

Distributed Freeze Tag: How to efficiently wake up a swarm of robots?

Cyril Gavoille, Nicolas Hanusse, Gabriel Le Boudier, Taïssir Marcé

ALGOTEL

June 6, 2025



université
de BORDEAUX

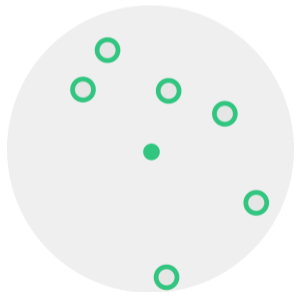
LaBRI

The Freeze Tag Problem: How to wake up a swarm of robots ?

[Arkin *et al.*, SODA'02]

Input: n sleeping robots + 1 awake

Robots move and wake up the others

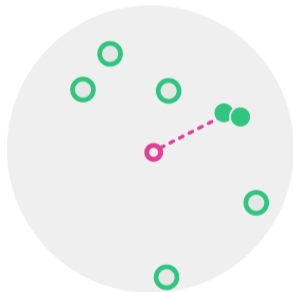


The Freeze Tag Problem: How to wake up a swarm of robots ?

[Arkin *et al.*, SODA'02]

Input: n sleeping robots + 1 awake

Robots move and wake up the others

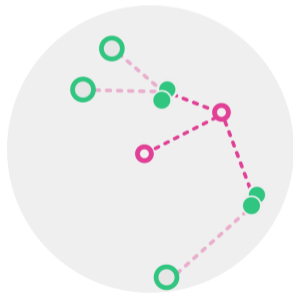


The Freeze Tag Problem: How to wake up a swarm of robots ?

[Arkin *et al.*, SODA'02]

Input: n sleeping robots + 1 awake

Robots move and wake up the others

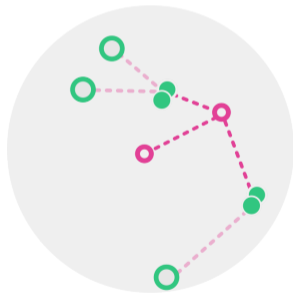


Optimal makespan t^* ?

The Freeze Tag Problem: How to wake up a swarm of robots ?

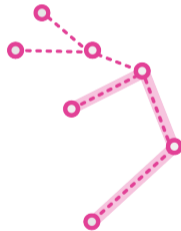
[Arkin *et al.*, SODA'02]

Input: n sleeping robots + 1 awake
Robots move and wake up the others



Optimal makespan t^* ?

Solution: Wake-up tree rooted in source
Makespan = weighted depth

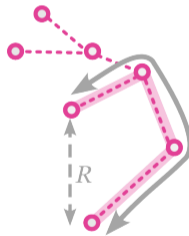
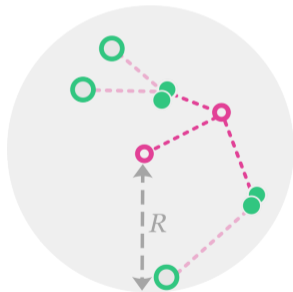


The Freeze Tag Problem: How to wake up a swarm of robots ?

[Arkin *et al.*, SODA'02]

Input: n sleeping robots + 1 awake
Robots move and wake up the others

Solution: Wake-up tree rooted in source
Makespan = weighted depth



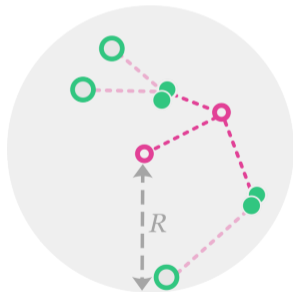
Optimal makespan t^* ?

👁 $t^* \geq \text{radius } R$

The Freeze Tag Problem: How to wake up a swarm of robots ?

[Arkin *et al.*, SODA'02]

Input: n sleeping robots + 1 awake
Robots move and wake up the others



Optimal makespan t^* ?

👁 $t^* \geq \text{radius } R$

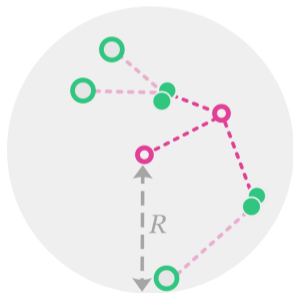
Thm. [Z. Abel, H. Akitaya, J. Yu, 2017]

Finding t^* is NP-hard

The Freeze Tag Problem: How to wake up a swarm of robots ?

[Arkin *et al.*, SODA'02]

Input: n sleeping robots + 1 awake
Robots move and wake up the others



Optimal makespan t^* ?

