

On the Freeze-Tag Problem

Cyril Gavaille joint work with
N. Bonichon, A. Casteigts, N. Hanusse



1st ANR TEMPOGRAL Meeting
Château du Clos de la Ribaudière – Chasseneuil-du-Poitou
November 24th, 2022

Context

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

Subject to:

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

Goal:

- to minimize the **time** to awake all the robots
- with a good schedule ...

Example

(1/3)

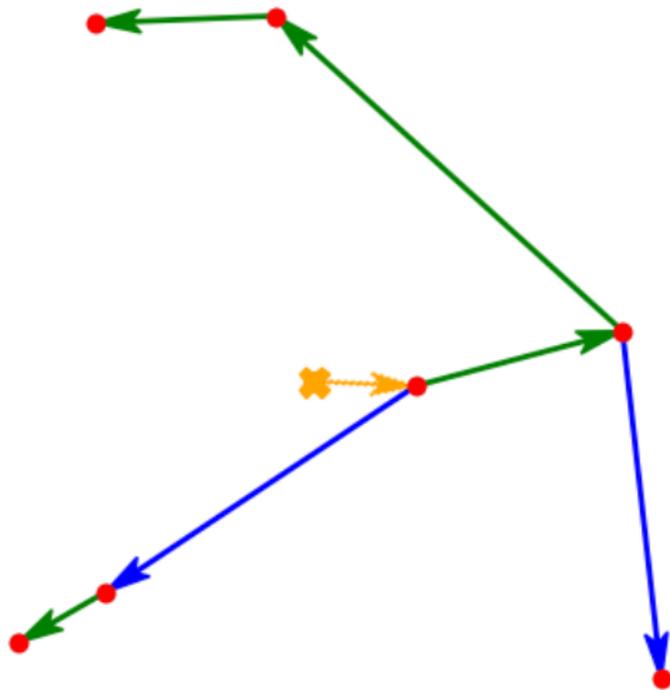
#pts = 8
diam = 1.890
ecc = 1.000



Example

(1/3)

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



Example

(2/3)

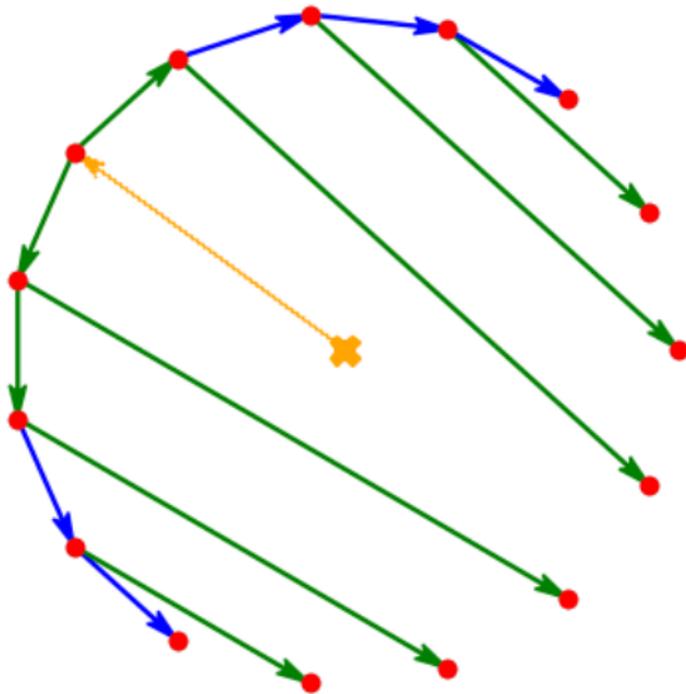
#pts = 16
diam = 1.989
ecc = 1.000



Example

(2/3)

optimal = 3.318
depth = 3.318
#pts = 16
diam = 1.989
ecc = 1.000
time = 207.710s



Example

(3/3)

