Approximation Algorithms
Chapter 5: Freeze Tag and Scan Cover

Sándor P. Fekete

Algorithms Division
Department of Computer Science
TU Braunschweig

1. Introduction
2. Freeze Tag
3. Angular Freeze Tag
4. Angular Scan Cover

1. Introduction
2. Freeze Tag
3. Angular Freeze Tag
4. Angular Scan Cover

Freeze Tag

Algorithmica (2006) 46: 193–221
DOI: 10.1007/s00453-006-1206-1

Algorithmica

The Freeze-Tag Problem: How to Wake Up a Swarm of Robots[1]

Esther M. Arkin,[2] Michael A. Bender,[3] Sándor P. Fekete,[4]
Joseph S. B. Mitchell,[2] and Martin Skutella[5]

Abstract. An optimization problem that naturally arises in the study of swarm robotics is the *Freeze-Tag Problem* (*FTP*) of how to awaken a set of "asleep" robots, by having an awakened robot move to their locations. Once a robot is awake, it can assist in awakening other slumbering robots. The objective is to have all robots awake as early as possible. While the FTP bears some resemblance to problems from areas in combinatorial optimization such as routing, broadcasting, scheduling, and covering, its algorithmic characteristics are surprisingly different.

We consider both scenarios on graphs and in geometric environments. In graphs, robots sleep at vertices and there is a length function on the edges. Awake robots travel along edges, with time depending on edge length. For most scenarios, we consider the offline version of the problem, in which each awake robot knows the position of all other robots. We prove that the problem is NP-hard, even for the special case of star graphs. We also establish hardness of approximation, showing that it is NP-hard to obtain an approximation factor better than $\frac{5}{3}$, even for graphs of bounded degree.

These lower bounds are complemented with several positive algorithmic results, including:

- We show that the natural greedy strategy on star graphs has a tight worst-case performance of $\frac{7}{3}$ and give a polynomial-time approximation scheme (PTAS) for star graphs.
- We give a simple $O(\log \Delta)$-competitive online algorithm for graphs with maximum degree Δ and locally bounded edge weights.
- We give a PTAS, running in nearly linear time, for geometrically embedded instances.

Technische Universität Braunschweig

The Freeze-Tag Problem: How to Wake Up a Swarm of Robots

Estie Arkin[1]
Michael Bender[1]
Sándor Fekete[2]
Joe Mitchell[1]
Martin Skutella[3]

[1] University at Stony Brook
[2] TU Braunschweig
[3] TU Berlin

Freeze Tag!:

Given: n robots at points in a metric space

$n-1$ robots are "asleep"

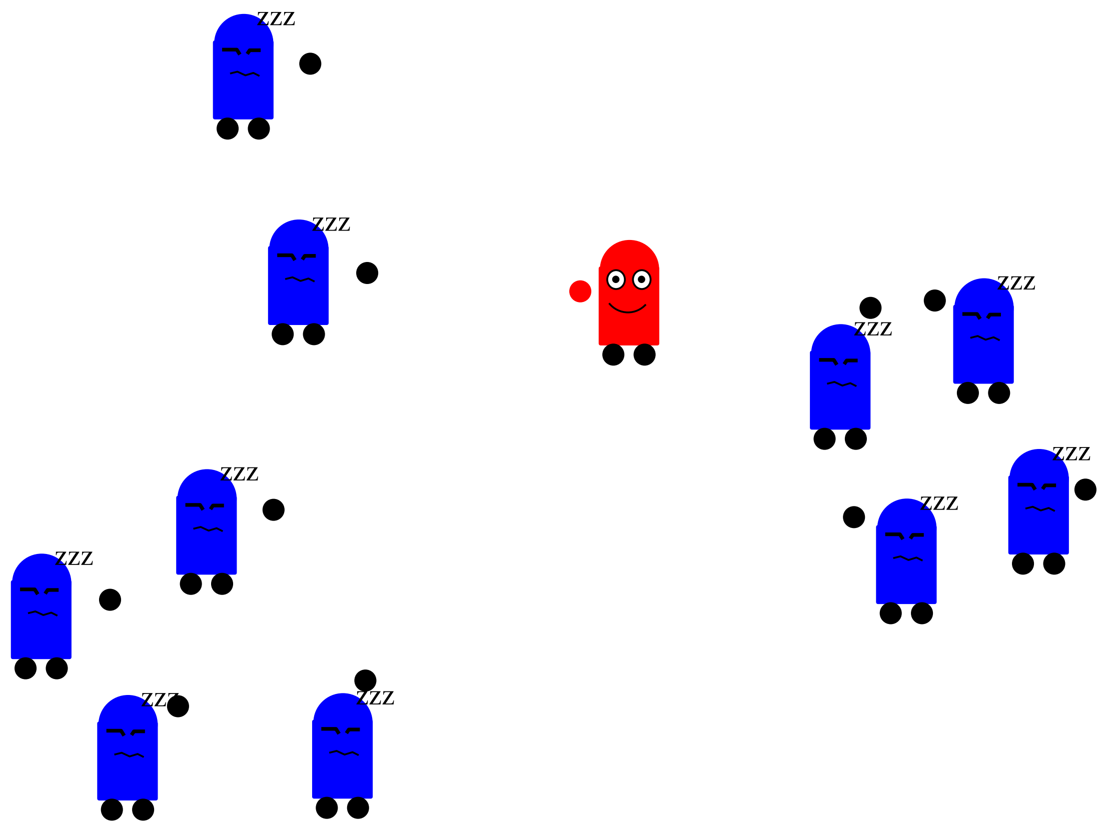
1 robot is awake

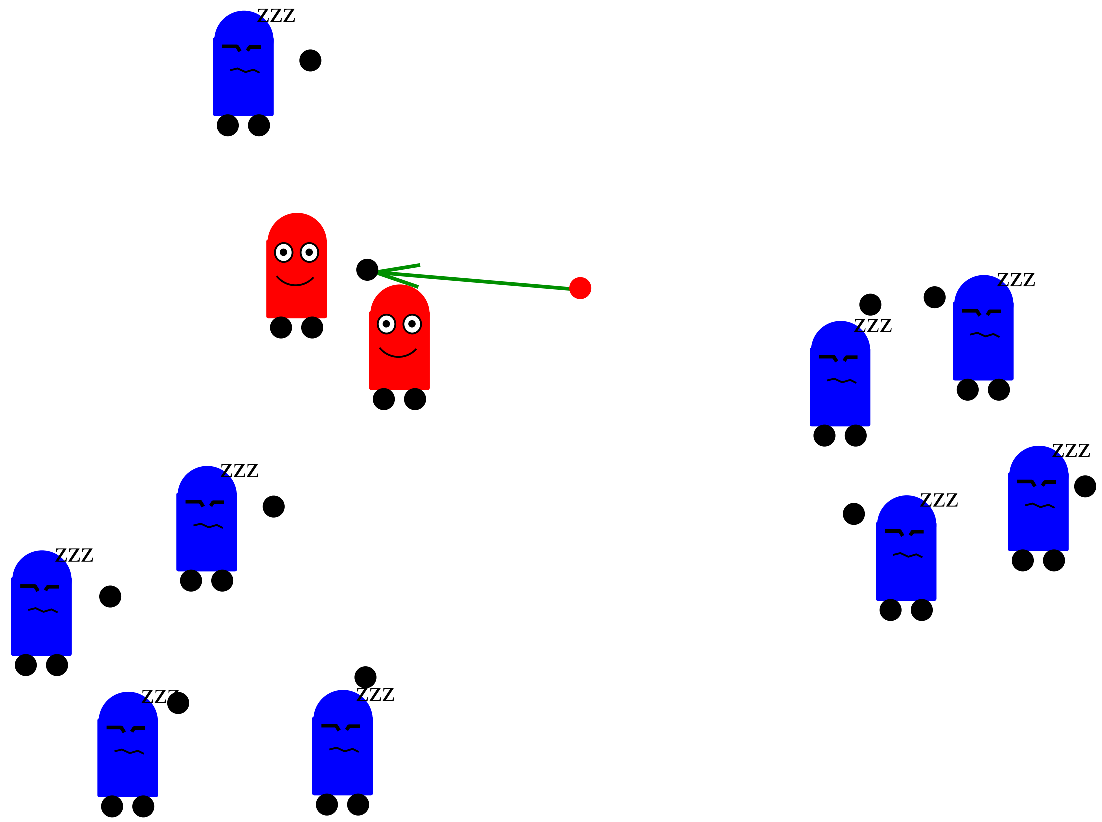
An awake robot "wakes up" a sleeping robot at point p by going to p

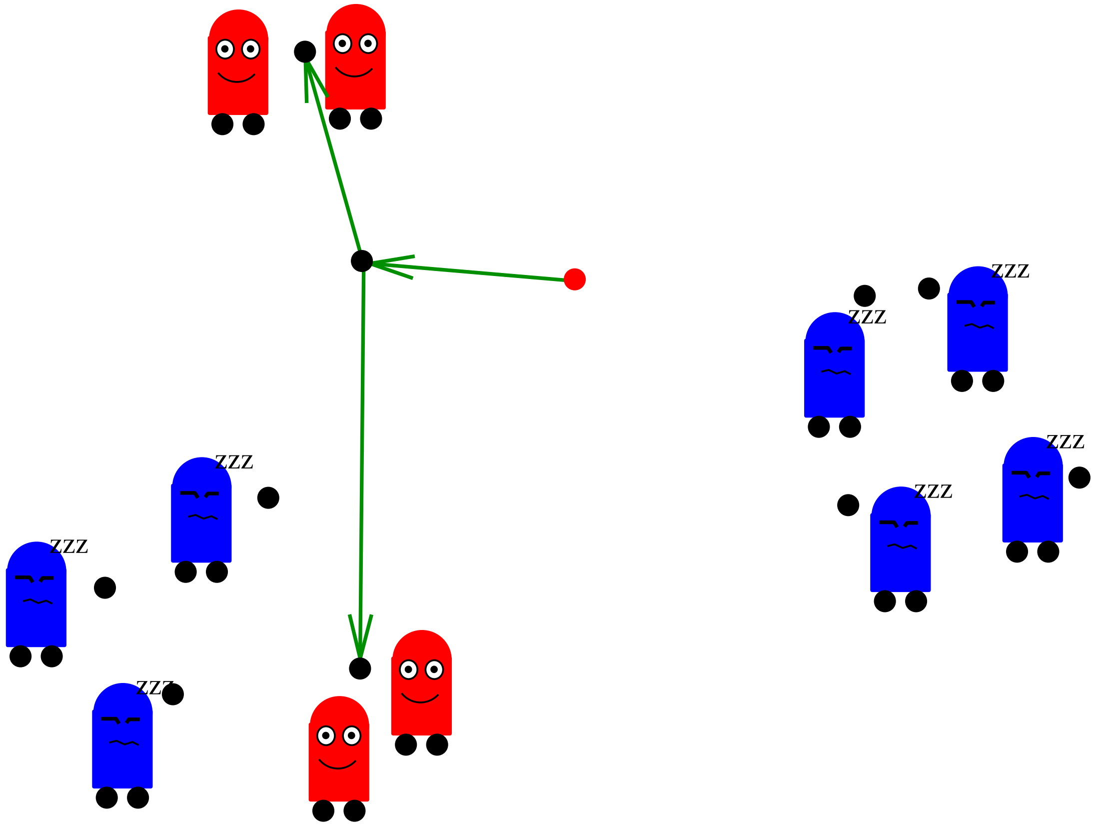
As robots wake up, there are more robots to assist in waking up others

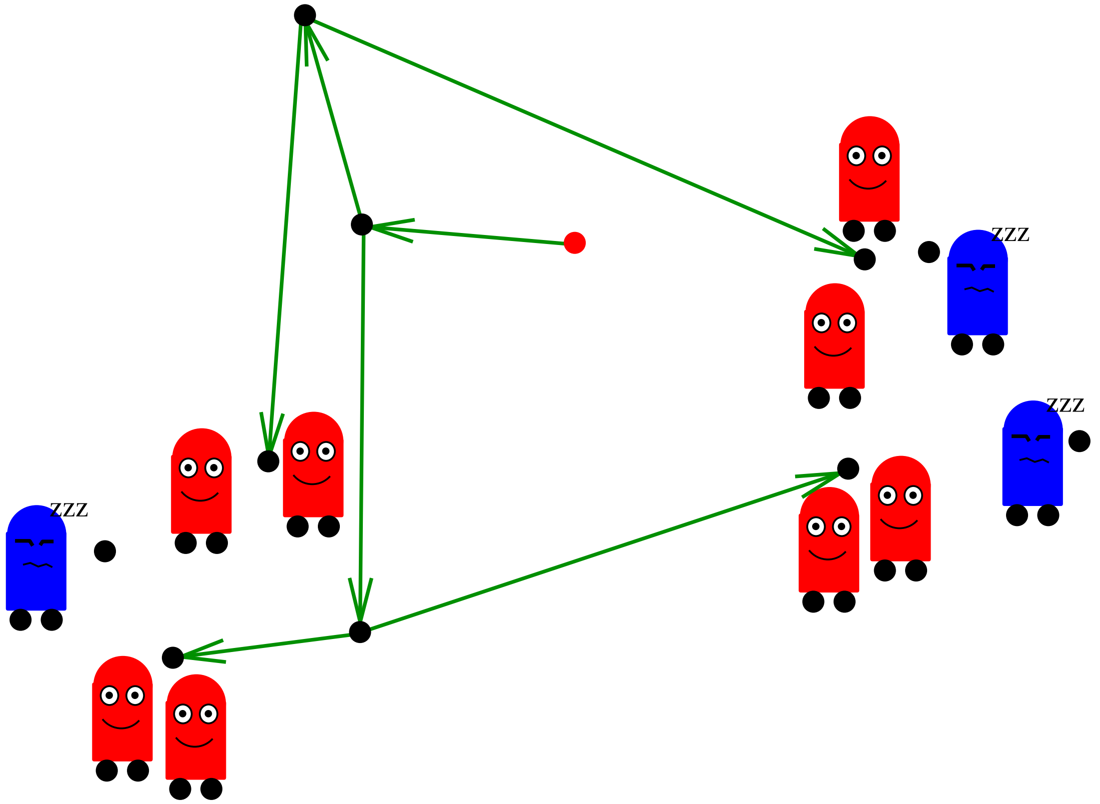
Goal: Wake up all robots as soon as possible