👁 $t^* \geq \text{radius } R$

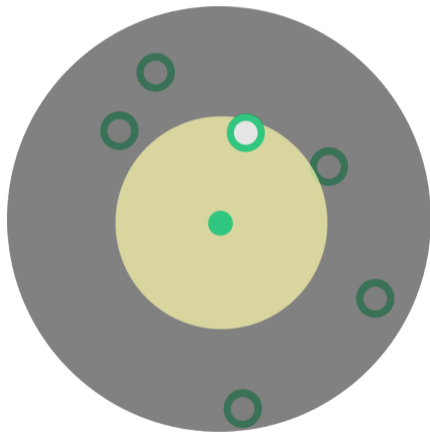
Thm.[Z. Abel, H. Akitaya, J. Yu, 2017]

Finding t^* is NP-hard

Thm.[Bonichon, Gavoille, Hanusse, Odak, CCCG '24]

Wake-up tree with depth at most $5R$ can be computed in time $O(n)$.

Distributed Freeze Tag

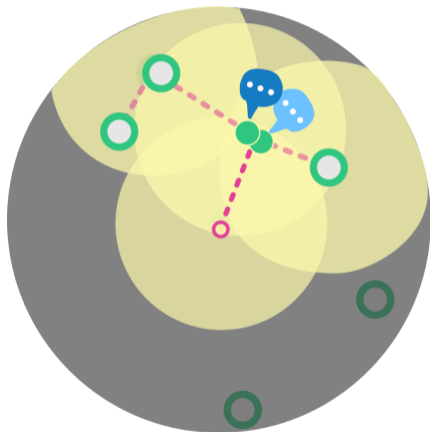


Distributed model

"What if robots ignore others' positions?"

- Detection at bounded distance
- No distant communication

Distributed Freeze Tag

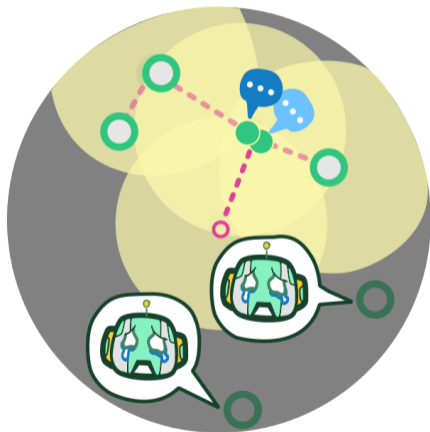


Distributed model

"What if robots ignore others' positions?"

- Detection at bounded distance
- No distant communication

Distributed Freeze Tag

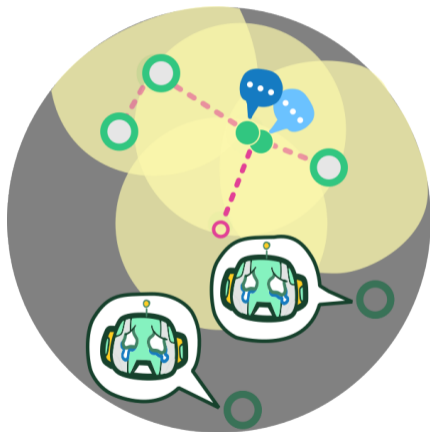


Distributed model

"What if robots ignore others' positions?"

- Detection at bounded distance : **robots may need to explore beyond unit distance**
- No distant communication

Distributed Freeze Tag



Distributed model

"What if robots ignore others' positions?"

- Detection at bounded distance : **robots may need to explore beyond unit distance**
- No distant communication
- 👁 Detecting termination requires knowledge on the set of robots (e.g. R or n)

Hypothesis and contribution



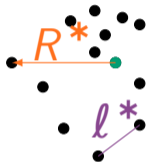
l -disk graph

There is an edge $(A, B) \iff |AB| \leq l$

Connectivity threshold

$l^* \equiv$ Smallest $l > 0$ such that the l -disk graph is connected.

Hypothesis and contribution



Algorithms input

An upper bound ℓ on ℓ^*

An upper bound R on the radius R^* .

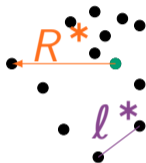
ℓ -disk graph

There is an edge $(A, B) \iff |AB| \leq \ell$

Connectivity threshold

$\ell^* \equiv$ Smallest $\ell > 0$ such that the ℓ -disk graph is connected.

Hypothesis and contribution



ℓ -disk graph

There is an edge $(A, B) \iff |AB| \leq \ell$

Connectivity threshold

$\ell^* \equiv$ Smallest $\ell > 0$ such that the ℓ -disk graph is connected.

