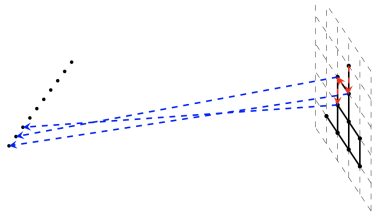


A **wake-up tree** for  $(s, P)$  is a binary tree spanning  $\{s\} \cup P$  of root  $s$  which has at most 1 child. **Makespan** is its weighted-depth w.r.t.  $\ell_2$ -norm.

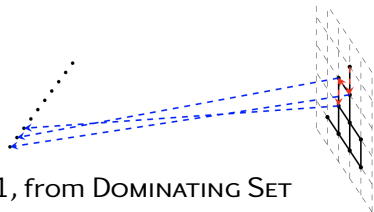
**Variants:**  $\ell_p$ -norm, metric-spaces, weighted graphs,  $\mathbb{R}^3$ , ...

- [ABFMS06] NP-Hard for moves restricted to weighted star graphs, from NUMERICAL 3D MATCHING, and for unweighted trees, from 3-PARTITION (asleep robots at leaves)

- [ABFMS06] NP-Hard for moves restricted to weighted star graphs, from NUMERICAL 3D MATCHING, and for unweighted trees, from 3-PARTITION (asleep robots at leaves)
- [AY16,J17] NP-Hard for  $(\mathbb{R}^3, \ell_2)$ , from HAMILTONIAN PATH IN 2D SUBGRID



- [ABFMS06] NP-Hard for moves restricted to weighted star graphs, from NUMERICAL 3D MATCHING, and for unweighted trees, from 3-PARTITION (asleep robots at leaves)
- [AY16,J17] NP-Hard for  $(\mathbb{R}^3, \ell_2)$ , from HAMILTONIAN PATH IN 2D SUBGRID

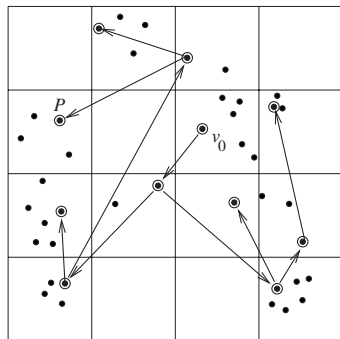


- [DR17] NP-Hard for  $(\mathbb{R}^3, \ell_p)$ ,  $p > 1$ , from DOMINATING SET
- [AAJ17] NP-Hard for  $(\mathbb{R}^2, \ell_2)$ , from MONOTONE 3SAT
- [PdOS23] NP-Hard for  $(\mathbb{R}^3, \ell_1)$ , from MONOTONE 3SAT
- 
- [ABFMS02] **Conjecture.** NP-Hard for  $(\mathbb{R}^2, \ell_1)$ .

[ABFMS06]  $O(\log n)$ -approximation for locally-bounded weighted graphs (bounded aspect ratio for incident edges)

[ABFMS06]  $(1 + \varepsilon)$ -approximation for  $(\mathbb{R}^d, \ell_p)$  for fixed  $d$  and any  $p$ , in time  $O(n \log n) + 2^{(1/\varepsilon)^{O(1)}}$  if  $\varepsilon < 1/57$

TSP-inspired scheme:  
 Lowest level: wake-up  
 tree of depth  $O_{d,p}(\varepsilon)$



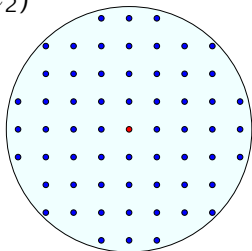
$r$  = distance to the farthest asleep robot

[Folklore] Lower bound: makespan  $\geq r$ .

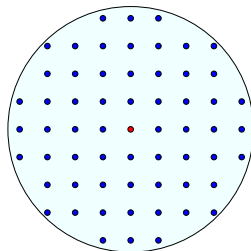
[ABFMS06]  $\exists$  wake-up tree of depth  $\leq \alpha r$ , for *wake-up constant*  $\alpha$ .

$$L_2 = (\mathbb{R}^2, \ell_2)$$

$$r = 1$$



$$\text{FTP}(n) = \Theta(1)$$



$$\text{TSP}(n) = \Theta(\sqrt{n})$$

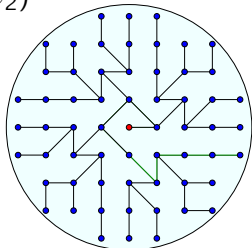
$r$  = distance to the farthest asleep robot

[Folklore] Lower bound: makespan  $\geq r$ .