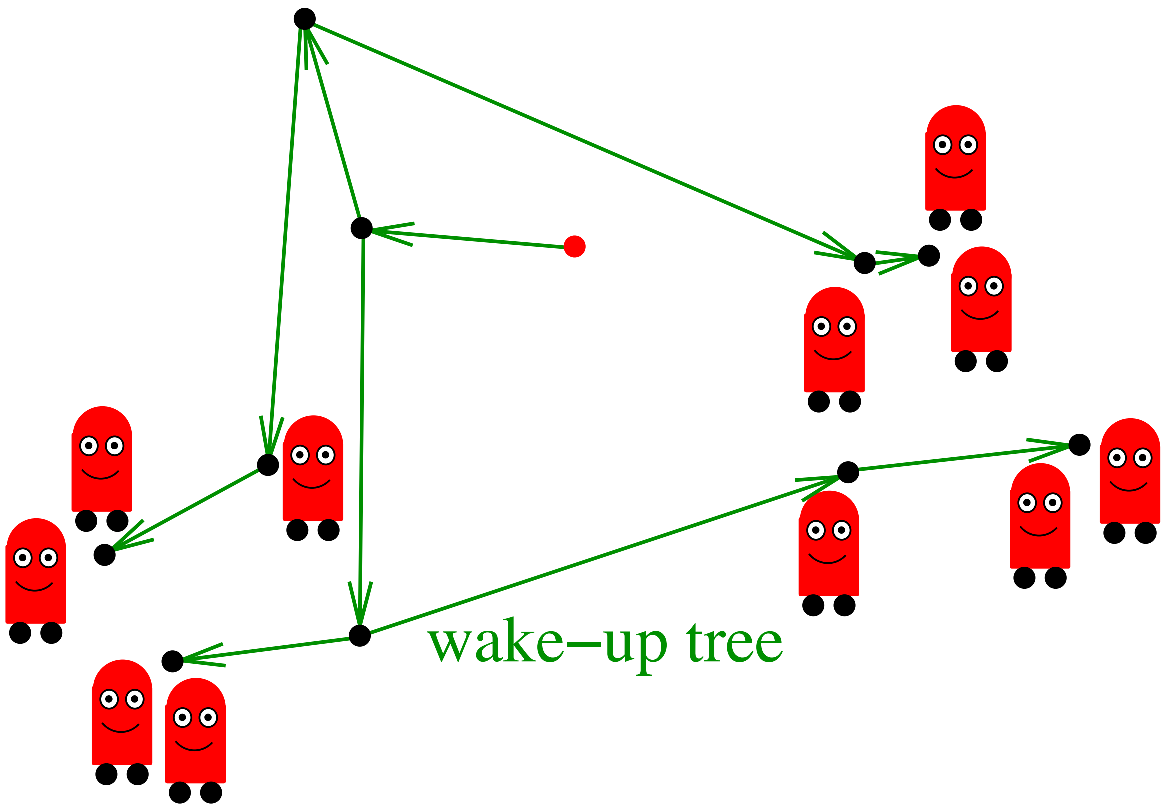
Minimize makespan


zzz
zzz
zzz
zzz
zzz
zzz
zzz
zzz
zzz
zzz


zzz
zzz
zzz
zzz
zzz
zzz
zzz
zzz
zzz
zzz


zzz
zzz
zzz
zzz
zzz
zzz
zzz
zzz


zzz
zzz
zzz


wake–up tree

Other Motivations:

• Distribute data (or other commodity) to agents, where physical proximity is necessary for transmittal

• Secret sharing by whisper

• Natural network optimization problem:

Minimize the length of a root-to-leaf path in a *binary* spanning tree

Related Work:

Dissemination of data in graphs:

- minimum broadcast time problem
- multicast problem
- minimum gossip time problem

Key differences from FTP:

- messages sent along edges of graph (no need for proximity)
- broadcast problem poly in trees, but FTP is NP-hard even for stars

Simple Approximation Bounds:

Any "brain-dead" strategy gives $O(\log n)$-approx:

- Source robot awakens one other robot

  (travelling distance $\leq D$, diameter)

  Now 2 awake robots.

- Each travels to a distinct other asleep robot (dist $\leq D$), awakens it and waits (if necessary) for the other robot to reach its destination

  Now 4 awake robots.

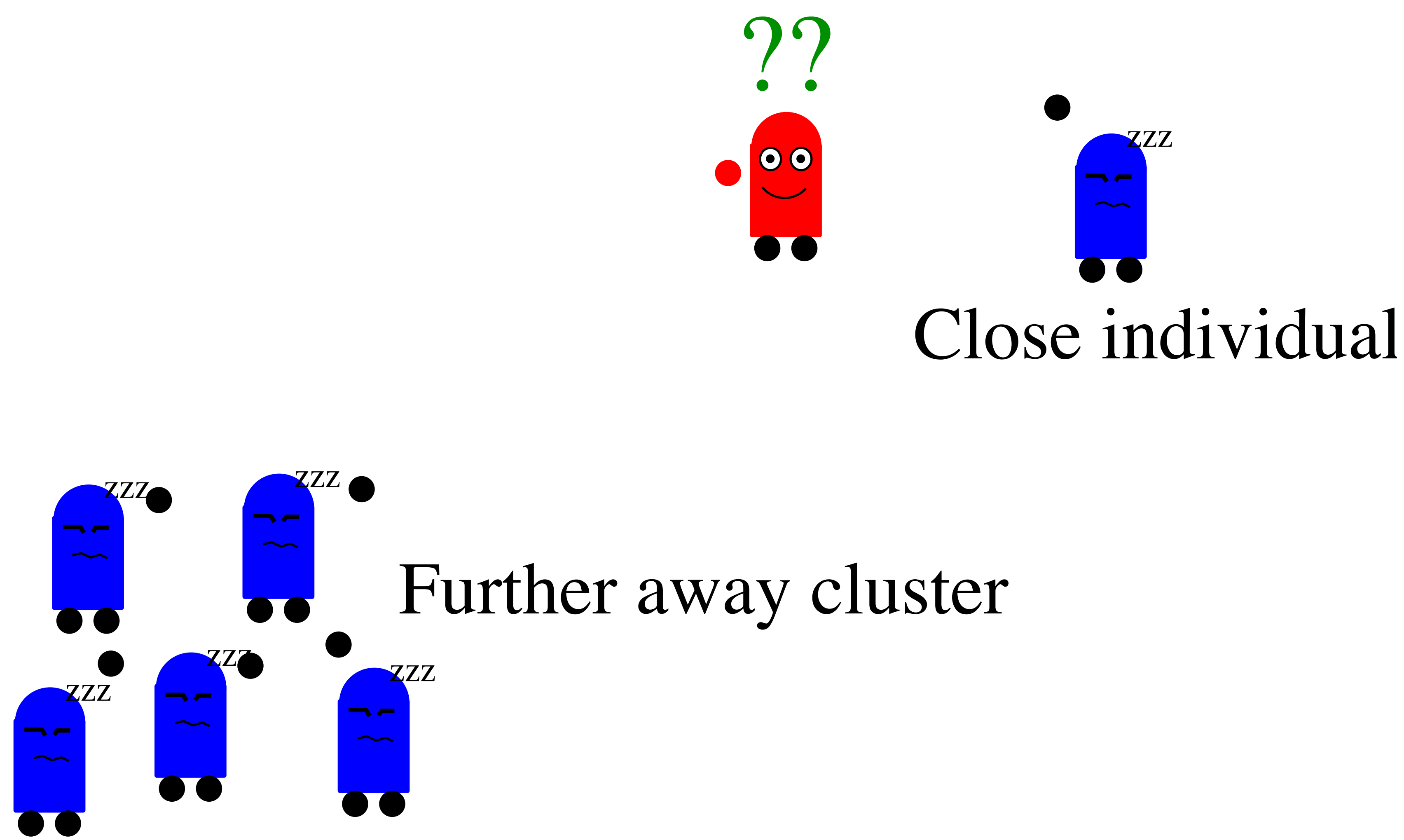
- Etc, etc, $\log n$ rounds, each of length $\leq D$

- Lower bound on OPT: $t^* \geq R_0 = \max_{p_i} d(v_0, p_i) \geq D/2$

OPEN: Is there an $o(\log n)$ approximation algorithm?

Fundamental Question:

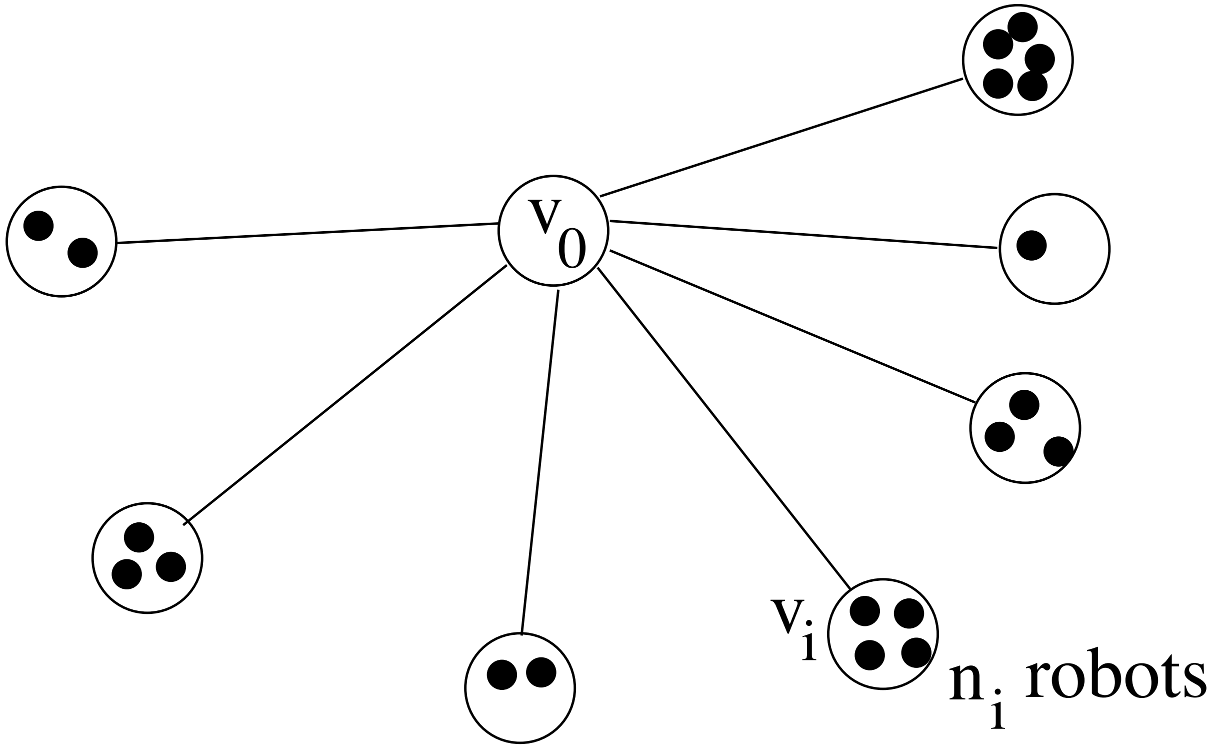
Whether to awaken a nearby robot or go further to (start to) awaken a larger swarm?



Summary of Results:

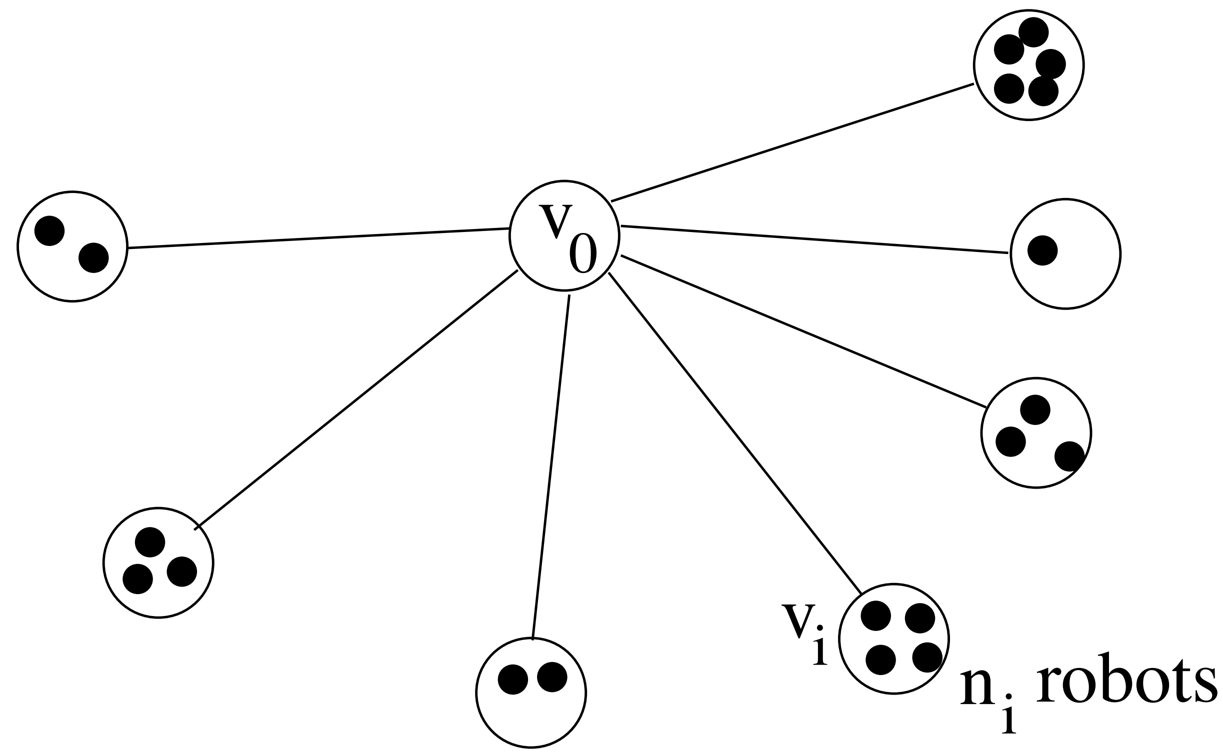
1. FTP is NP-hard, even for stars, with one robot per leaf
2. $O(1)$-approx for (general) stars
3. PTAS for stars, same number of robots at each leaf
4. Tight analysis of greedy heuristic on stars: 7/3-approx
5. $o(\log n)$-approx for FTP in ultrametrics ($2^{O(\sqrt{\log\log n})}$-approx)
6. Simple linear-time on-line algorithm, $O(\log\Delta)$-competitive
   ($\Delta \leq$ max degree)
7. NP-hard to get <5/3-approx in offline problem, even if $\Delta \leq 5$
8. PTAS for geometric instances in fixed-dimension, L_p metric
   Time $O(n\log n + 2^{poly(1/\varepsilon)})$

Stars: Equal-Length Spokes:



Natural greedy strategy:

> A robot arriving to v_0 chooses the (unclaimed) leaf having the most asleep robots



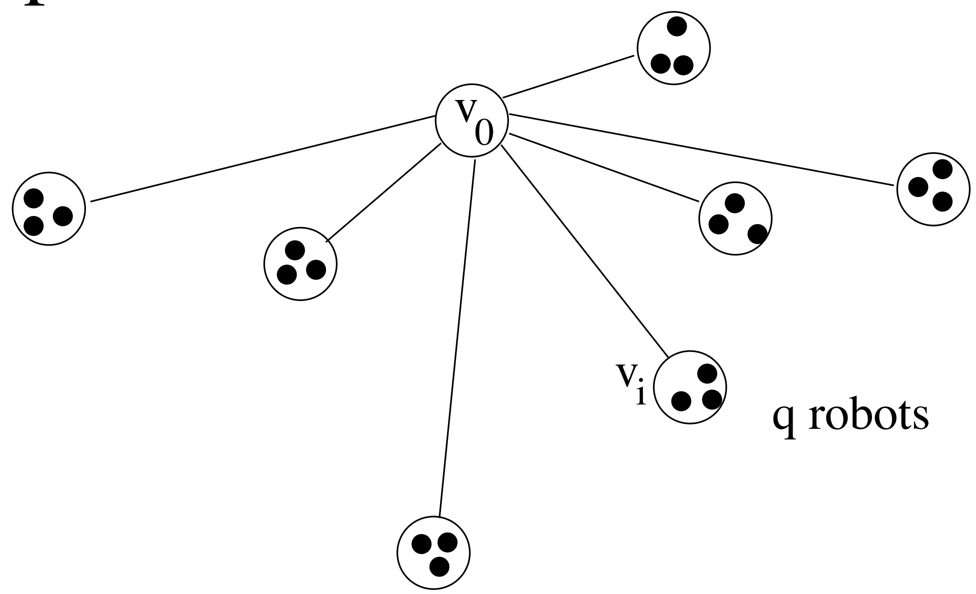
Claim: Greedy is optimal

Proof: two exchange arguments:

- $\exists$ optimal schedule in which no robot is idle if $\exists$ work to be done
- if a robot chooses a branch B with fewer robots than unclaimed branch B', swap

Stars: Unequal Length Spokes:

Assume q robots asleep at each of n leaves

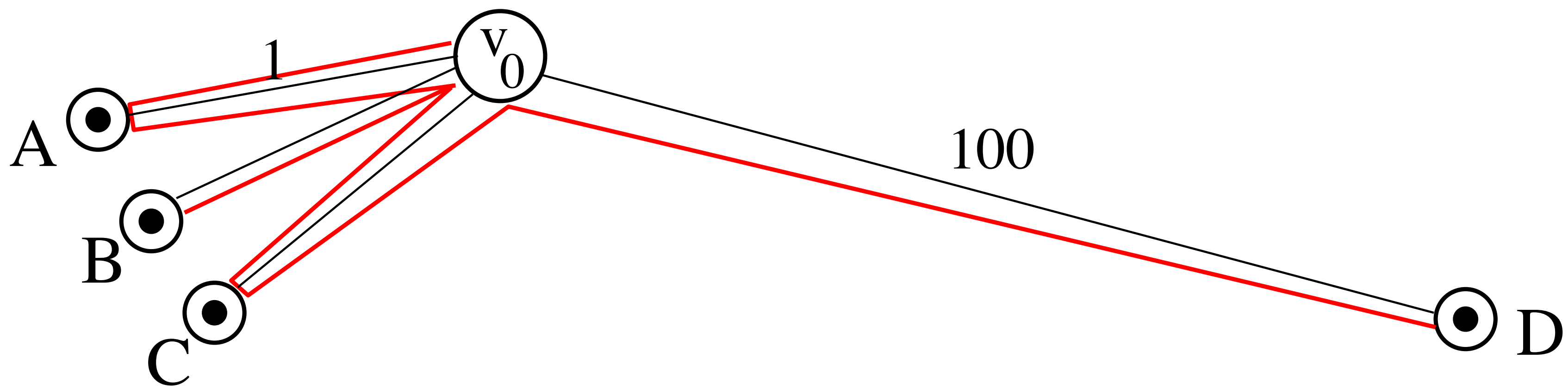


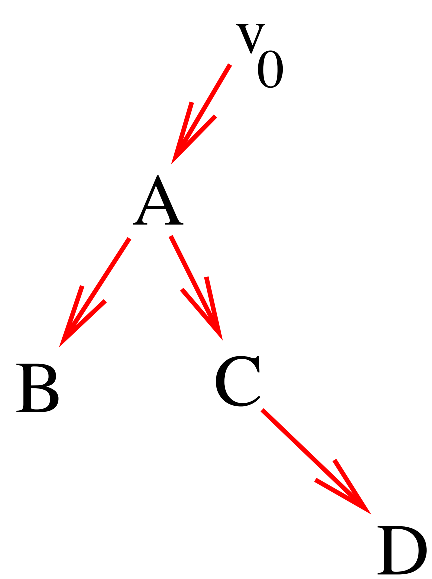
Natural greedy strategy:

A robot arriving to v_0 chooses the *shortest* (unclaimed) spoke having asleep robots

Question: How good is Greedy in this case?

Example: Greedy Strategy:

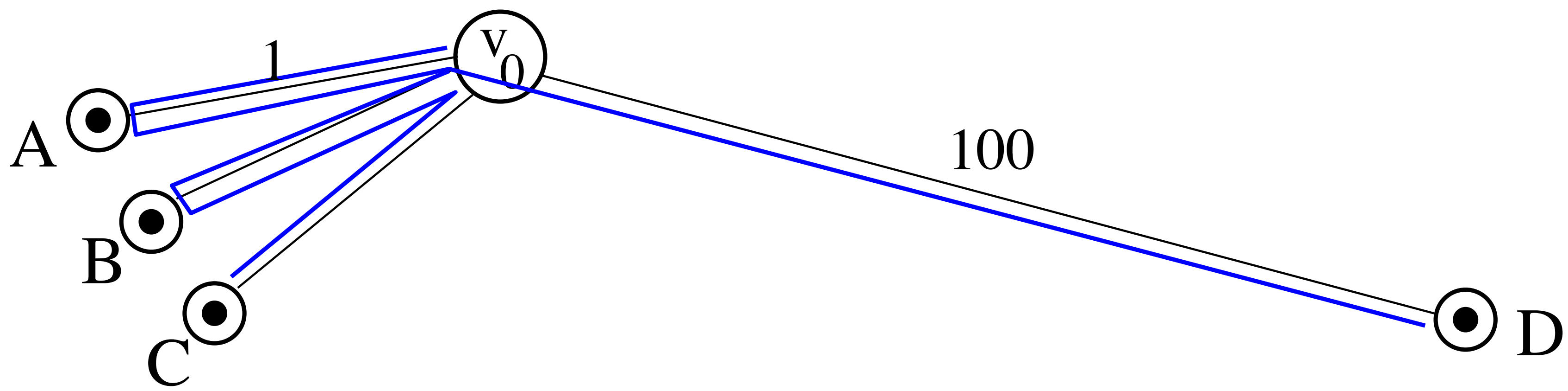


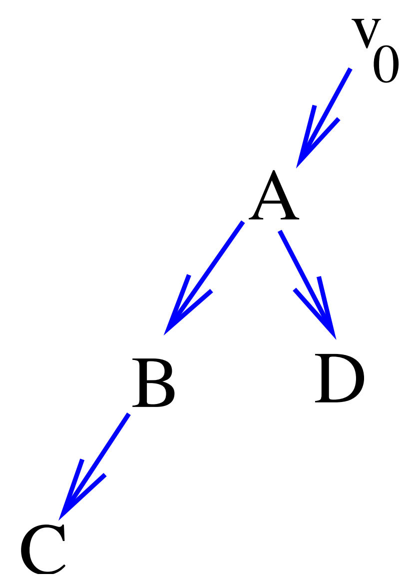


GREEDY wake−up tree

Makespan = 104

Example: Optimal Strategy:





OPT wake–up tree

Makespan = 102

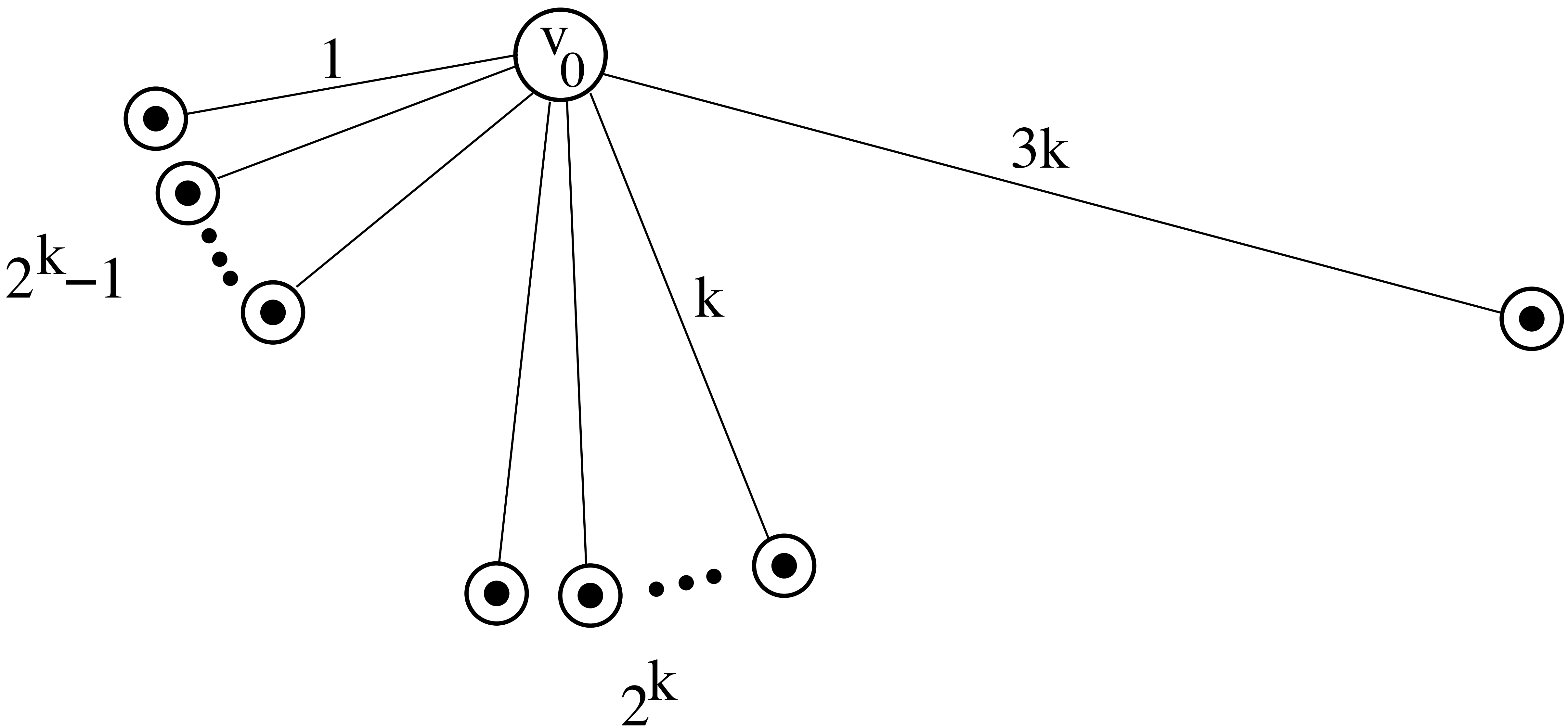
Thus, Greedy can be *suboptimal* (104 vs. 102)

Analysis of Greedy on Stars:

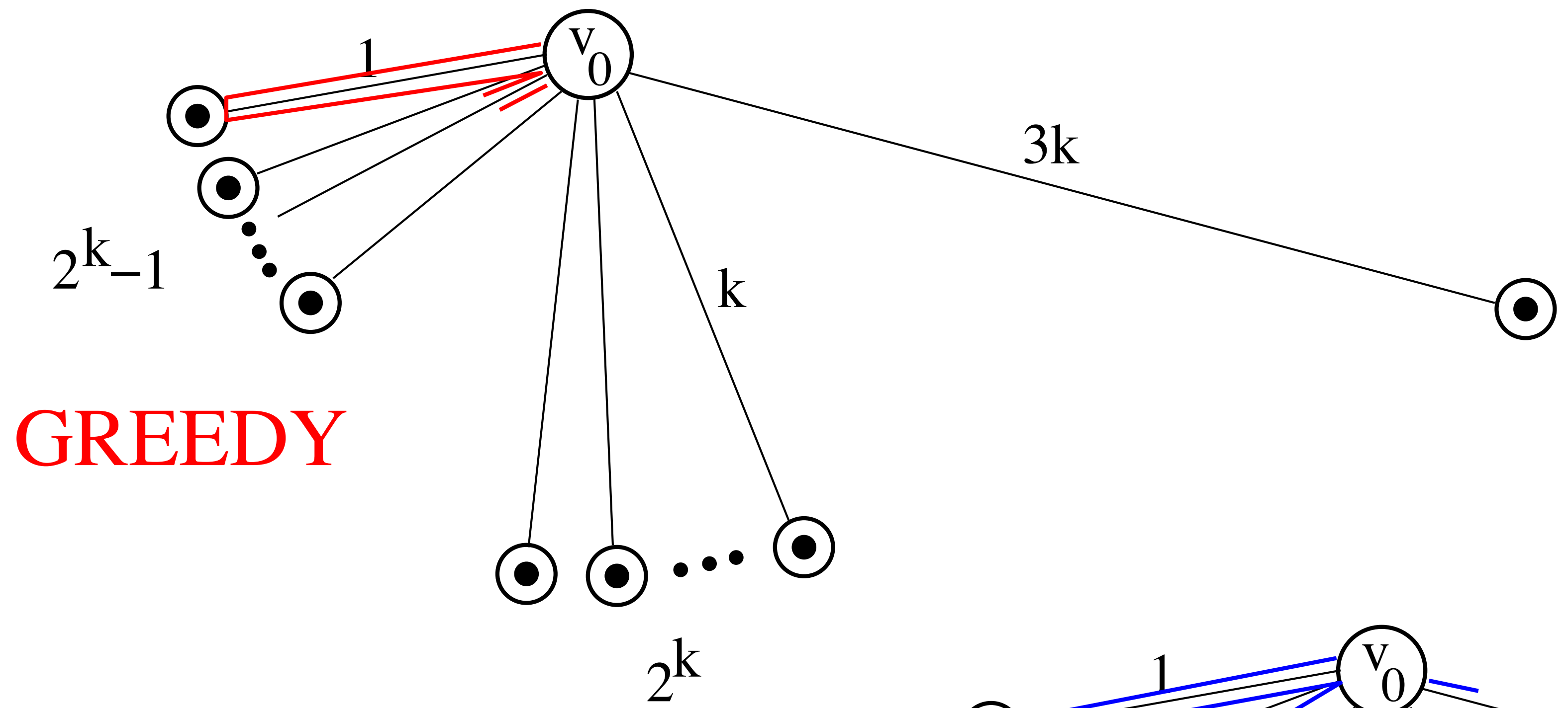
Amazingly, greedy is a 7/3-approx, and this bound is *tight*