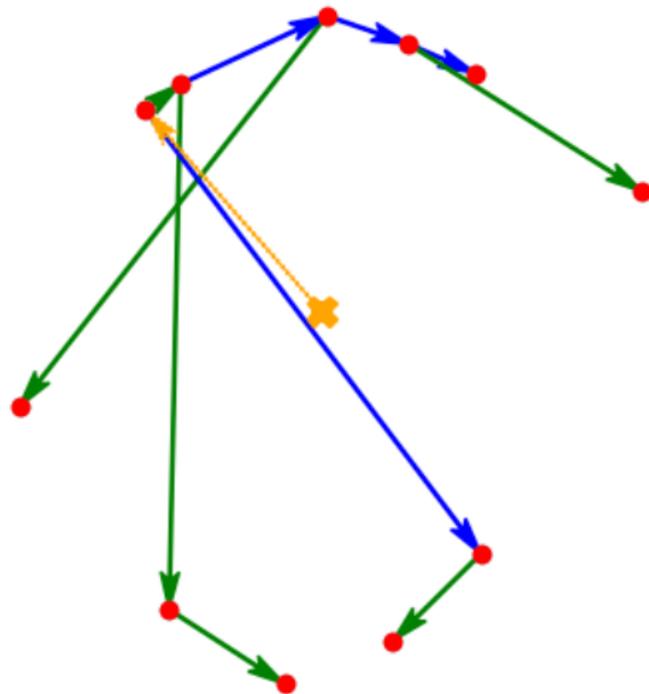


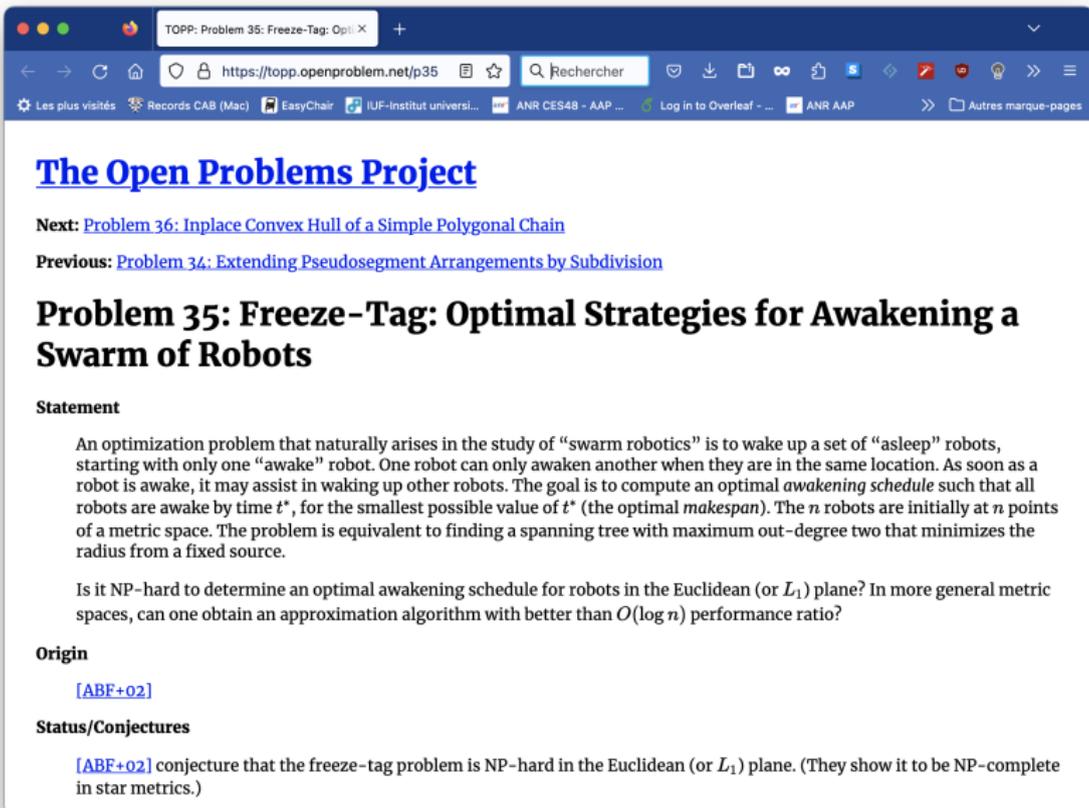
convex pts
#pts = 12
diam = 1.783
ecc = 1.000
time = 1.372s
seed = 3906

Example

(3/3)

optimal = 2.594
depth = 2.594
#pts = 12
diam = 1.783
ecc = 1.000
seed = 3906





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

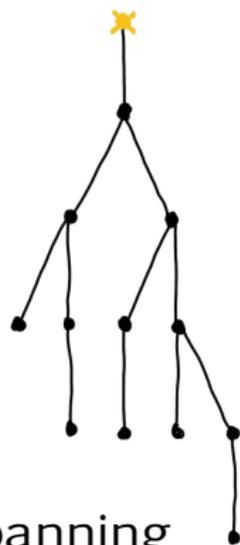
[\[ABF+02\]](#) conjecture that the freeze-tag problem is NP-hard in the Euclidean (or L_1) plane. (They show it to be NP-complete in star metrics.)