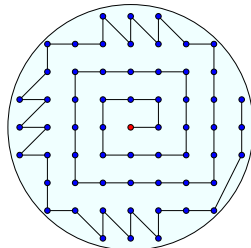
[ABFMS06]  $\exists$  wake-up tree of depth  $\leq \alpha r$ , for *wake-up constant*  $\alpha$ .

$L_2 = (\mathbb{R}^2, \ell_2)$

$r = 1$



FTP( $n$ ) =  $\Theta(1)$



TSP( $n$ ) =  $\Theta(\sqrt{n})$

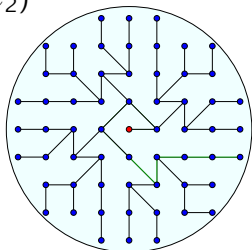
$r$  = distance to the farthest asleep robot

[Folklore] Lower bound: makespan  $\geq r$ .

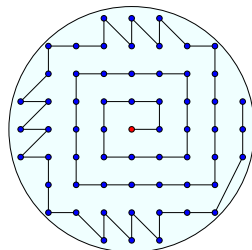
[ABFMS06]  $\exists$  wake-up tree of depth  $\leq \alpha r$ , for *wake-up constant*  $\alpha$ .

$L_2 = (\mathbb{R}^2, \ell_2)$

$r = 1$



FTP( $n$ ) =  $\Theta(1)$



TSP( $n$ ) =  $\Theta(\sqrt{n})$

[ABFMS06] Wake-up constant for  $L_2$  is  $\alpha(\ell_2) < 57$

[YBMK15] Wake-up constant for  $L_2$  is  $\alpha(\ell_2) < 5 + 2\sqrt{2} + \sqrt{5} \approx 10.06$



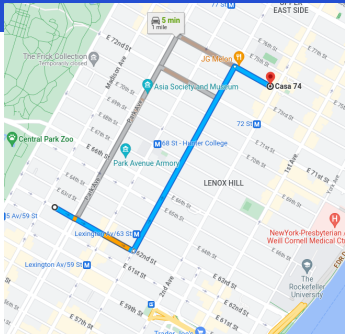


# Contributions

$L_1 = (\mathbb{R}^2, \ell_1)$ ,  $\ell_1 =$  Manhattan distance



robot moves  $\in \{\leftrightarrow, \updownarrow\}$



## Theorem (Main)

The wake-up constant for  $L_1$  is  $\alpha(\ell_1) = 5$ .

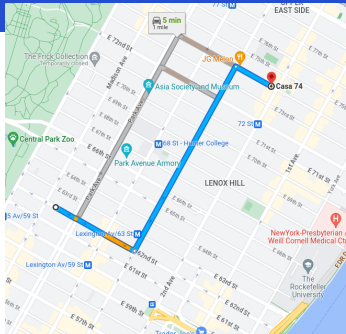
$\Rightarrow$  The wake-up constant for  $L_2$  is  $\alpha(\ell_2) \leq 5\sqrt{2} \approx 7.07$

# Contributions

$$L_1 = (\mathbb{R}^2, \ell_1), \ell_1 = \text{Manhattan distance}$$



robot moves  $\in \{\leftrightarrow, \updownarrow\}$



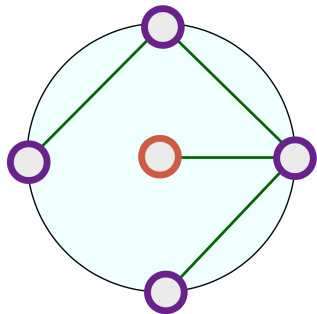
## Theorem (Main)

*The wake-up constant for  $L_1$  is  $\alpha(\ell_1) = 5$ .*

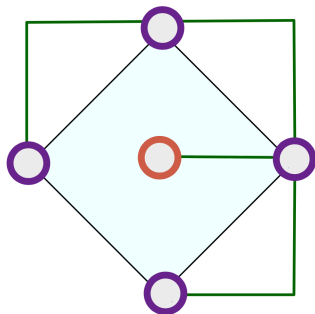
$\Rightarrow$  The wake-up constant for  $L_2$  is  $\alpha(\ell_2) \leq 5\sqrt{2} \approx 7.07$