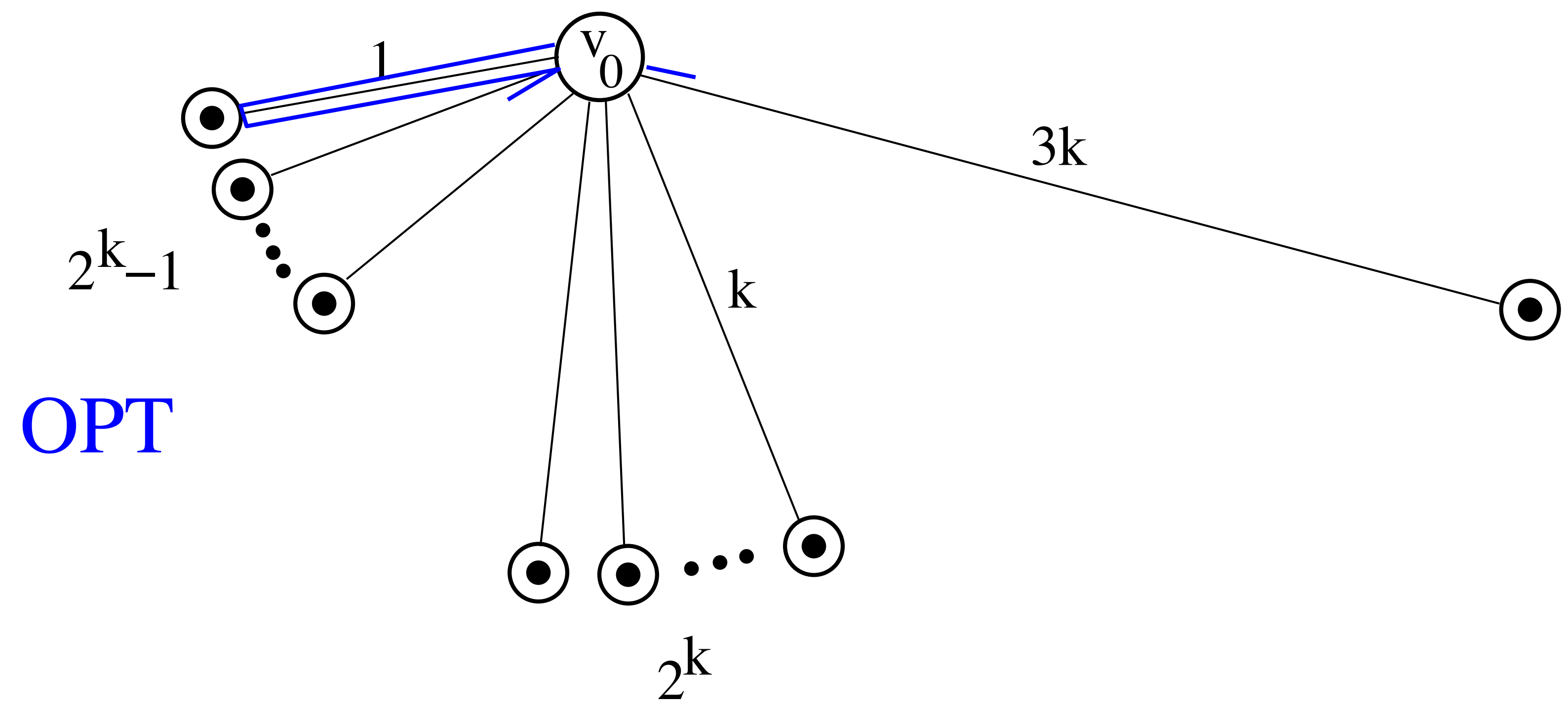
Claim: Greedy is at best a 7/3-approx

Example:

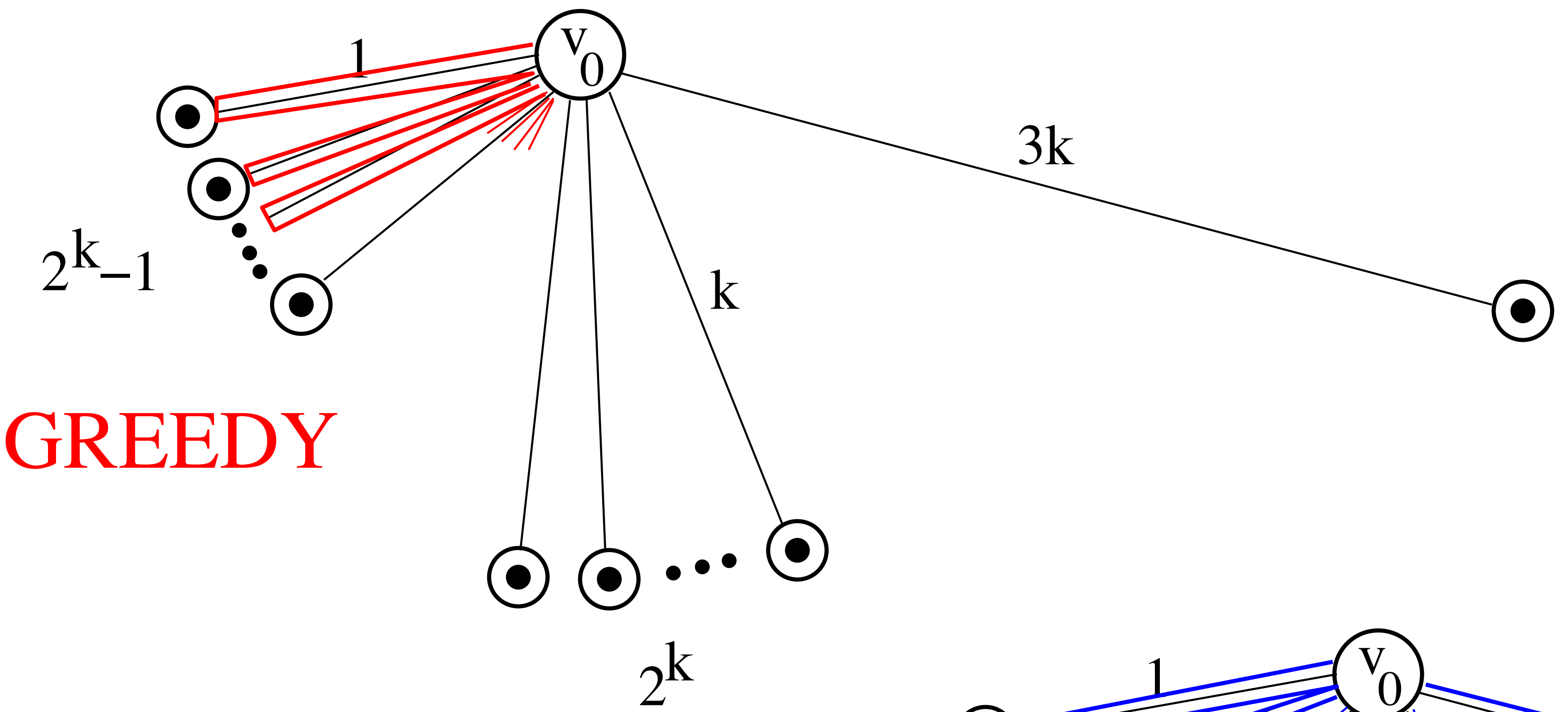


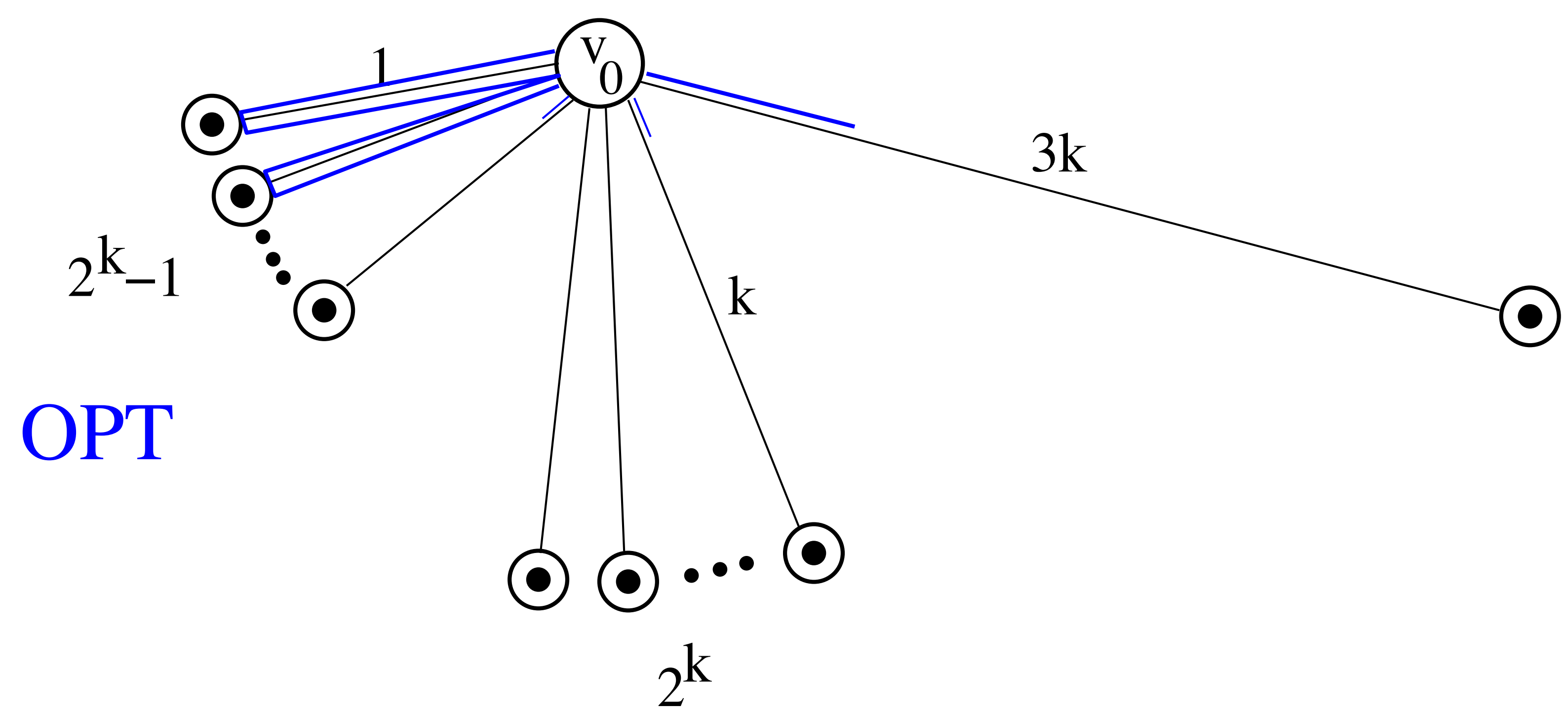
Time = 2:



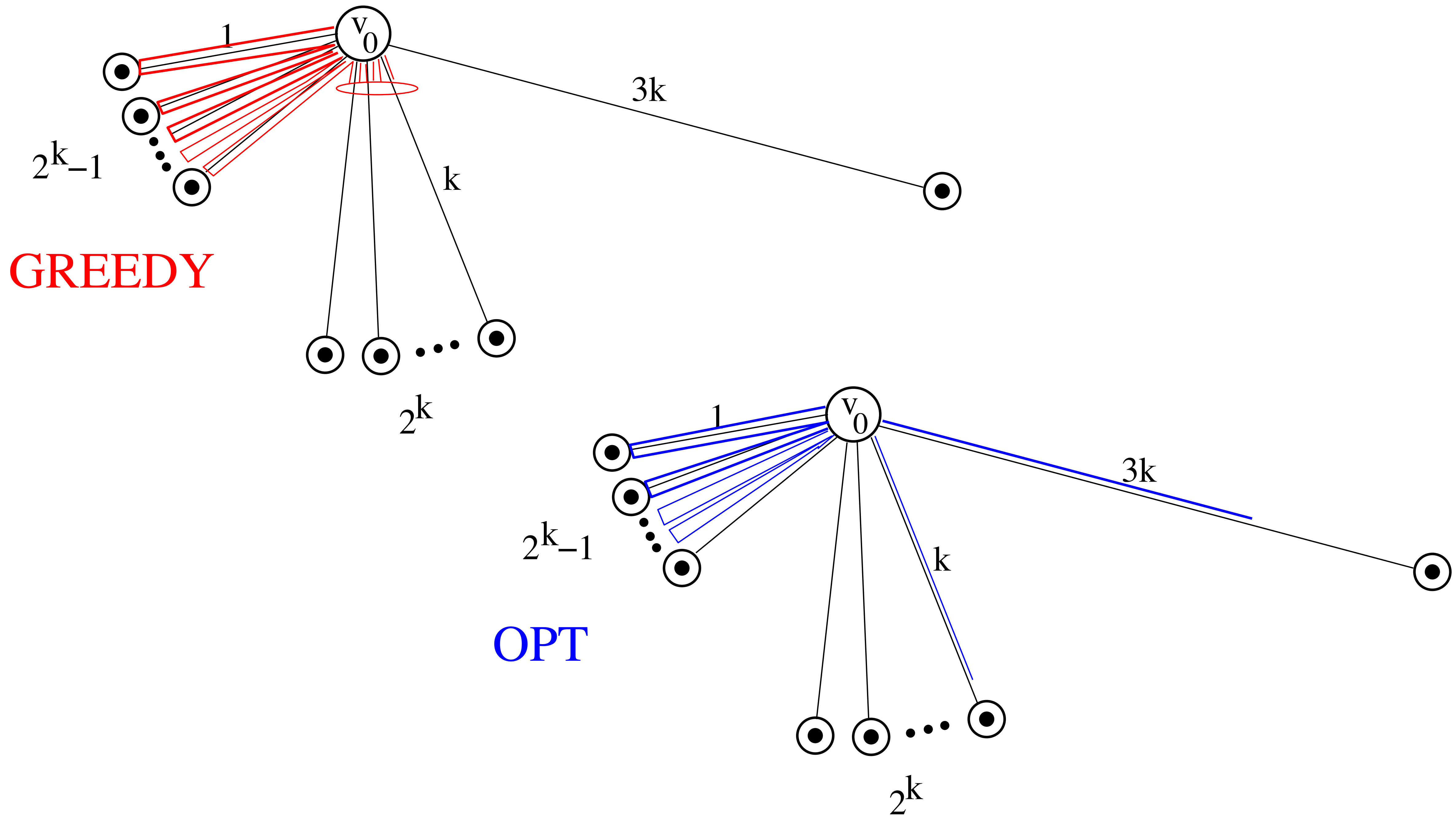


Time = 4:

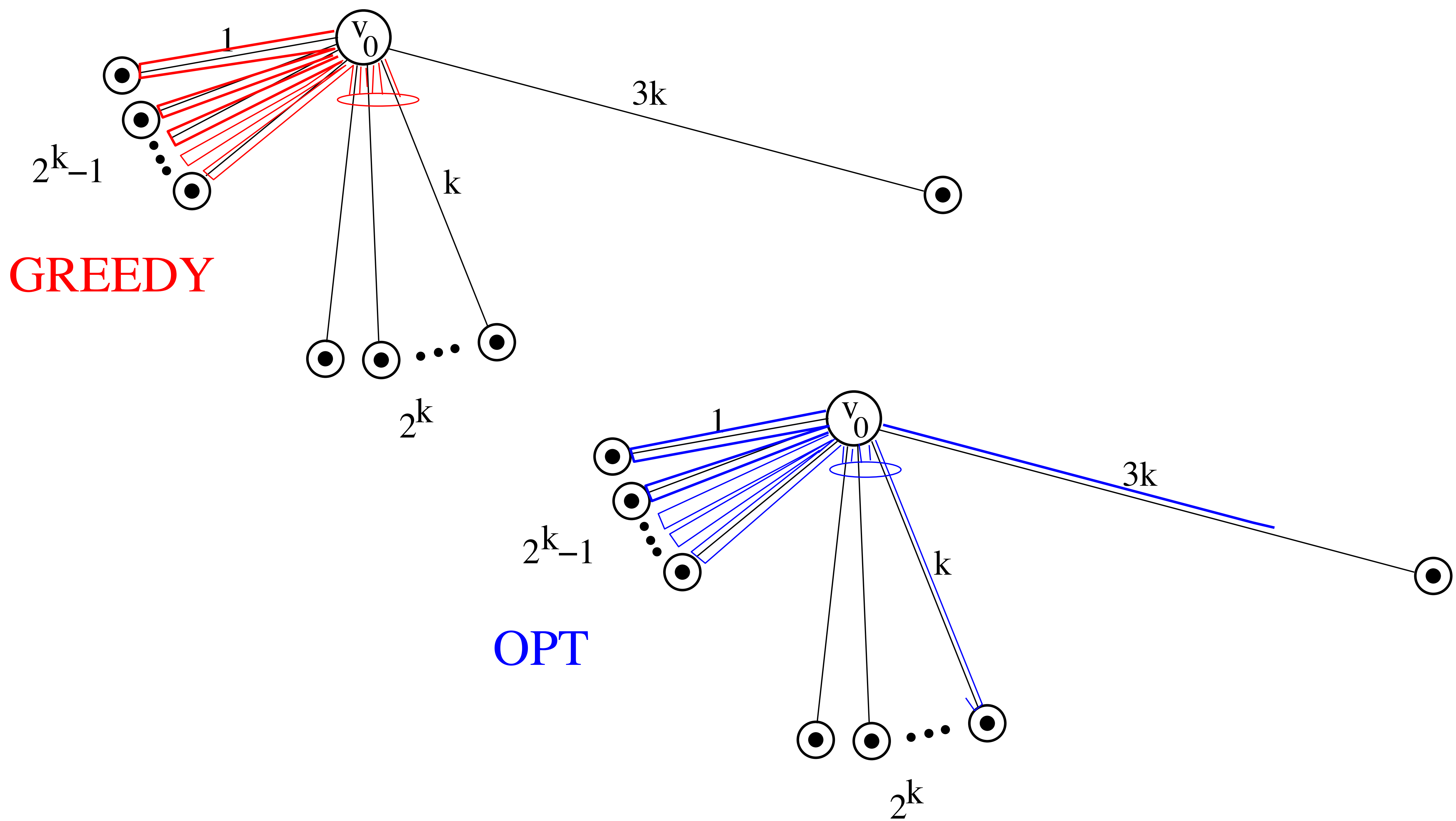




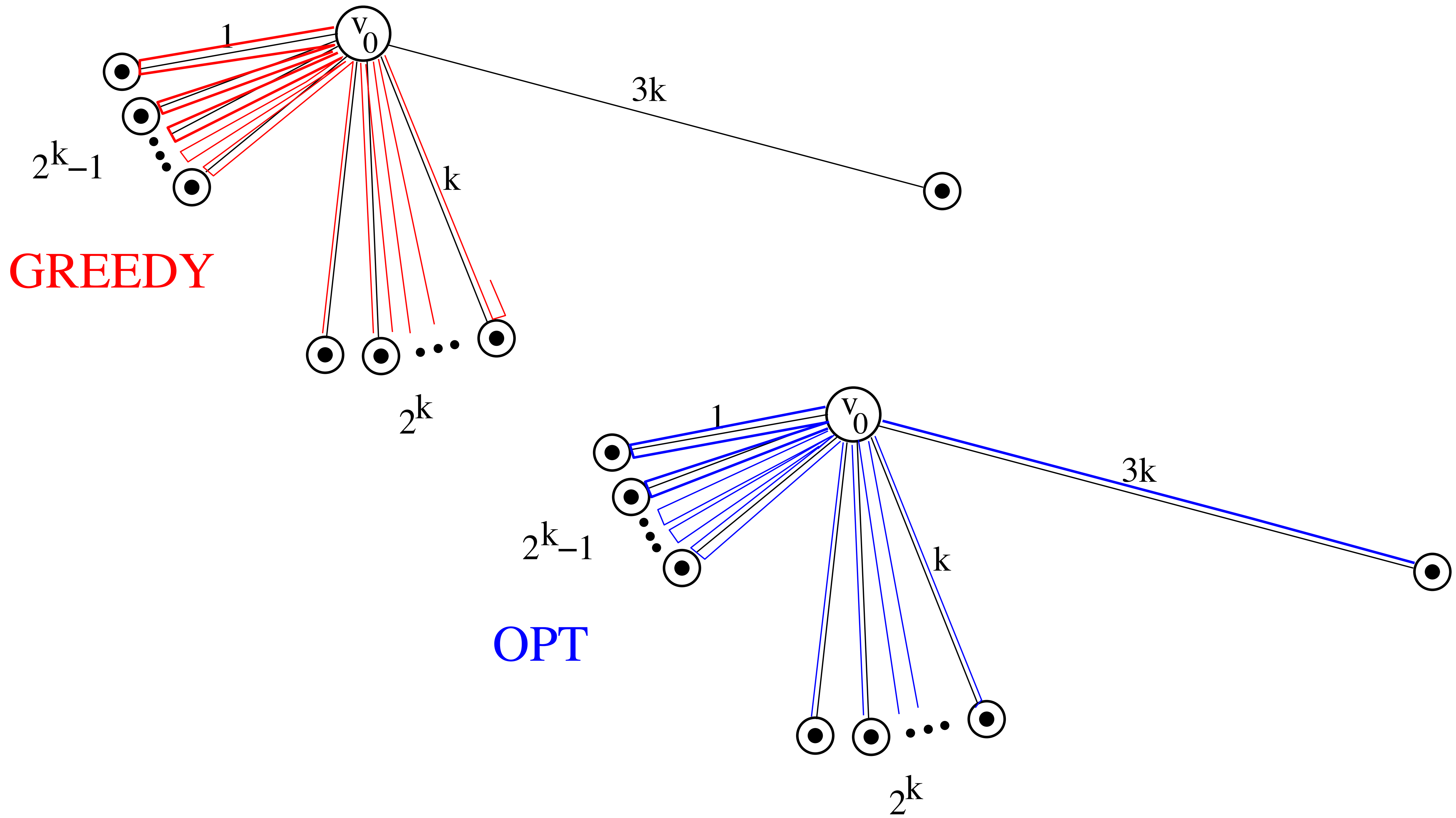
Time = 2k:



Time = 2k+4:



Time = 3k+2:



Greedy still has another $\approx 4k$ to go!

Analysis of Greedy on Stars:

Our analysis relies on the following fact, of independent interest:

Theorem: Greedy minimizes the *average completion time*

(completion time of robot i is the time when the robot is awake and no longer busy (moving))

Proof Idea: Exchange arguments

Analysis of Greedy on Stars:

Theorem: Greedy gives 7/3-approx in stars, and this is tight

(assume q asleep robots at each leaf initially)

NP-hardness for Stars:

Theorem: FTP is strongly NP-hard, even for weighted stars with $q = 1$ robot at each leaf

Proof: Reduction from NUMERICAL 3-DIM MATCHING (N3DM)
Input: Sets W, X, Y, each with n elements of integral sizes a_i, b_j, c_k, and a number d
Question: Can $W \cup X \cup Y$ be partitioned into n disjoint sets $S_1, \ldots, S_n$, with each S_h a triple (one element of W, of X, of Y) of size d?

WLOG: $n = 2^K$ $\qquad a_i, b_j, c_k \leq d$

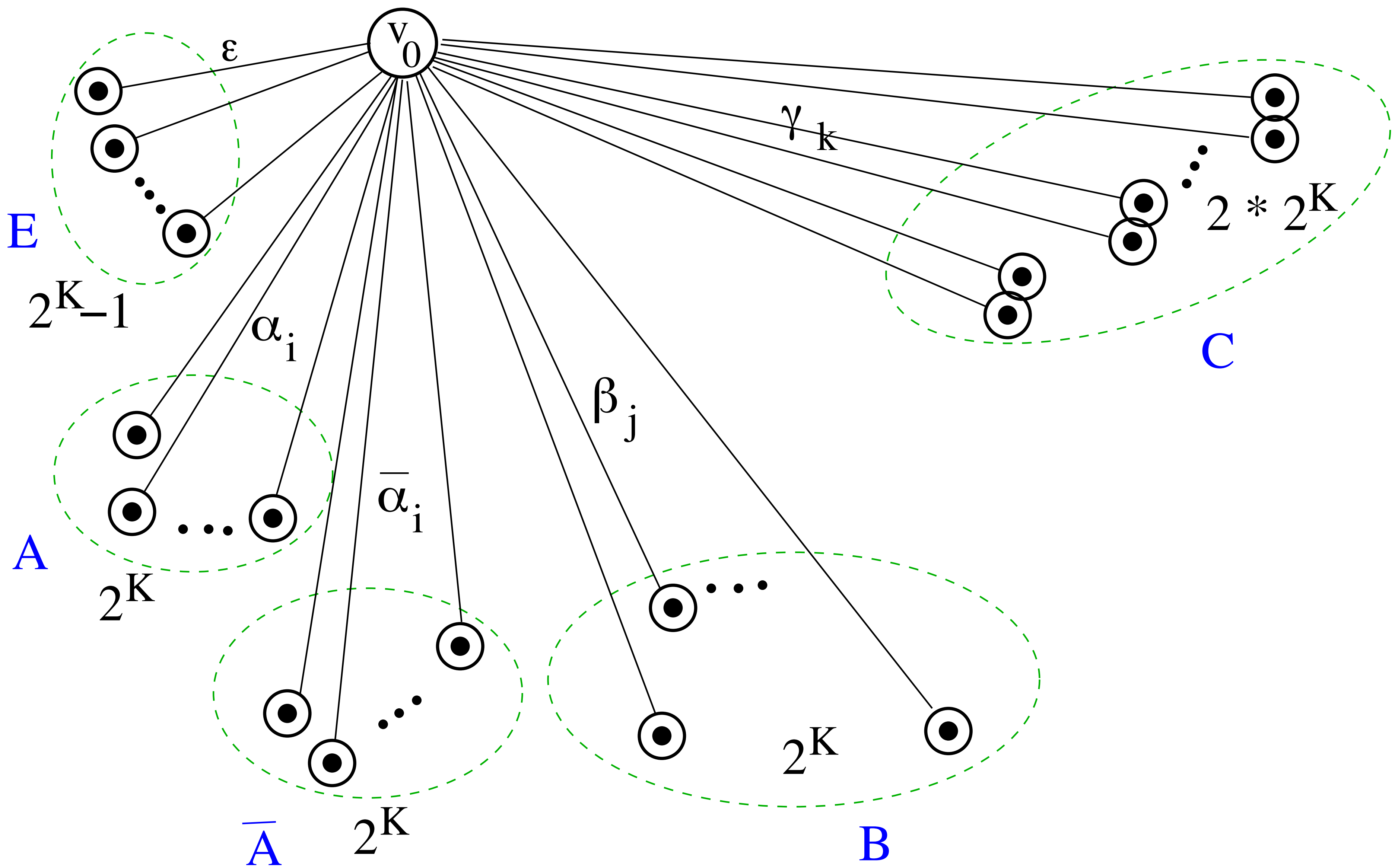
Let ε be sufficiently small $\qquad (\varepsilon < 1/(2K))$

Let L be sufficiently large $\qquad (L := 15d)$

Hardness Construction:

$\alpha_i := a_i/2 - \varepsilon K + d \qquad \overline{\alpha_i} := L - a_i - 2d$

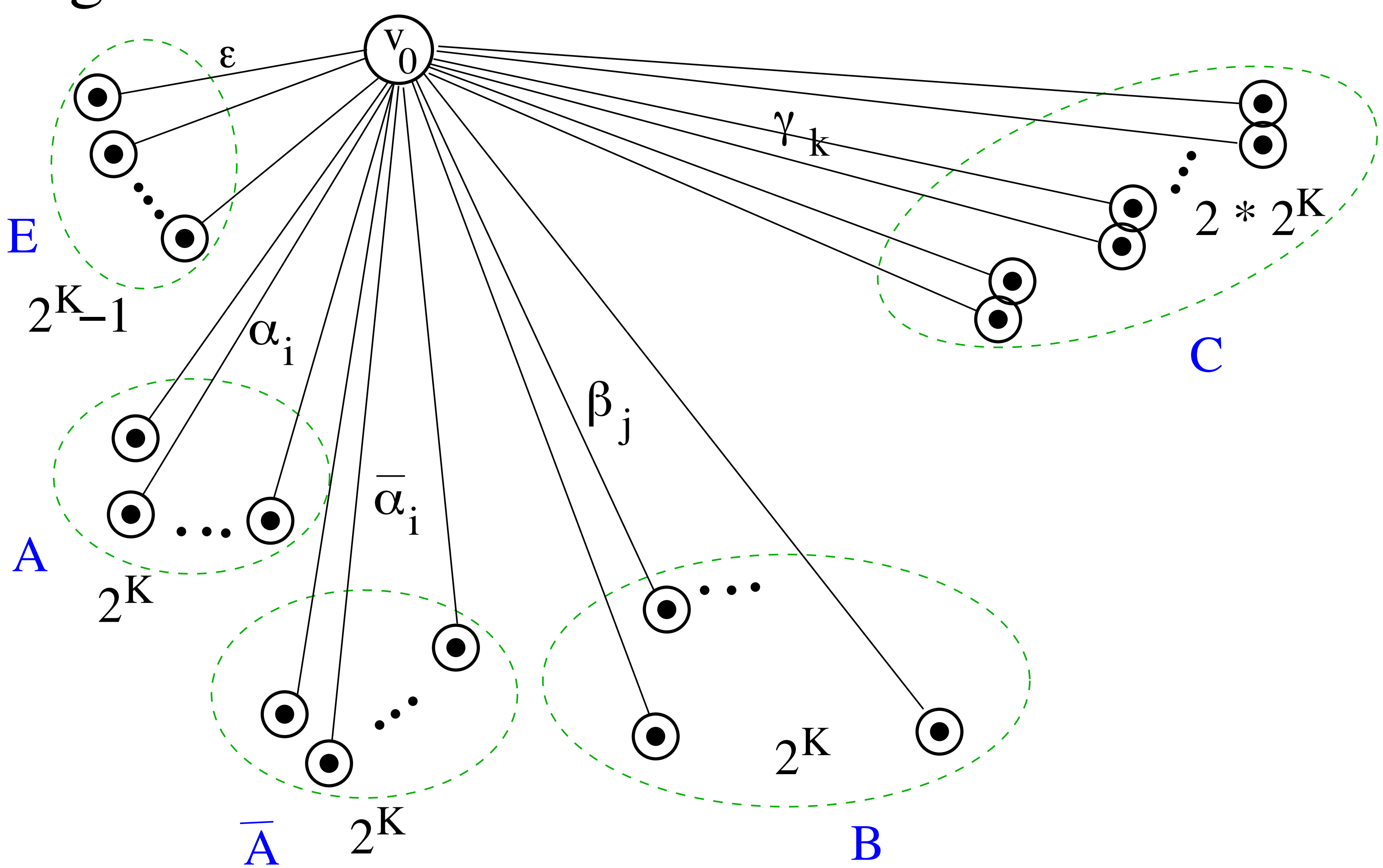
$\beta_j := b_j/2 + 2d \qquad \gamma_k := L - 7d + c_k$



Claim: $\exists$ schedule of makespan to awaken all robots within time L iff $\exists$ solution to N3DM