Problem Statement

FREEZE-TAG PROBLEM (basic)

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) of minimum depth w.r.t. ℓ_2 -norm



A **wake-up tree** for (s, P) is a binary tree spanning $\{s\} \cup P$ of root s which has at most 1 child.

Variants (space for moves): ℓ_p -norm, other metrics, weighted graph, non-metric, ...

Known Results: Hardness (1/3)

[ABF+06] NP-Hard for moves restricted to weighted star graphs, from NUMERICAL 3D MATCHING, and for unweighted trees, from 3-PARTITION (robots at leaves)

Known Results: Hardness (1/3)

- [ABF+06] NP-Hard for moves restricted to weighted star graphs, from NUMERICAL 3D MATCHING, and for unweighted trees, from 3-PARTITION (robots at leaves)
- [AY16,J17] NP-Hard for (\mathbb{R}^3, ℓ_2) , from HAMILTONIAN PATH IN 2D SUBGRID

Known Results: Hardness (1/3)

- [ABF+06] NP-Hard for moves restricted to weighted star graphs, from NUMERICAL 3D MATCHING, and for unweighted trees, from 3-PARTITION (robots at leaves)
- [AY16,J17] NP-Hard for (\mathbb{R}^3, ℓ_2) , from HAMILTONIAN PATH IN 2D SUBGRID
- [DR17] NP-Hard for (\mathbb{R}^3, ℓ_p) , $p > 1$, from DOMINATING SET

Known Results: Hardness (1/3)

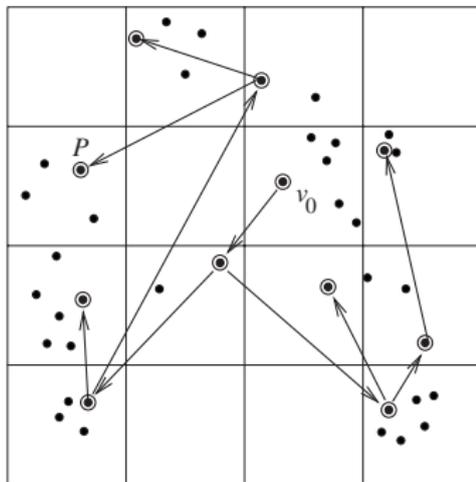
- [ABF+06] NP-Hard for moves restricted to weighted star graphs, from NUMERICAL 3D MATCHING, and for unweighted trees, from 3-PARTITION (robots at leaves)
- [AY16,J17] NP-Hard for (\mathbb{R}^3, ℓ_2) , from HAMILTONIAN PATH IN 2D SUBGRID
- [DR17] NP-Hard for (\mathbb{R}^3, ℓ_p) , $p > 1$, from DOMINATING SET
- [AAY17] NP-Hard for (\mathbb{R}^2, ℓ_2) , from MONOTONE 3SAT

Known Results: Approximation (2/3)

[ABF+06] $O(\log n)$ -approximation for locally-bounded weighted graphs (bounded aspect ratio for incident edges)

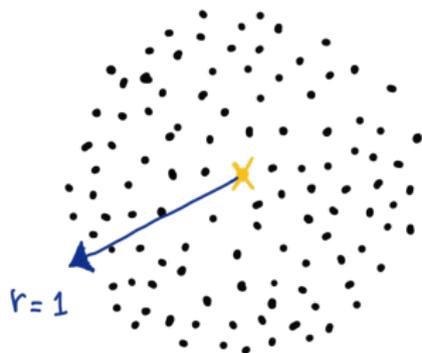
Known Results: Approximation (2/3)

- [ABF+06] $O(\log n)$ -approximation for locally-bounded weighted graphs (bounded aspect ratio for incident edges)
- [ABF+06] $(1 + \varepsilon)$ -approximation for (\mathbb{R}^d, ℓ_p) for fixed d and any p , in time $O(n \log n) + 2^{(1/\varepsilon)^{O(1)}}$ if $\varepsilon < 1/57$



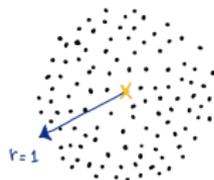
Known Results: Upper Bound (3/3)

[YBMK15] For (\mathbb{R}^2, ℓ_2) and unit source radius,
 \exists wake-up tree of depth < 10.1
computable in $O(n)$ time



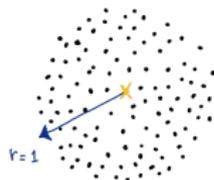
Contributions

For (\mathbb{R}^2, ℓ_2) and unit source radius



Contributions

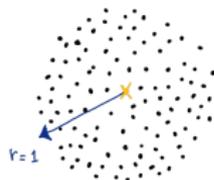
For (\mathbb{R}^2, ℓ_2) and unit source radius