Algorithms input

An upper bound ℓ on ℓ^*

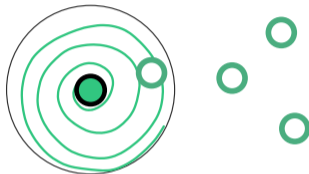
An upper bound R on the radius R^* .

Theorem

[Gavoille, Hanusse, Le Boudier, M.]

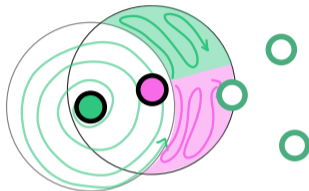
Given n, R and ℓ it takes $\Theta(R + \ell^2 \log(R/\ell))$ time to awake robots

Depth-First Search



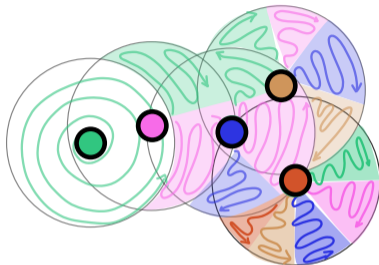
1 robots explores an ℓ -disk in time $O(\ell^2)$

Depth-First Search



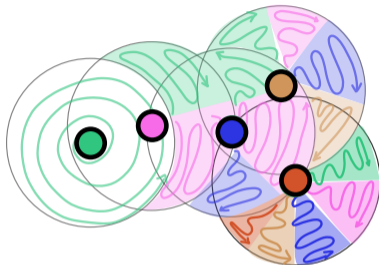
2 robots explores an ℓ -disk in time $O(\ell + \ell^2/2)$

Depth-First Search



ℓ robots explores an ℓ -disk in time $O(\ell + \ell^2/\ell) = O(\ell)$.
(No more speed up beyond ℓ)

Depth-First Search



ℓ robots explores an ℓ -disk in time $O(\ell + \ell^2/\ell) = O(\ell)$.

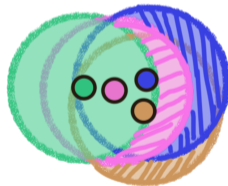
(No more speed up beyond ℓ)

Efficient for waking up a team of ℓ robots

$$\sum_{i=1}^{\ell} (\ell + \ell^2/i) = O(\ell^2 \log \ell)$$

Depth-First Sampling

Goal: Whenever you get a large team ($\approx O(\ell)$ robots), **grow to 4ℓ** using **sampling** to cover surface.

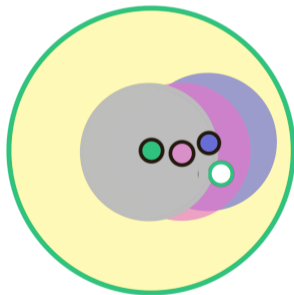


Depth-First Sampling

Goal: Whenever you get a large team ($\approx O(\ell)$ robots), **grow to 4ℓ** using **sampling** to cover surface.

Sampling Method

- Exploration at distance 2ℓ
- Recruited robots are at distance $> \ell$ apart



Depth-First Sampling

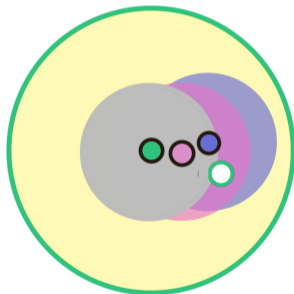
Goal: Whenever you get a large team ($\approx O(\ell)$ robots), **grow to 4ℓ** using **sampling** to cover surface.

Sampling Method

- Exploration at distance $2\ell \Rightarrow O(\ell^2/\ell)$
- Recruited robots are at distance $> \ell$ apart

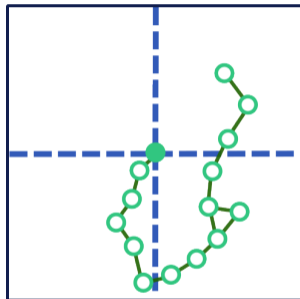
Lemma

DFSampling stops within time $O(R + \ell^2)$ and either a team of 4ℓ robots were recruited, or the whole area is covered by recruited robots.



Geometric Separator

Goal: As soon as you get a large team ($\approx 4\ell$ robots), **Divide** the surface and conquer!