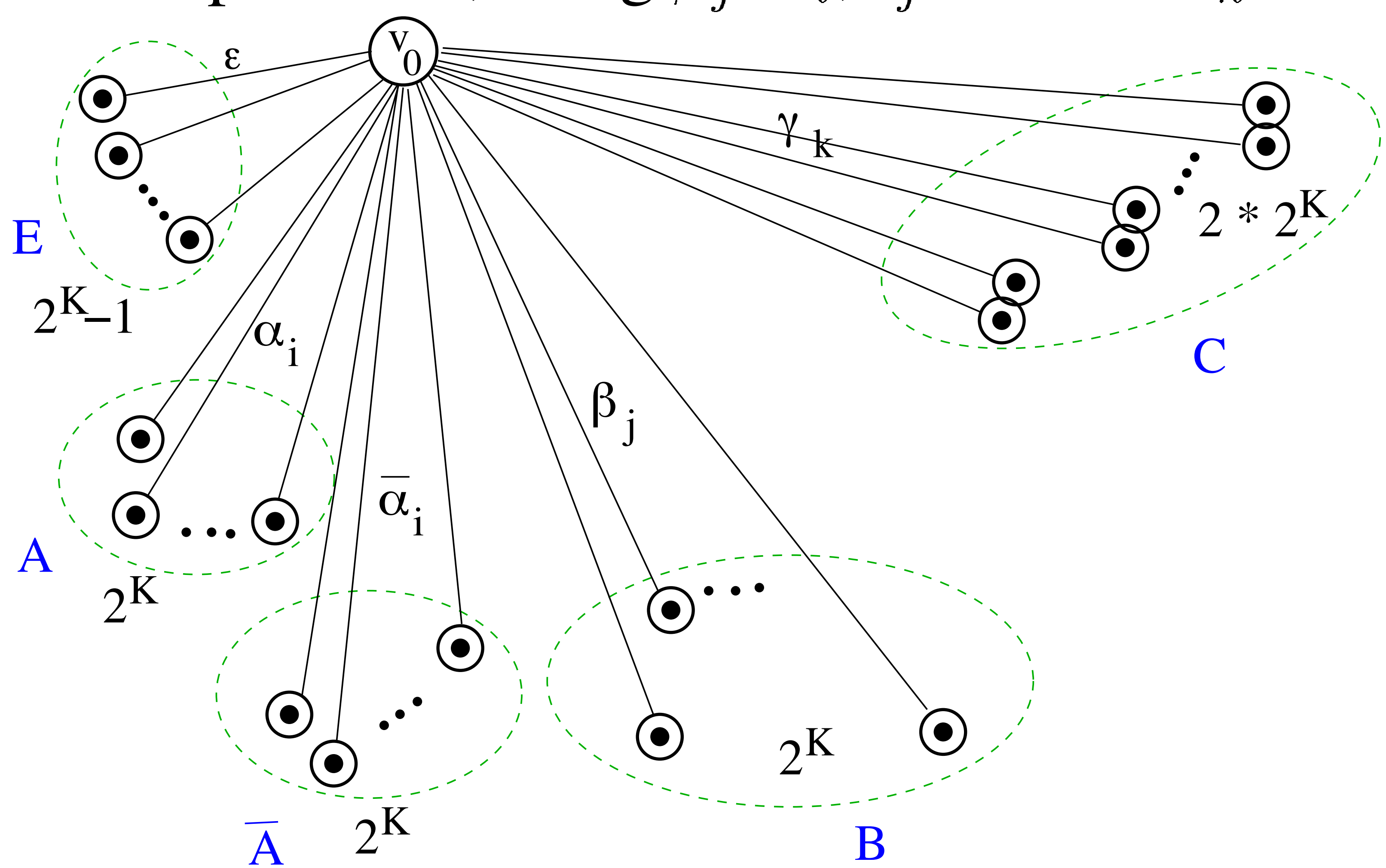
"IF": (**"ONLY IF"** is more involved)

- "greedy cascade" brings all $n = 2^K$ robots to v_0, time $2\epsilon K$
- these robots go to A-leaves



$$\alpha_i := a_i/2 - \varepsilon K + d, \qquad \overline{\alpha_i} := L - a_i - 2d$$

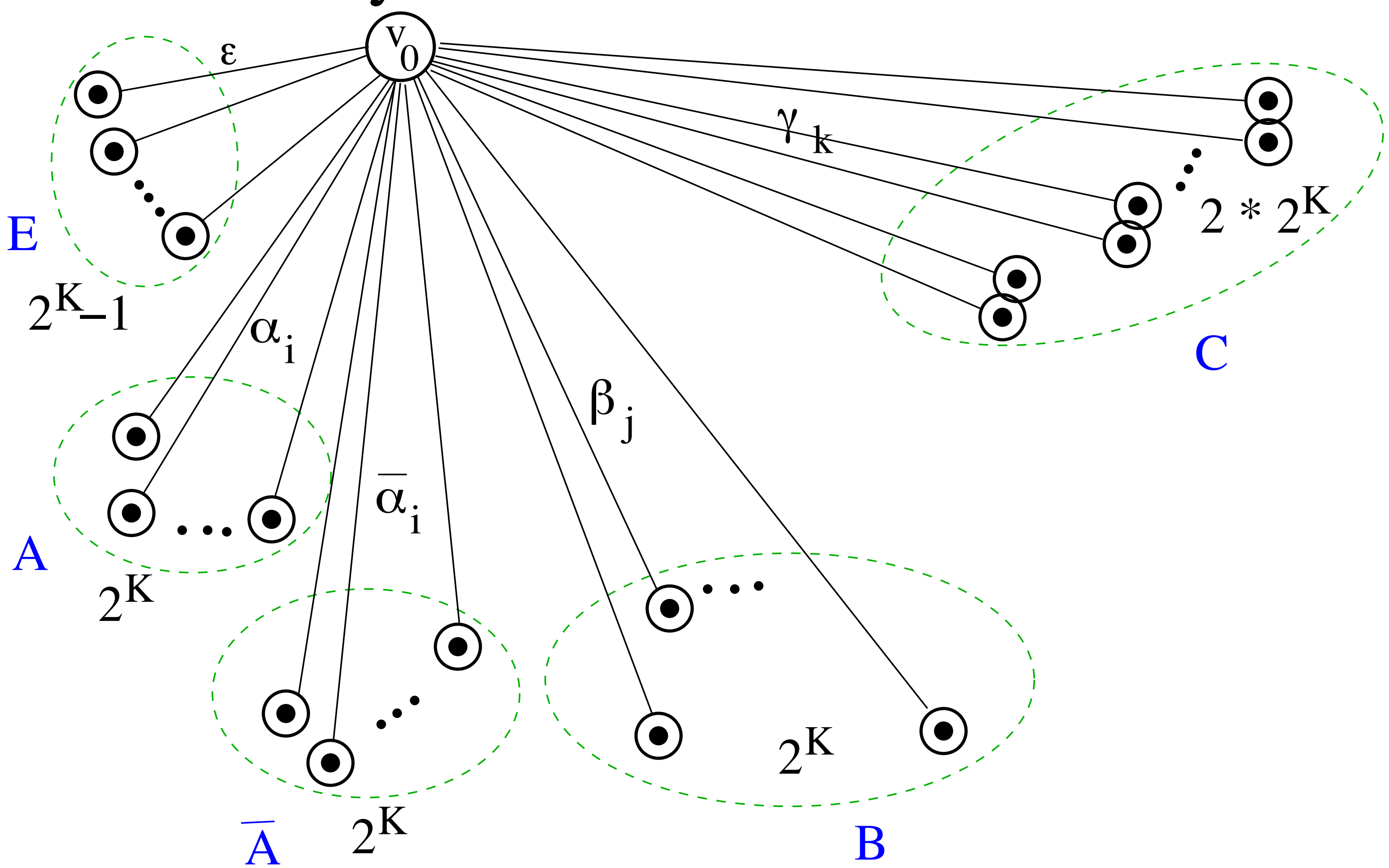
- 2 each return to v_0 at times $a_i + 2d$
- of each pair to $\overline{A}$, along α_i
- the other of each pair to B, along β_j: a_i, b_j in same S_h



$$\alpha_i := a_i/2 - \varepsilon K + d, \qquad \overline{\alpha_i} := L - a_i - 2d$$

$$\beta_j := b_j/2 + 2d, \qquad \gamma_k := L - 7d + c_k$$

- 2 robots get back to v_0 by time $a_i + b_j + 6d$
- they go to C, down pair of edges of length γ_k: $a_i + b_j + c_k = d$
- all robots at C awake by time L



$$\alpha_i := a_i/2 - \varepsilon K + d, \qquad \overline{\alpha_i} := L - a_i - 2d$$
$$\beta_j := b_j/2 + 2d, \qquad \gamma_k := L - 7d + c_k$$

PTAS for Stars with q Robots per Leaf:

Theorem: There is a PTAS for weighted stars with q robots at each leaf

Proof idea:

- Let $T \leq t^*$ be a good lower bound on makespan
  (use 3/7 times greedy solution)
- Partition edges: "short" (length $\leq \epsilon T$) and "long" (length $> \epsilon T$)
- Round up lengths of long edges to multiples of $\epsilon^2 T$
- Suffices to consider schedules in which each long edge is entered by a robot at a time that is a multiple of $\epsilon^2 T$
- Only $O(1/\epsilon^2)$ different lengths/start-times of long edges
- Enumerate (in $O(n^{O(1/\epsilon^4)})$) all possibilities of how many long edges of a given length are started at a given time
- In this way, we have "guesssed" the correct positions of all long edges
- Fill in short edges with a variant of greedy

General Stars with n_i Robots at Leaf v_i:

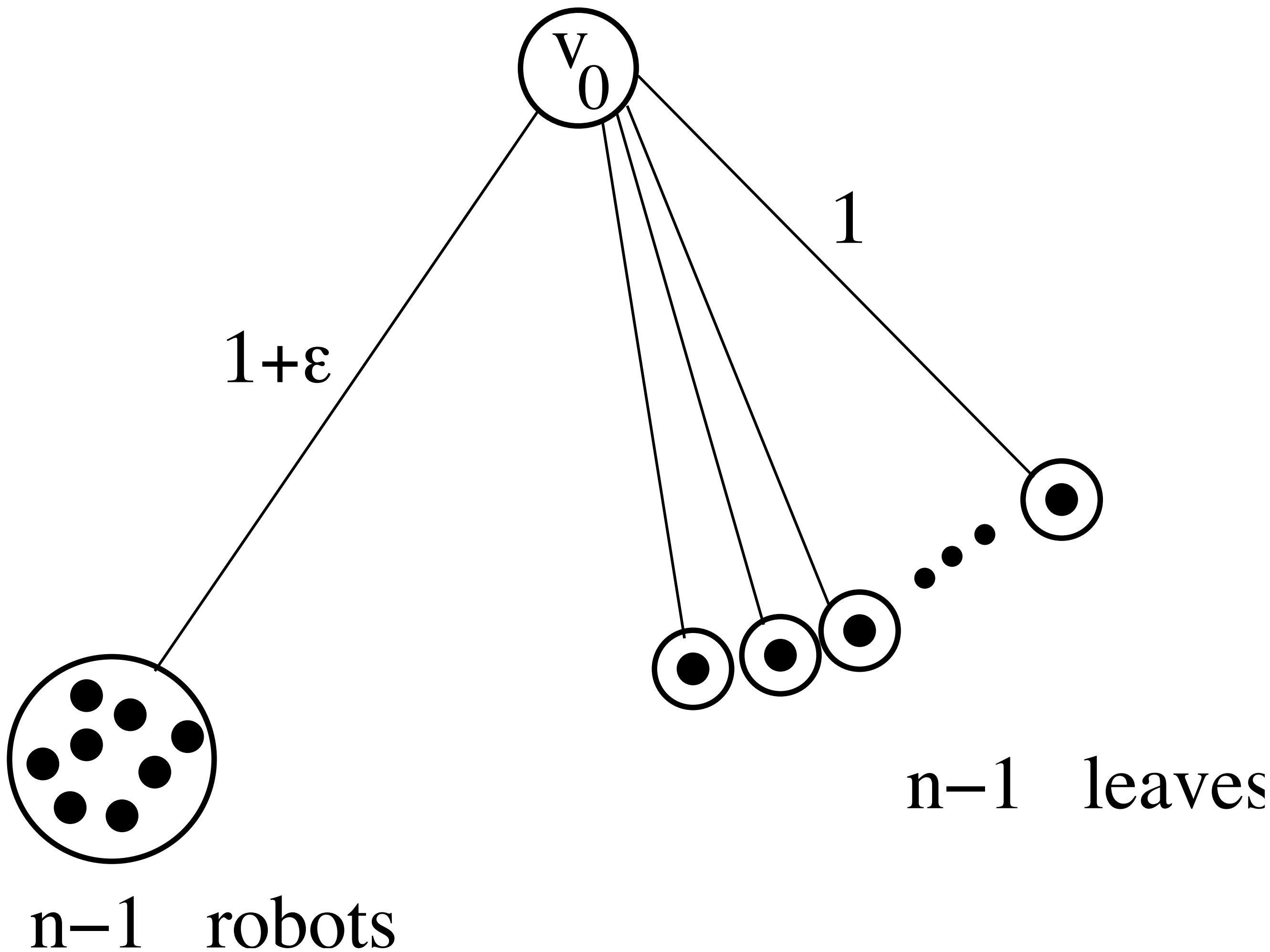
Now: *centroid metric*

(star with various spoke lengths, various n_i)

Consider a natural greedy strategy:

"Shortest Edge First" (SEF): An awake robot at v_0 picks a shortest edge leading to asleep robots, breaking ties according to # robots

Example: Shortest Edge First Can be Bad:



SEF: $\Theta(\log n)$ **OPT:** $O(1)$

Key Dilemma: Choose a short edge leading to few robots or a long edge leading to many?

Devising an Alternative Strategy:

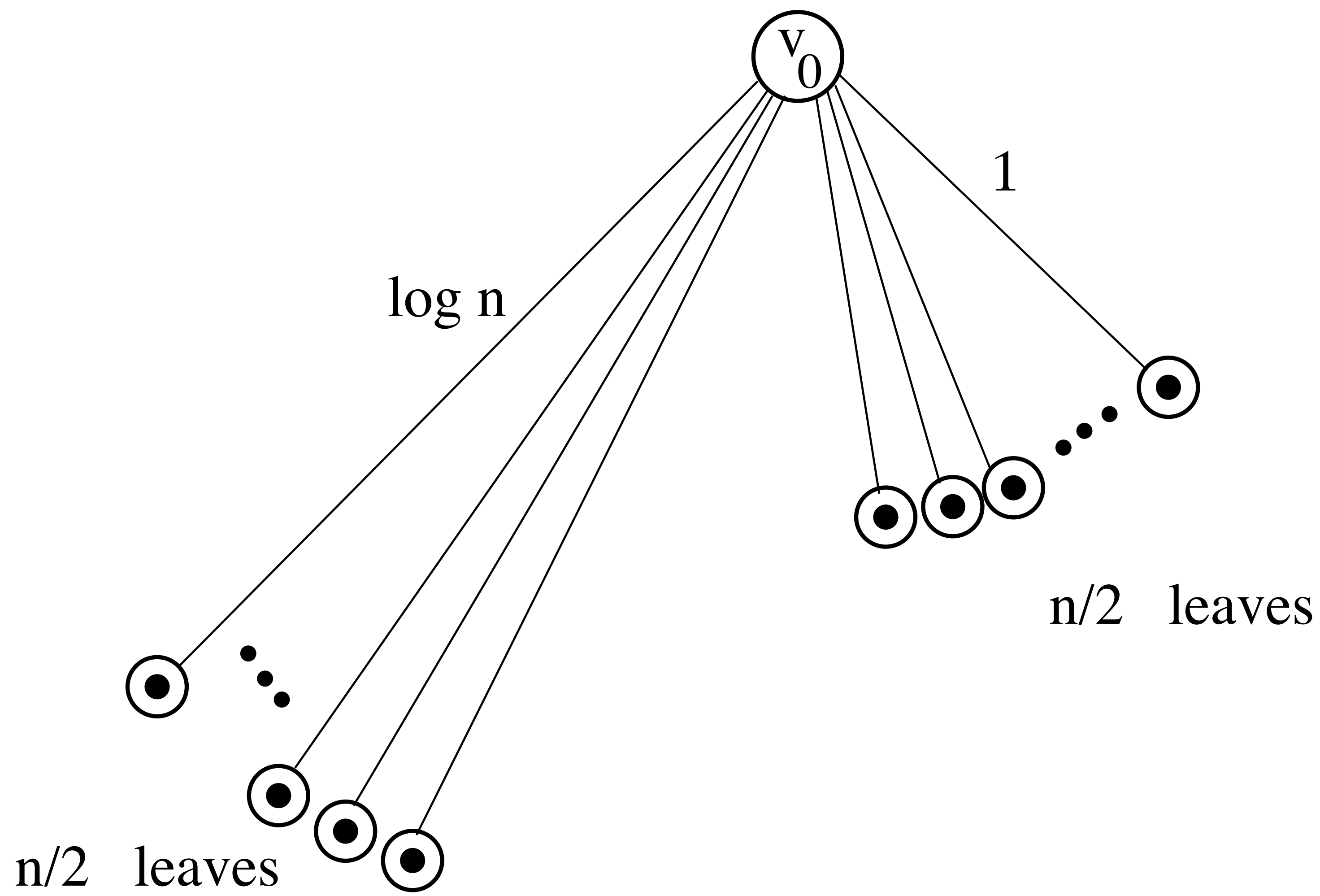
Round edge lengths to powers of 2 (may double approx factor)

Issue: How to pick what length class to visit first?