- $\forall P$ in convex position, \exists wake-up tree of depth $\leq 1 + 2\sqrt{2} \approx 3.83$, achieved for $n = |P| = 4$

Contributions

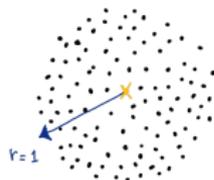
For (\mathbb{R}^2, ℓ_2) and unit source radius



- $\forall P$ in convex position, \exists wake-up tree of depth $\leq 1 + 2\sqrt{2} \approx 3.83$, achieved for $n = |P| = 4$
- $\forall P, \exists$ wake-up tree of depth $\leq 3 + o(1)$

Contributions

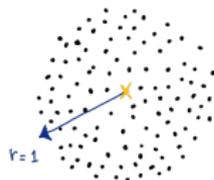
For (\mathbb{R}^2, ℓ_2) and unit source radius



- $\forall P$ in convex position, \exists wake-up tree of depth $\leq 1 + 2\sqrt{2} \approx 3.83$, achieved for $n = |P| = 4$
- $\forall P, \exists$ wake-up tree of depth $\leq 3 + o(1)$
- $\exists P, \forall$ wake-up tree has depth $\geq 3 + \Omega(1/n)$

Contributions

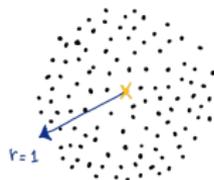
For (\mathbb{R}^2, ℓ_2) and unit source radius



- $\forall P$ in convex position, \exists wake-up tree of depth $\leq 1 + 2\sqrt{2} \approx 3.83$, achieved for $n = |P| = 4$
- $\forall P, \exists$ wake-up tree of depth $\leq 3 + o(1)$
- $\exists P, \forall$ wake-up tree has depth $\geq 3 + \Omega(1/n)$
- Exact algorithm running in time $3^n \cdot n^{O(1)}$
(holds for any non-metric space)

Contributions

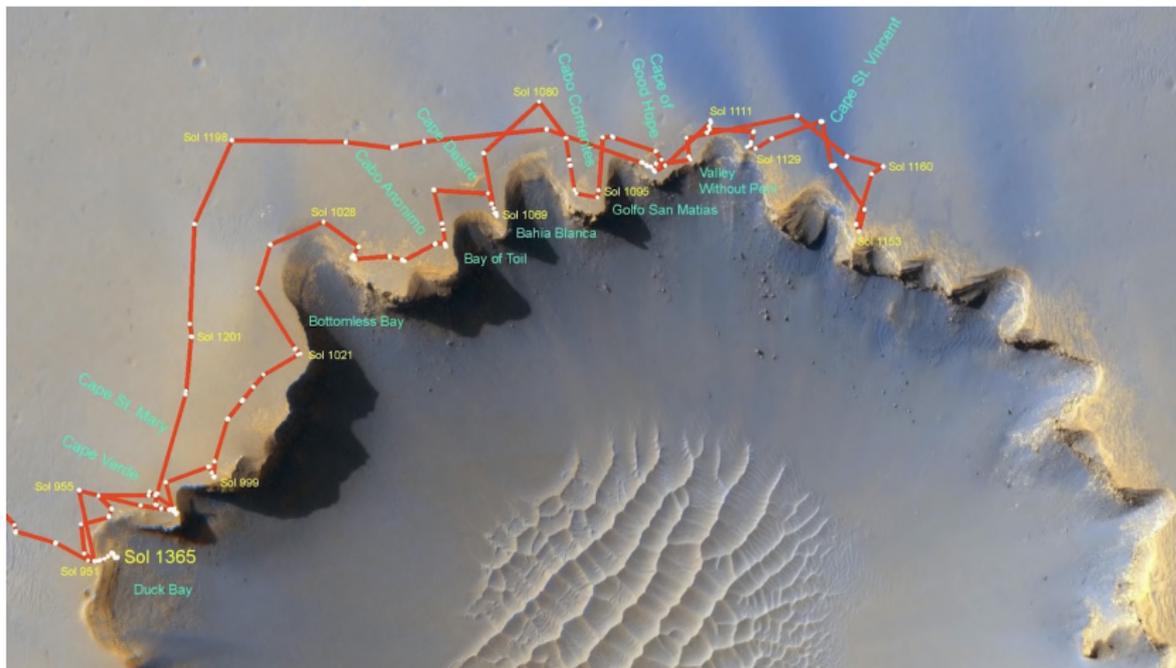
For (\mathbb{R}^2, ℓ_2) and unit source radius



- $\forall P$ in convex position, \exists wake-up tree of depth $\leq 1 + 2\sqrt{2} \approx 3.83$, achieved for $n = |P| = 4$
- $\forall P, \exists$ wake-up tree of depth $\leq 3 + o(1)$
- $\exists P, \forall$ wake-up tree has depth $\geq 3 + \Omega(1/n)$
- Exact algorithm running in time $3^n \cdot n^{O(1)}$
(holds for any non-metric space)
- Experiments, heuristics, ℓ_1 -norm, ...