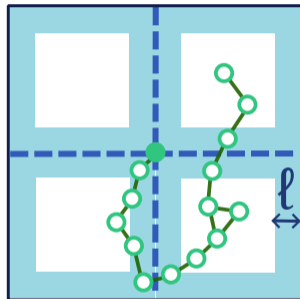


Geometric Separator

Goal: As soon as you get a large team ($\approx 4\ell$ robots), **Divide** the surface and conquer!

Partitioning method

Four ℓ -team explore four **Geometric Separators** of width $\ell \Rightarrow O(R)$.



Geometric Separator

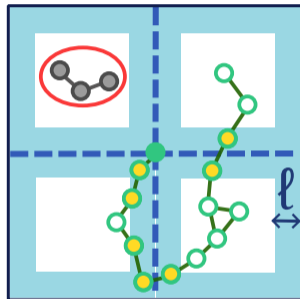
Goal: As soon as you get a large team ($\approx 4\ell$ robots), **Divide** the surface and conquer!

Partitioning method

Four ℓ -team explore four **Geometric Separators** of width $\ell \Rightarrow O(R)$.

Cool trick

Detects if a large area is empty (since ℓ -disk graph is connected)



Algorithm Overview

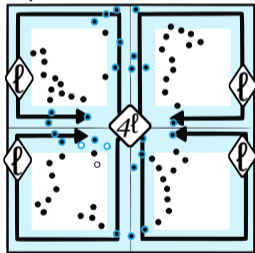
Initialization: Recruits $O(\ell)$ robots
 $O(R + \ell^2 \log \ell)$

Algorithm Overview

Initialization: Recruits $O(\ell)$ robots
 $O(R + \ell^2 \log \ell)$

Partition:

ℓ -teams explores
separators.

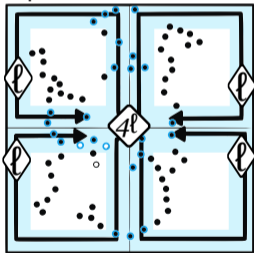


$O(R_i)$
with $R_i = R/2^i$

Algorithm Overview

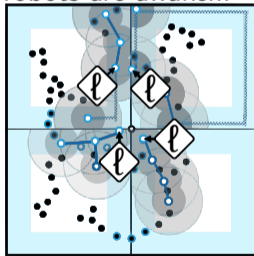
Initialization: Recruits $O(\ell)$ robots
 $O(R + \ell^2 \log \ell)$

Partition:
 ℓ -teams explores
separators.



$O(R_i)$
with $R_i = R/2^i$

Recruitment:
 ℓ -teams recruits until 4ℓ
robots are awoken.

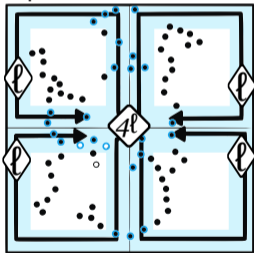


With sampling $O(R + \ell^2)$

Algorithm Overview

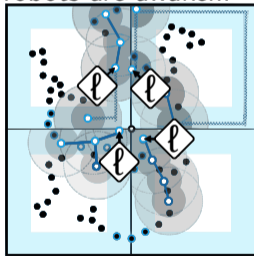
Initialization: Recruits $O(\ell)$ robots
 $O(R + \ell^2 \log \ell)$

Partition:
 ℓ -teams explores
separators.



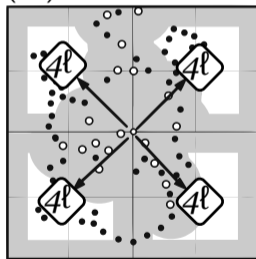
$O(R_i)$
with $R_i = R/2^i$

Recruitment:
 ℓ -teams recruits until 4ℓ
robots are awoken.



With sampling $O(R + \ell^2)$

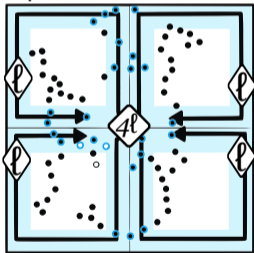
Iteration:
repeat until
 $(R_i)^2 \leq \ell^3$



Algorithm Overview

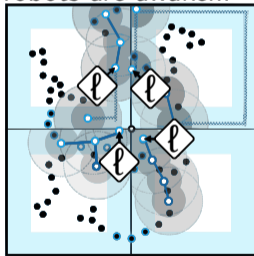
Initialization: Recruits $O(\ell)$ robots
 $O(R + \ell^2 \log \ell)$

Partition:
 ℓ -teams explores
separators.