## Theorem (Generic Algorithm, and more – not in this talk)

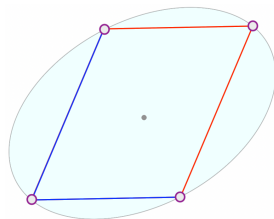
$\forall$  norm  $\eta$ , (1) wake-up tree of depth  $\leq \alpha(\eta)$  can be constructed in linear time; (2)  $\alpha(\eta) \leq 3 + \varphi \cdot \pi(\eta) < 9.48$ , where  $\varphi = \frac{1+\sqrt{5}}{2}$  and  $\pi(\eta) \in [3, 4]$ .



$$\alpha(\ell_2) \geq 1 + 2\sqrt{2} \approx 3.83$$



$$\alpha(\ell_1) \geq 1 + 2 \cdot 2 = 5$$



$\eta$  is a norm  $\Leftrightarrow$  unit  $\eta$ -disk is a centrally symmetric convex

$$\Lambda(\eta) := \sup_{\substack{u, v \in \mathbb{R}^2 \\ \eta(u), \eta(v) \leq 1}} \{ \eta(u+v) + \eta(u-v) \} \in [2, \pi(\eta)]$$

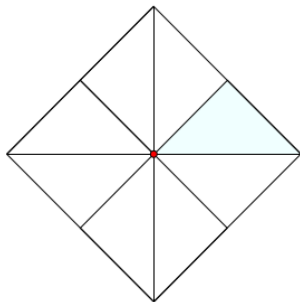
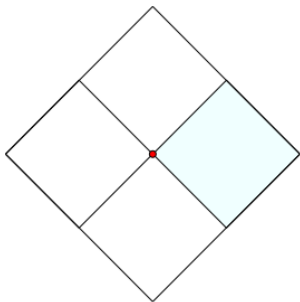
[  $\frac{1}{2}$ -perimeter of the largest inscribed parallelogram in the unit  $\eta$ -disk ]

### Theorem (Real 2D Normed Spaces)

$\forall$  norm  $\eta$ , the wake-up constant for  $(\mathbb{R}^2, \eta)$  is  $\alpha(\eta) \geq 1 + \Lambda(\eta)$ .

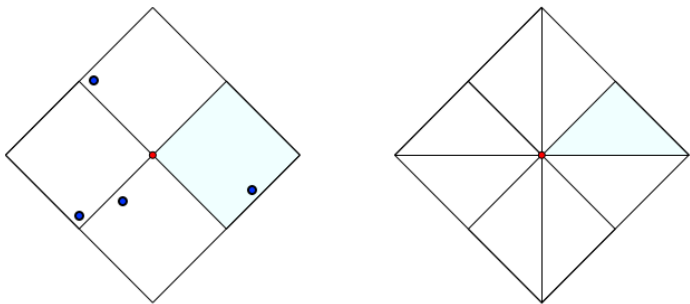
# Upper bound for $L_1$

**Strategy.** Depends on  $n_0 = \#$ robots in the densest square.



## Upper bound for $L_1$

Strategy. Depends on  $n_0 = \#$ robots in the densest square.

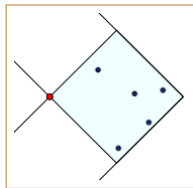


E.g.,  $n_0 = 1$ .

## Key lemmas for $n_0 > 1$

### Lemma (S5)

*One can wake up  $n_0 \leq 5$  robots in a square in time 2.*

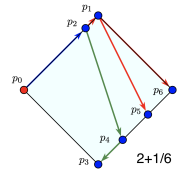
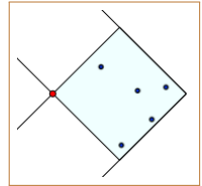




# Key lemmas for $n_0 > 1$

## Lemma (S5)

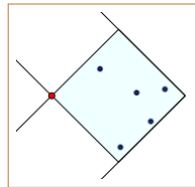
*One can wake up  $n_0 \leq 5$  robots in a square in time 2.*



# Key lemmas for $n_0 > 1$

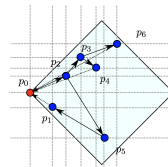
## Lemma (S5)

*One can wake up  $n_0 \leq 5$  robots in a square in time 2.*



## Lemma (S6)

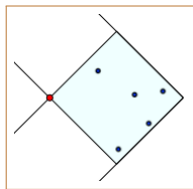
*One can wake up  $n_0 \leq 6$  robots in a square and gathering at the origin in time 3.*



# Key lemmas for $n_0 > 1$

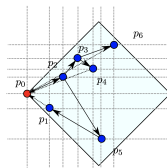
## Lemma (S5)

*One can wake up  $n_0 \leq 5$  robots in a square in time 2.*



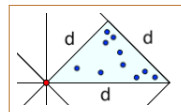
## Lemma (S6)

*One can wake up  $n_0 \leq 6$  robots in a square and gathering at the origin in time 3.*



## Lemma (T)

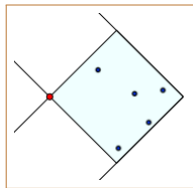
*One can wake up  $k$  robots in a isosceles right triangle with side  $d$  in time  $2d$ .*



# Key lemmas for $n_0 > 1$

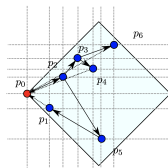
## Lemma (S5)

One can wake up  $n_0 \leq 5$  robots in a square in time 2.