Brute-Force Algorithms

Looks like to TSP (cycle / path)



$n!c^n$ outputs (= wake-up trees) to explore !

Dynamic Programming

FOR RELEASE:
A. M.'s, Thursday
January 2, 1964

FROM: International Business Machines Corp.
Data Processing Division
112 East Post Road
White Plains, New York

Bert Reisman
914 White Plains 9-1900

WHITE PLAINS, N. Y., Jan. 2 IBM mathematicians (left to right) Michael Held, Richard Shreshian and Richard M. Karp review the manual describing a new computer program which provides business and industry with a practical scientific method for handling a wide variety of complex scheduling tasks. The program, available to users of the IBM 7090 and 7094 data processing systems, consists of a set of 4,500 instructions which tell the computer what to do with data fed into it. It grew out of the trio's efforts to find solutions for a classic mathematical problem -- the "Traveling Salesman" problem -- which has long defied solution by man, or by the fastest computers he uses.



[Held-Karp'64] $O(n^2 2^n)$

Best known complexity for non-metric TSP

Recurrence & Memorization

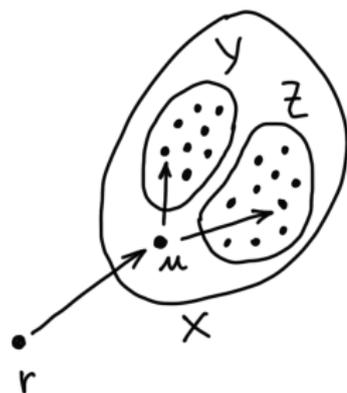
$\text{opt}(r, X, b)$:= optimal time to awake all asleep robots of $X \subseteq P$ from $r \notin X$ with $b \in \{1, 2\}$ awaked robots at r

$\text{opt}(s, P, 1)$?

Recurrence & Memorization

$\text{opt}(r, X, b) :=$ optimal time to awake all asleep robots of $X \subseteq P$ from $r \notin X$ with $b \in \{1, 2\}$ awaked robots at r

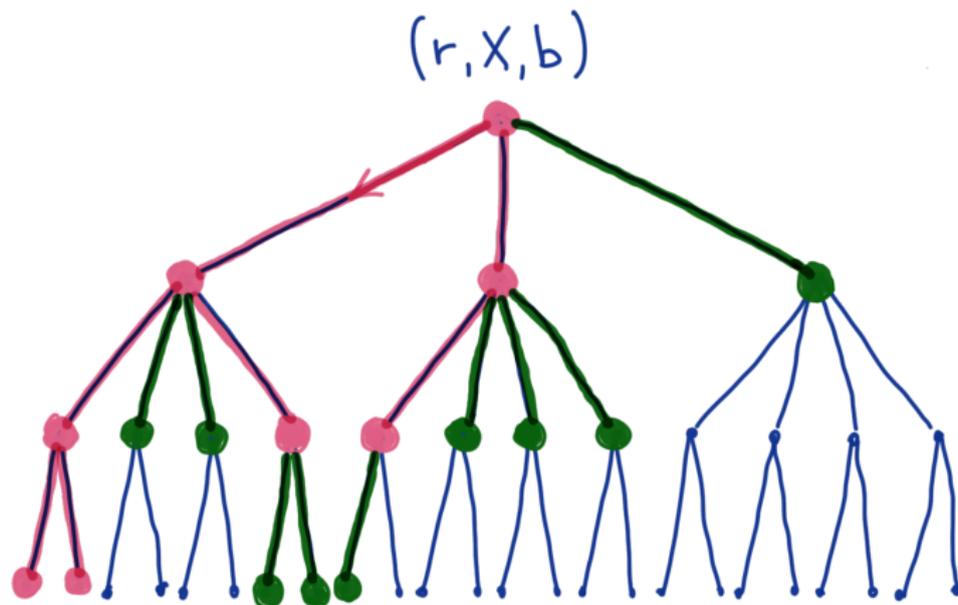
$\text{opt}(s, P, 1)?$



$\text{opt}(r, X, 1) = 0$ if $|X| = 0$

$\text{opt}(r, X, 1) = \min_{u \in X} \{ \text{dist}(r, u) + \text{opt}(u, X \setminus \{u\}, 2) \}$