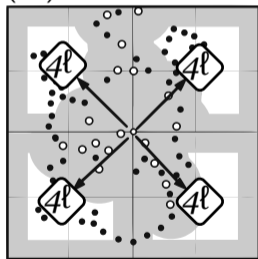
$O(R_i)$
with $R_i = R/2^i$

Recruitment:
 ℓ -teams recruits until 4ℓ
robots are awoken.



With sampling $O(R + \ell^2)$

Iteration:
repeat until
 $(R_i)^2 \leq \ell^3$



$O(\log R / \ell^{3/2})$ iterations

Conclusion

✓ Optimal approximation algorithm for the Distributed Freeze-Tag

Conclusion

✓ Optimal approximation algorithm for the Distributed Freeze-Tag

Related results

DFTP with **bounded energy** per robots: nearly optimal

[Gavoille, Hanusse, Le Boudier, M - PODC'25]

Conclusion

✓ Optimal approximation algorithm for the Distributed Freeze-Tag

Related results

DFTP with **bounded energy** per robots: nearly optimal

[Gavoille, Hanusse, Le Boudier, M - PODC'25]

Further works

- Several sources and/or different spatio-temporal reference
- Under asynchronous assumptions

Conclusion

✓ Optimal approximation algorithm for the Distributed Freeze-Tag

Related results

DFTP with **bounded energy** per robots: nearly optimal

[Gavoille, Hanusse, Le Boudier, M - PODC'25]

Further works

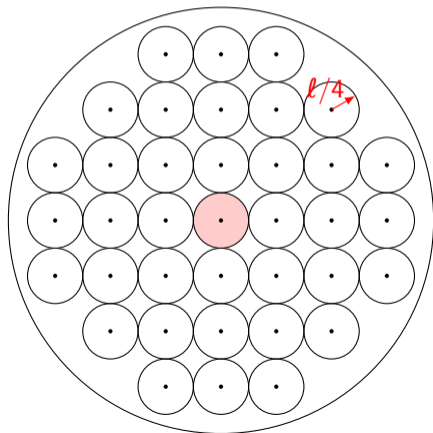
- Several sources and/or different spatio-temporal reference
- Under asynchronous assumptions

Thanks!

Lower Bound: Scheme of the construction

Goal

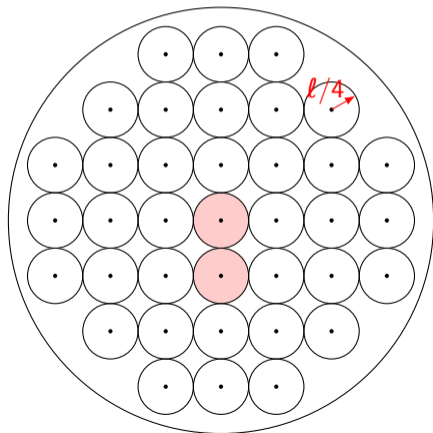
Given an algorithm \mathcal{A} with inputs R, ℓ, n , produce an n -point set with radius at most R and connectivity threshold at most ℓ such that \mathcal{A} has makespan $\Omega(R + \ell^2 \log R/\ell)$.



Lower Bound: Scheme of the construction

Goal

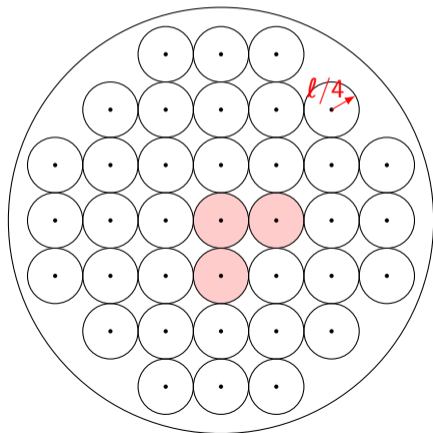
Given an algorithm \mathcal{A} with inputs R, ℓ, n , produce an n -point set with radius at most R and connectivity threshold at most ℓ such that \mathcal{A} has makespan $\Omega(R + \ell^2 \log R/\ell)$.



Lower Bound: Scheme of the construction

Goal

Given an algorithm \mathcal{A} with inputs R, ℓ, n , produce an n -point set with radius at most R and connectivity threshold at most ℓ such that \mathcal{A} has makespan $\Omega(R + \ell^2 \log R/\ell)$.



Lower Bound: Scheme of the construction

Goal

Given an algorithm \mathcal{A} with inputs R, ℓ, n , produce an n -point set with radius at most R and connectivity threshold at most ℓ such that \mathcal{A} has makespan $\Omega(R + \ell^2 \log R/\ell)$.