Idea: Hedge our bets by repeated doubling

"Repeated Doubling" (RD): Edge lengths traversed repeatedly (roughly) double in size; in each length class be greedy in # robots

Example: Repeated Doubling Can be Bad:



RD: $\Theta(\log^2 n)$ **OPT:** $O(\log n)$

Idea: Merge the SEF and RD strategies

Tag-Team Algorithm:

Awaken one edge in each length class $1, 2, 4, 8, \ldots$, **but** before going to the next length class, awaken the shortest available edge.

This *combination* of two $\Theta(\log n)$-approx methods yields an $O(1)$-approx!

Theorem: The Tag-Team Algorithm gives a 14-approx

General Weighted Graphs:

General graph $G = (V, E)$ with non-negative edge weights

$r(v) =$# asleep robots at v $\qquad$ $\delta(v) =$degree of v

Lemma: Suppose $r(v_0) \geq \delta(v_0)$ for the source node v_0, and $r(v_i) \geq \delta(v_i) - 1$ at any other node in G. Then the FTP can be solved by breadth-first search.

General Weighted Graphs:

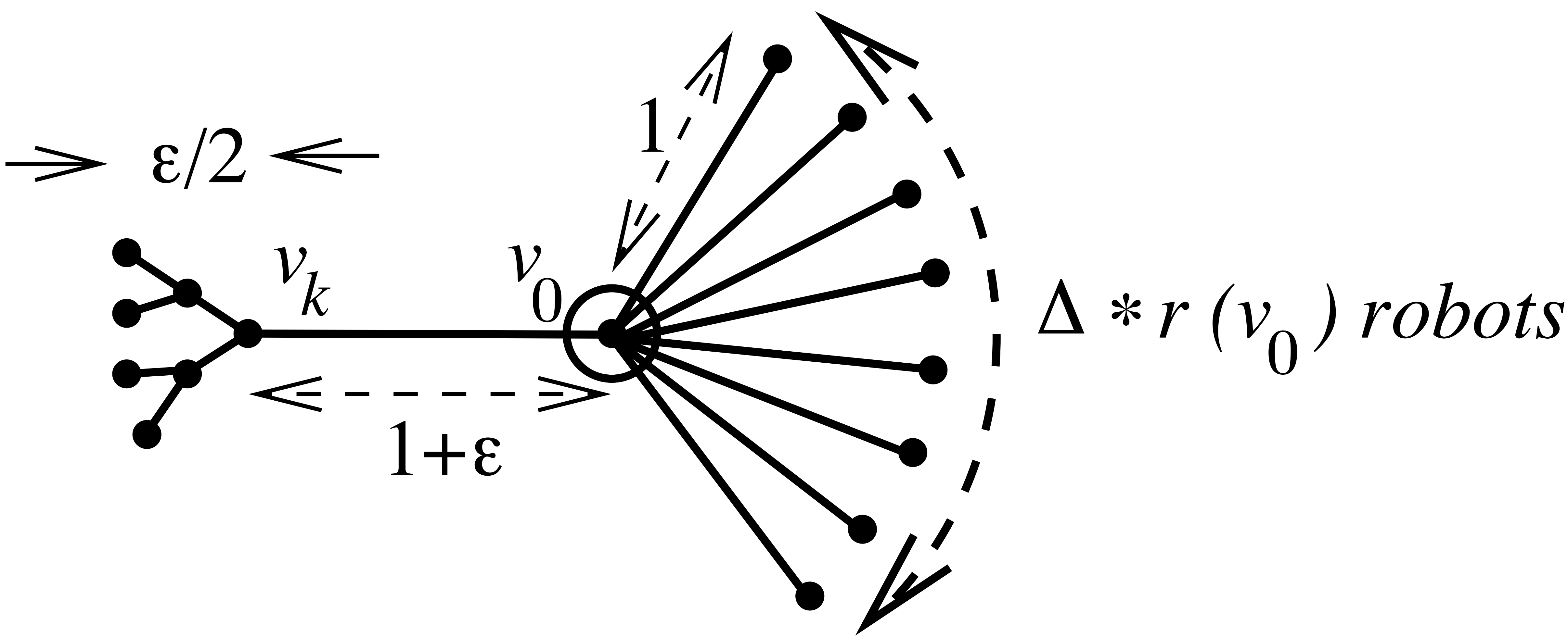
$$\Delta_G := \max\{\frac{\delta(v_0)}{r(v_0)}, \frac{\delta(v_i)-1}{r(v_i)}, i = 1, \ldots, n-1\}$$

Theorem: There is a linear-time on-line algorithm for the FTP in G that guarantees a competitive ratio of $O(\log \Delta_G)$.

Proof Sketch: Simulate BFS: At v_i use a greedy strategy to wake up all robots adjacent to v_i, with binary tree of depth $\lceil \log \frac{\delta(v_0)}{r(v_0)} \rceil$ for the root, and $\lceil \log \frac{\delta(v_i)-1}{r(v_i)} \rceil$ for v_i

Greedy implies that any edge e in the resulting wake-up tree can only be placed below edges f that satisfy $w_f \leq w_e$.

Bad Example for Local Greedy Strategy:



Local greedy takes $\Theta(\log n)$; OPT takes $3(1+\epsilon)$

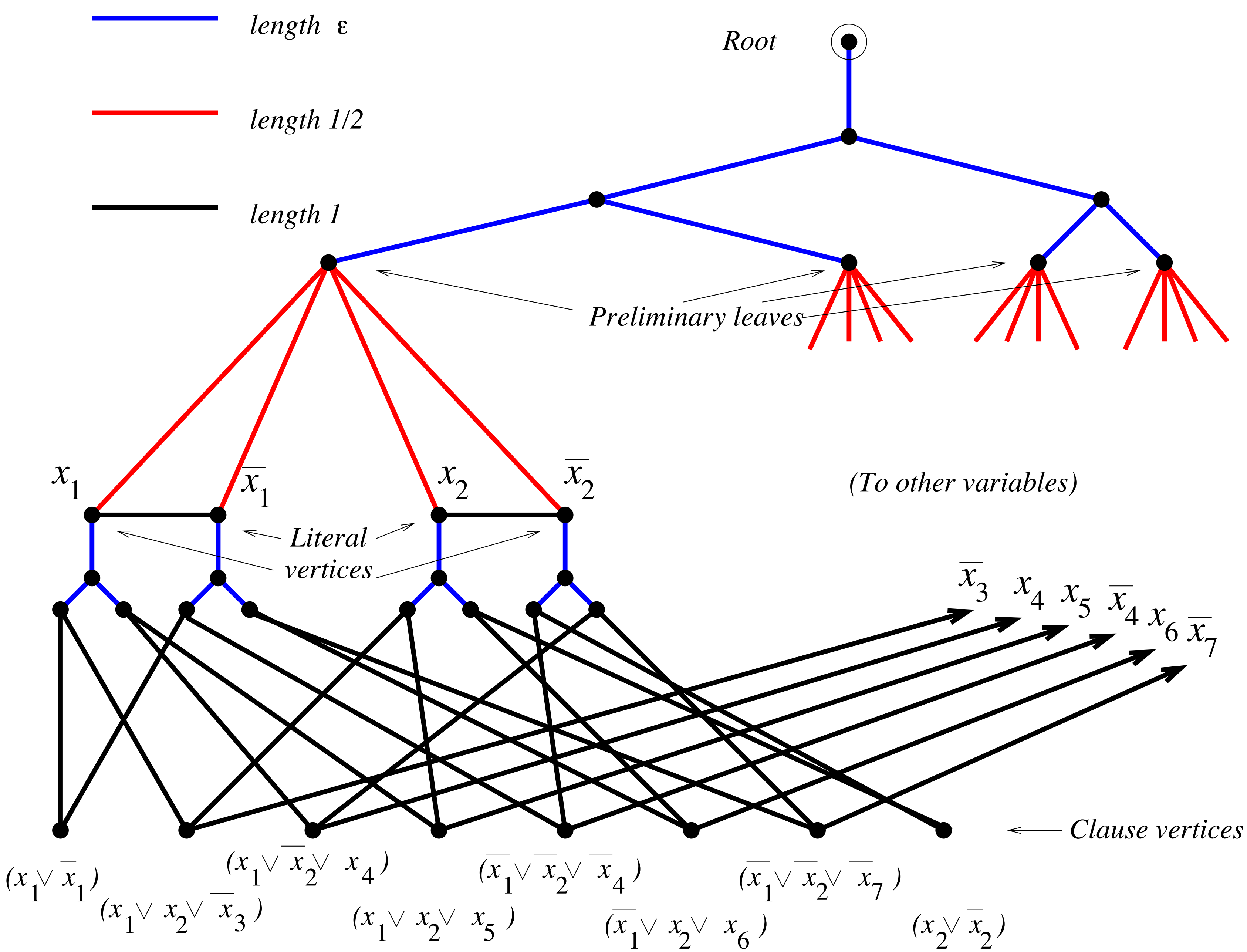
Hardness of Approx:

- FTP is NP-hard even if *one* high-degree vertex (v_0)

Question: If degrees are all small, can we do much better?

Theorem: Even if all nodes have degree ≤ 5 and at most one robot per node, it is NP-hard to get better than 5/3-approx.

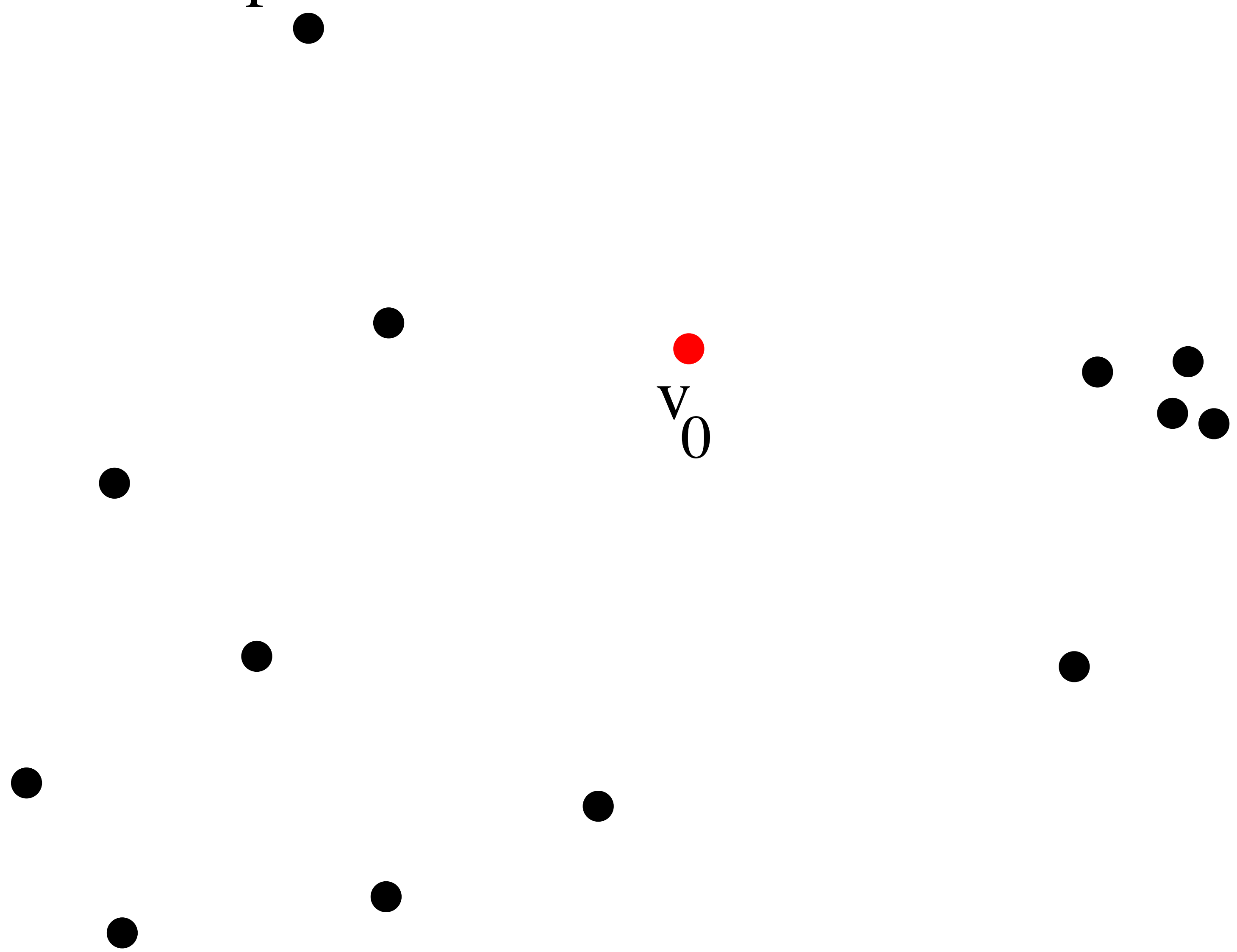
Proof Sketch: From 3SAT:



A solution with makespan $3/2+O(\epsilon \log n)$ if $\exists$ satisfying truth assignment; makespan $\geq 5/2$ if none (pick $\epsilon = o(\log n)$)

Geometric Instances:

Robots at points in the plane:



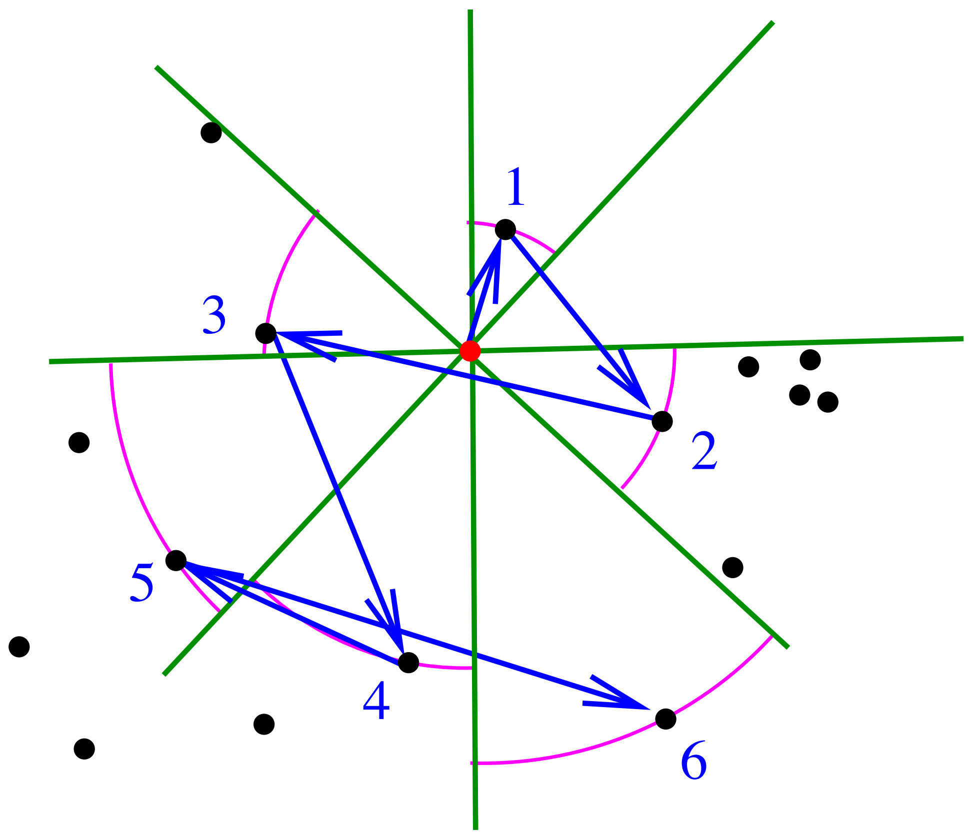
Question: Can we exploit geometry to get good approx?