$\text{opt}(u, X, 2) = \min_{X=Y \dot{\cup} Z} \max \{ \text{opt}(u, Y, 1), \text{opt}(u, Z, 1) \}$



$$[\#nodes \geq 2^n \cdot 2^{n-1} \cdot 2^{n-2} \dots \geq (n!2^n)^{n/\log n}]$$

$$\begin{aligned} \text{TIME}(r, X, b) &= \#visit(r, X, b) \times \text{lookup}(r, X, b) \\ &\quad + \#(r, X, b) \times \text{cost}(r, X, b) \end{aligned}$$

$$\begin{aligned} \text{TIME}(r, X, b) = & \# \text{visit}(r, X, b) \times \text{lookup}(r, X, b) \\ & + \#(r, X, b) \times \text{cost}(r, X, b) \end{aligned}$$

$$\text{TIME}(r, X, b) = \# \text{visit}(r, X, b) \times \text{lookup}(r, X, b) \\ + \#(r, X, b) \times \text{cost}(r, X, b)$$

$$\#(r, X, b) = (1 + n) \times \binom{n}{x} \times 2 \qquad x := |X|$$

$$\text{TIME}(r, X, b) = \# \text{visit}(r, X, b) \times \text{lookup}(r, X, b) \\ + \#(r, X, b) \times \text{cost}(r, X, b)$$

$$\#(r, X, b) = (1 + n) \times \binom{n}{x} \times 2 \\ \# \text{visit} < \#(r, X, b) \times \Delta$$

$$x := |X| \\ \Delta = 2^x$$

$$\text{TIME}(r, X, b) = \# \text{visit}(r, X, b) \times \text{lookup}(r, X, b) \\ + \#(r, X, b) \times \text{cost}(r, X, b)$$

$$\#(r, X, b) = (1 + n) \times \binom{n}{x} \times 2$$

$$\# \text{visit} < \#(r, X, b) \times \Delta$$

$$\text{lookup} = O(\log \#(r, X, b)) = O(n)$$

$$x := |X|$$

$$\Delta = 2^x$$

$$\text{TIME}(r, X, b) = \# \text{visit}(r, X, b) \times \text{lookup}(r, X, b) \\ + \#(r, X, b) \times \text{cost}(r, X, b)$$

$$\#(r, X, b) = (1 + n) \times \binom{n}{x} \times 2$$

$$x := |X|$$

$$\# \text{visit} < \#(r, X, b) \times \Delta$$

$$\Delta = 2^x$$

$$\text{lookup} = O(\log \#(r, X, b)) = O(n)$$

$$\text{cost} = O(\text{lookup} + x \cdot 2^x)$$

$$\text{TIME}(r, X, b) = \# \text{visit}(r, X, b) \times \text{lookup}(r, X, b) \\ + \#(r, X, b) \times \text{cost}(r, X, b)$$

$$\#(r, X, b) = (1 + n) \times \binom{n}{x} \times 2$$

$$x := |X|$$

$$\# \text{visit} < \#(r, X, b) \times \Delta$$

$$\Delta = 2^x$$

$$\text{lookup} = O(\log \#(r, X, b)) = O(n)$$

$$\text{cost} = O(\text{lookup} + x \cdot 2^x)$$

$$\max_{X \subseteq P} \left\{ 2^x \cdot \binom{n}{x} \right\} = 3^n / \Theta(\sqrt{n})$$

$$\text{TIME}(r, X, b) = \# \text{visit}(r, X, b) \times \text{lookup}(r, X, b) \\ + \#(r, X, b) \times \text{cost}(r, X, b)$$