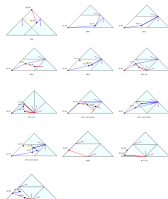
## Lemma (S6)

One can wake up  $n_0 \leq 6$  robots in a square and gathering at the origin in time 3.

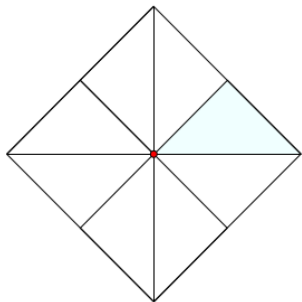
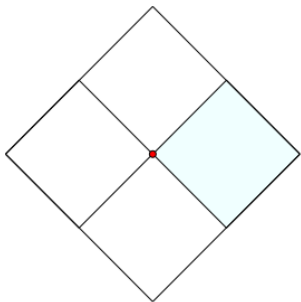


## Lemma (T)

One can wake up  $k$  robots in a isosceles right triangle with side  $d$  in time  $2d$ .



## Putting it all together



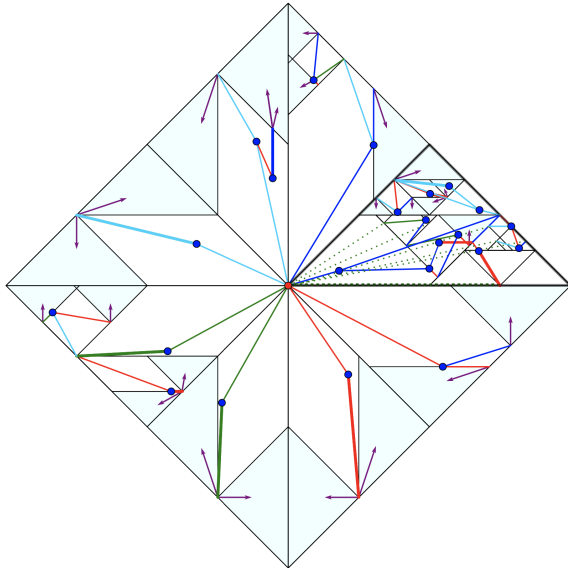
If  $n_0 = 1$ , use any strategy.

If  $n_0 = 2, 3, 4, 5$ , use Lemma (S5) twice.

If  $n_0 = 6, \boxed{7}, 8, 9, 10$ , use Lemma (S6), then Lemma (S5).

If  $n_0 \geq 11$ , use Lemma (T) with  $k \geq 6$ . (This case  $\Leftrightarrow n \geq 41$ .)

# Running the algorithm with Lemma (T)



# Conclusion & Problems

We have proved:

- $\alpha(\ell_1) = 1 + \Lambda(\ell_1) = 5$  ( $= \alpha(\ell_\infty)$ )
- $\forall \text{norm } \eta, 1 + \Lambda(\eta) \leq \alpha(\eta) < 9.48$
- Wake-up trees of depth  $\alpha(\eta)$  can be constructed in linear time

# Conclusion & Problems

We have proved:

- $\alpha(\ell_1) = 1 + \Lambda(\ell_1) = 5$  ( $= \alpha(\ell_\infty)$ )
- $\forall \text{norm } \eta, 1 + \Lambda(\eta) \leq \alpha(\eta) < 9.48$
- Wake-up trees of depth  $\alpha(\eta)$  can be constructed in linear time

Problems and Conjecture.

- Is  $\alpha(\ell_2) = 1 + \Lambda(\ell_2) = 1 + 2\sqrt{2}$ ?  $\alpha(\ell_2) \leq 7.07$  **4.63** [CCCG '24]  
True, if  $n > 280$  or in convex position. What about  $n = 8$ ?
- $\forall \eta, \alpha(\eta) = 1 + \Lambda(\eta)$ ?  
[ Conjecture. *Wake up  $n$  robots is quicker than wake up four.* ]

*"This problem is infectious and it is quite deceptively difficult."*  
– Anonymous referee.