$$\#(r, X, b) = (1 + n) \times \binom{n}{x} \times 2$$

$$x := |X|$$

$$\# \text{visit} < \#(r, X, b) \times \Delta$$

$$\Delta = 2^x$$

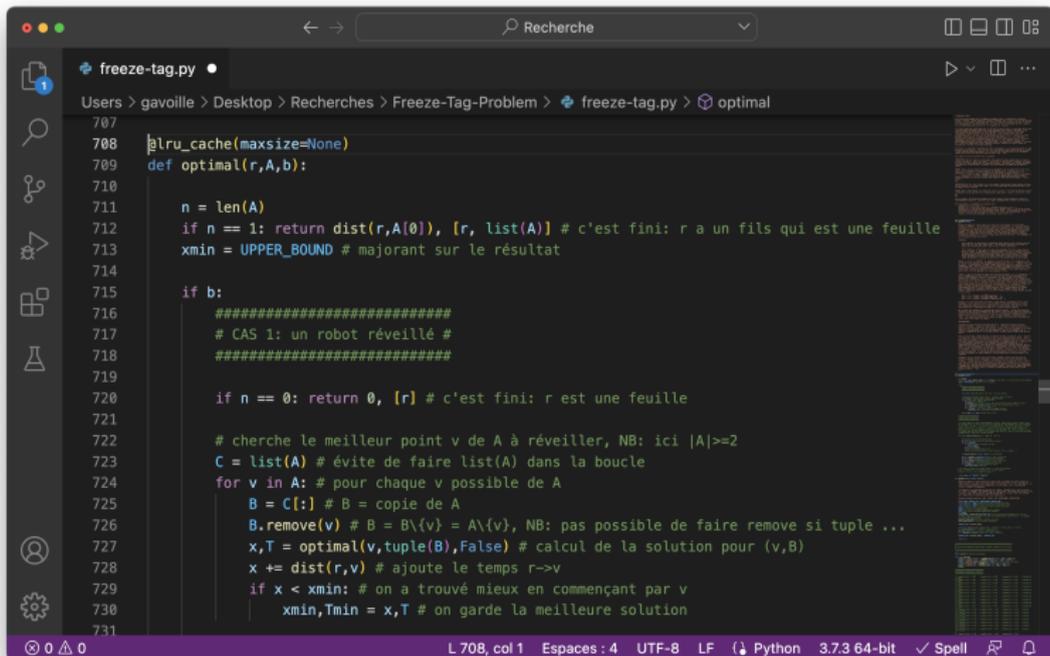
$$\text{lookup} = O(\log \#(r, X, b)) = O(n)$$

$$\text{cost} = O(\text{lookup} + x \cdot 2^x)$$

$$\max_{X \subseteq P} \left\{ 2^x \cdot \binom{n}{x} \right\} = 3^n / \Theta(\sqrt{n})$$

$$\Rightarrow \boxed{\text{TIME}(s, P, 1) = 3^n \cdot O(n^{3/2})}$$

Python



```
707
708 |lru_cache(maxsize=None)
709 def optimal(r,A,b):
710
711     n = len(A)
712     if n == 1: return dist(r,A[0]), [r, list(A)] # c'est fini: r a un fils qui est une feuille
713     xmin = UPPER_BOUND # majorant sur le resultat
714
715     if b:
716         #####
717         # CAS 1: un robot reveille #
718         #####
719
720     if n == 0: return 0, [r] # c'est fini: r est une feuille
721
722     # cherche le meilleur point v de A a reveiller, NB: ici |A|>=2
723     C = list(A) # evite de faire list(A) dans la boucle
724     for v in A: # pour chaque v possible de A
725         B = C[:] # B = copie de A
726         B.remove(v) # B = B \ {v} = A \ {v}, NB: pas possible de faire remove si tuple ...
727         x,T = optimal(v,tuple(B),False) # calcul de la solution pour (v,B)
728         x += dist(r,v) # ajoute le temps r->v
729         if x < xmin: # on a trouve mieux en commençant par v
730             xmin,Tmin = x,T # on garde la meilleure solution
731
```

In practice, 17 points takes 5'30" (Python) vs.
 $17! \times 10^{-9}s \approx 355,687s \approx 4.1$ days

Bounds on α_n

$\alpha_n :=$ infimum real st. every unit source radius and n point set has a wake-up tree of depth $\leq \alpha_n$ for (\mathbb{R}^2, ℓ_2)

Bounds on α_n

$\alpha_n :=$ infimum real st. every unit source radius and n point set has a wake-up tree of depth $\leq \alpha_n$ for (\mathbb{R}^2, ℓ_2)

$$\alpha_0 = 0$$

Bounds on α_n

$\alpha_n :=$ infimum real st. every unit source radius and n point set has a wake-up tree of depth $\leq \alpha_n$ for (\mathbb{R}^2, ℓ_2)

$$\alpha_0 = 0$$

$$\alpha_1 = 1$$

Bounds on α_n

$\alpha_n :=$ infimum real st. every unit source radius and n point set has a wake-up tree of depth $\leq \alpha_n$ for (\mathbb{R}^2, ℓ_2)

$$\alpha_0 = 0$$

$$\alpha_1 = 1$$

$$\alpha_2 = 3, \quad \forall n \geq 2, \alpha_n \geq 3$$

Bounds on α_n

$\alpha_n :=$ infimum real st. every unit source radius and n point set has a wake-up tree of depth $\leq \alpha_n$ for (\mathbb{R}^2, ℓ_2)

$$\alpha_0 = 0$$

$$\alpha_1 = 1$$

$$\alpha_2 = 3, \quad \forall n \geq 2, \alpha_n \geq 3$$

$$\alpha_3 = 3$$

Bounds on α_n

α_n := infimum real st. every unit source radius and n point set has a wake-up tree of depth $\leq \alpha_n$ for (\mathbb{R}^2, ℓ_2)

$$\alpha_0 = 0$$

$$\alpha_1 = 1$$

$$\alpha_2 = 3, \quad \forall n \geq 2, \alpha_n \geq 3$$

$$\alpha_3 = 3$$

$$\alpha_4 = 1 + 2\sqrt{2}$$

Bounds on α_n

α_n := infimum real st. every unit source radius and n point set has a wake-up tree of depth $\leq \alpha_n$ for (\mathbb{R}^2, ℓ_2)

$$\alpha_0 = 0$$

$$\alpha_1 = 1$$

$$\alpha_2 = 3, \quad \forall n \geq 2, \alpha_n \geq 3$$

$$\alpha_3 = 3$$

$$\alpha_4 = 1 + 2\sqrt{2}$$

$$\alpha_{5,6,7} \leq \alpha_4$$

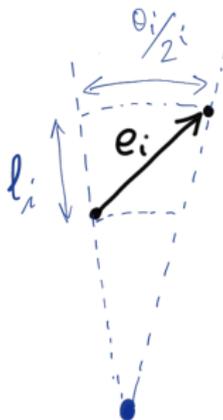
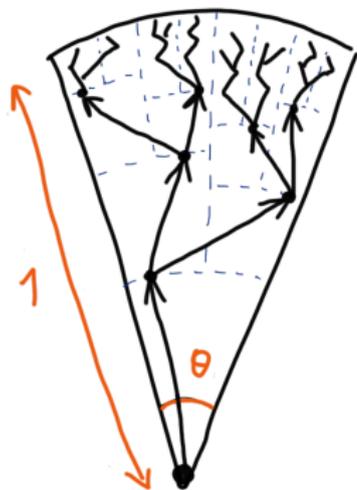
\vdots

$$\alpha_n < 5 + 2\sqrt{2} + \sqrt{5} \quad \text{from [YBMK15]}$$

Bounds on α_n

Theorem

$$\forall n \in \mathbb{N}, \quad \alpha_n \leq 3 + 8\pi/\sqrt{n}$$



$$\underbrace{\sum_i l_i}_1 + \underbrace{\sum_i \frac{\theta_i}{2^i}}_{2\theta}$$

The time to awake robots in a slope- θ cone from its apex is $\text{cone}(\theta) \leq 1 + 2\theta$

Bounds on α_n

Theorem

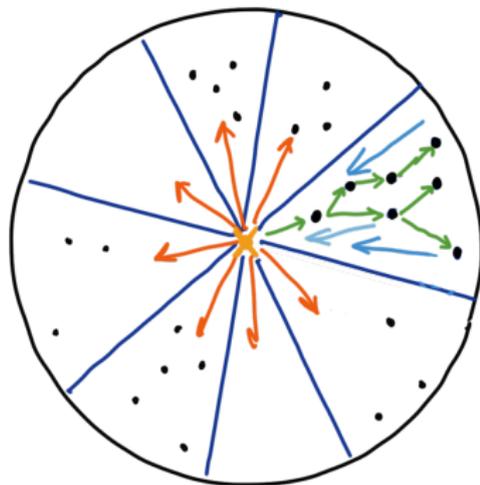
$$\forall n \in \mathbb{N}, \quad \alpha_n \leq 3 + 8\pi/\sqrt{n}$$

$$\Rightarrow \alpha_n \leq \text{cone}(2\pi) \leq 1 + 4\pi$$

Bounds on α_n

Theorem

$$\forall n \in \mathbb{N}, \quad \alpha_n \leq 3 + 8\pi/\sqrt{n}$$



- ① split in \sqrt{n} cones
- ① awake a dense cone
- ② go back to the source
- ③ awake all cones

$$\Rightarrow \alpha_n \leq 2\text{cone}(2\pi/\sqrt{n}) + 1 \leq 3 + 8\pi/\sqrt{n}$$

Open Problems

- $\alpha_n \leq \alpha_4$?
- $\alpha_{n,p} \leq \alpha_{4,p} = 1 + 2^{1+1/p}$?
- Polynomial time if P is convex?
- What if the moves are restricted to a temporal graph?
- What if the moves are restricted to other families? (path, binary tree, short cycles, ...)