OPEN: Is the problem NP-hard?

Geometric Instances: $O(1)$-approx:

Theorem: $\exists$ an $O(1)$-approx, time $O(n \log n)$, for the geometric FTP in any fixed dimension d. The algorithm gives wake-up schedule with makespan $O(diam(R))$.

Strategy: When robot at p awakens, it awakens nearest asleep robot in each of K sectors, in order of increasing distance from p



- $G_K = (R, E_K)$ is a Θ-graph (which is a t-spanner)

- Distances in G_K approx Euclidean

- Let v_ℓ be the last robot to be awakened

- If robot at point v is awakened at time t, then all neighbors of v in G_K are awakened by time $t + \xi$, where $\xi =$ length of path $v, u_1, u_2, \ldots, u_j$.

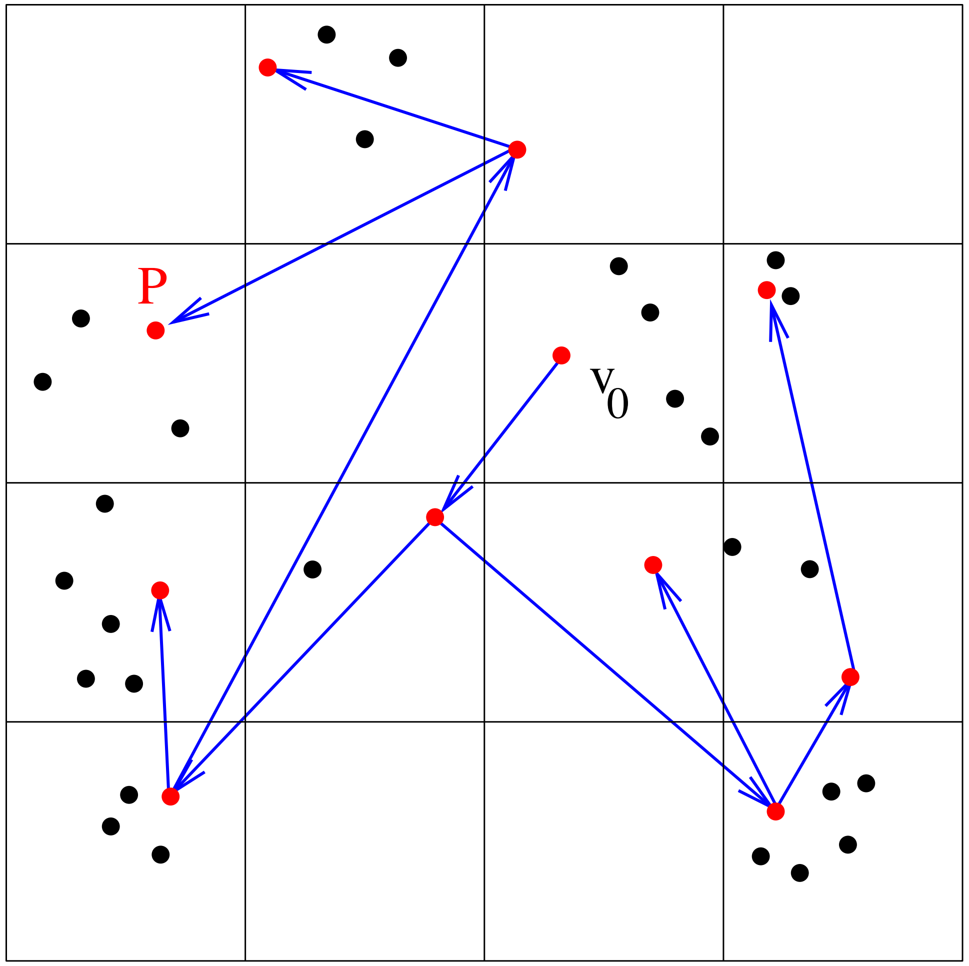
- $\xi \leq (2j - 1) \cdot d(v, u_j) \leq (2K - 1) \cdot d(v, u_j)$

- The path from v_0 to v_ℓ in the wake-up tree has length at most $(2K - 1) \cdot d_{G_K}(v_0, v_\ell) \leq O(1) \cdot d(v_0, v_\ell)$

PTAS for Geometric Instances:

Rescale so that all robots lie in unit square

Look at m-by-m grid of pixels, $m = O(1/\epsilon)$



Consider an enumeration over a special class of wake-up trees on a set P of representative points, one per occuppied pixel

A wake-up tree is *pseudo-balanced* if each root-to-leaf path has $O(\log^2 n)$ nodes

ALGORITHM:

0. Pick a representative point in each occuppied pixel $\rightarrow$ set P

1. Among all pseudo-balanced wake-up trees for P, pick one $(\mathcal{T}_b^*(P))$ of min makespan, $t_b^*(P)$

    (outdegree at most $\min\{m^2 - 1, \ell + 1\}$ if ℓ robots in pixel)

Only $2^{O(m^2 \log m)}$ trees.

2. Convert $\mathcal{T}_b^*(P)$ into a wake-up tree for *all* robots by replacing each $p \in P$ with an $O(1)$-approx wake-up tree for robots in p's pixel

(time $O(n \log n)$)

Total time: $O(2^{O(m^2 \log m)} + n \log n)$

Correctness:

Lemma 1. There is a choice of representative points P such that the makespan of an optimal wake-up tree of P is at most $t^*(R)$.

Proof: Just pick the representative point to be the location of the first robot that is awakened in an opt solution, $\mathcal{T}^*(R)$, for the set R of all robots.

Lemma 2. If $\exists$ wake-up tree, $\mathcal{T}$, of makespan t, then, for any $\mu > 0$, there exists a pseudo-balanced awakening tree, $\mathcal{T}_b$, of makespan $t_b \leq (1+\mu)t$.

Proof Sketch:

Use a heavy path decomposition of $\mathcal{T}$

Any root-to-leaf path has only $O(\log n)$ light edges

Decompose each heavy path into short subpaths of length $\xi = \mu t / \log n$

(only $O((1+1/\mu)\log n)$ subpaths on any root-to-leaf path of $\mathcal{T}$)

Modify wake-up tree $\mathcal{T}$ to transform each subpath into a wake-up tree of height $O(\log n)$, with a small increase in makespan

Lemma 3. For any two choices, P and P', of the set of representative points, $t_b^*(P) \leq t_b^*(P') + O((\log^2 m)/m)$.

Proof: Pixels have size $O(1/m)$ and there are at most $O(\log^2 m)$ awakenings in each root-to-leaf path of a pseudo-balanced tree; thus, any additional wake-up cost is bounded by $O((\log^2 m)/m)$.

Lemma 4. For any pseudo-balanced wake-up tree of P, there exists a wake-up tree, $\mathcal{T}(R)$, with makespan $t(R) \leq t_b(P) + O((\log^2 m)/m)$.

Putting the Pieces Together:

Theorem: There is a PTAS, with running time $O(2^{O(m^2 \log m)} + n \log n)$, for the geometric FTP in any fixed dimension d.

Proof:

The makespan, t, of the wake-up tree we compute obeys:

$$\begin{aligned} t &\leq t_b^*(P) + O((\log^2 m)/m) \\ &\leq t_b^*(P') + 2 \cdot O((\log^2 m)/m) \\ &\leq (1+\mu)t^* + O((\log^2 m)/m) \\ &\leq t^* \left(1 + \mu + \frac{C \log^2 m}{m}\right) \\ &\leq t^*(1+\epsilon), \end{aligned}$$

for appropriate choices of μ and m, depending on ϵ.

Experimental Studies [with Marcelo Sztainberg]:

Experimental analysis of 3 natural heuristics, including greedy

Analysis of greedy on geometric data: $\Omega(\sqrt{\log n})$-approx

Conclusion – Summary of Results:

1. FTP is NP-hard, even for stars, with one robot per leaf
2. $O(1)$-approx for (general) stars
3. PTAS for stars, same number of robots at each leaf
4. Tight analysis of greedy heuristic on stars: 7/3-approx
5. $o(\log n)$-approx for FTP in ultrametrics ($2^{O(\sqrt{\log\log n})}$-approx)
6. Simple linear-time on-line algorithm, $O(\log \Delta)$-competitive ($\Delta \leq$ max degree)
7. NP-hard to get <5/3-approx in offline problem, even if $\Delta \leq 5$
8. PTAS for geometric instances in fixed-dimension, L_p metric
   Time $O(n \log n + 2^{poly(1/\varepsilon)})$

Conclusion – Open Problems:

OPEN: Is FTP in the Euclidean plane NP-hard?

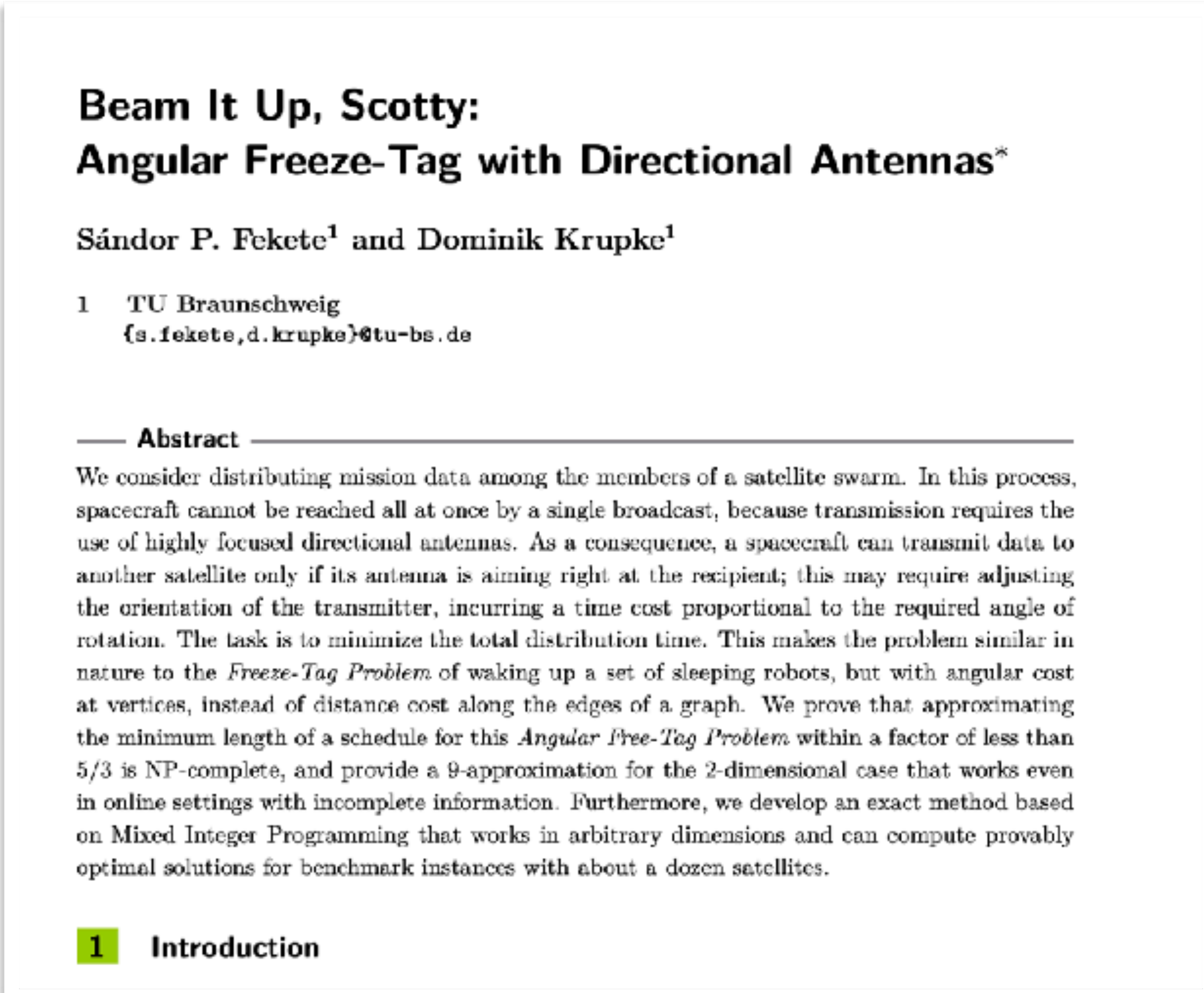
OPEN: Is there an $o(\log n)$-approx for general metric spaces?

1. Introduction
2. Freeze Tag
3. Angular Freeze Tag
4. Angular Scan Cover

Technische Universität Braunschweig

Angular Freeze Tag

Angular Freeze Tag



Beam It Up, Scotty: Angular Freeze-Tag with Directional Antennas*

Sándor P. Fekete[1] and Dominik Krupke[1]

1 TU Braunschweig
{s.fekete,d.krupke}@tu-bs.de

Abstract

We consider distributing mission data among the members of a satellite swarm. In this process, spacecraft cannot be reached all at once by a single broadcast, because transmission requires the use of highly focused directional antennas. As a consequence, a spacecraft can transmit data to another satellite only if its antenna is aiming right at the recipient; this may require adjusting the orientation of the transmitter, incurring a time cost proportional to the required angle of rotation. The task is to minimize the total distribution time. This makes the problem similar in nature to the *Freeze-Tag Problem* of waking up a set of sleeping robots, but with angular cost at vertices, instead of distance cost along the edges of a graph. We prove that approximating the minimum length of a schedule for this *Angular Free-Tag Problem* within a factor of less than 5/3 is NP-complete, and provide a 9-approximation for the 2-dimensional case that works even in online settings with incomplete information. Furthermore, we develop an exact method based on Mixed Integer Programming that works in arbitrary dimensions and can compute provably optimal solutions for benchmark instances with about a dozen satellites.

1 Introduction

Technische Universität Braunschweig

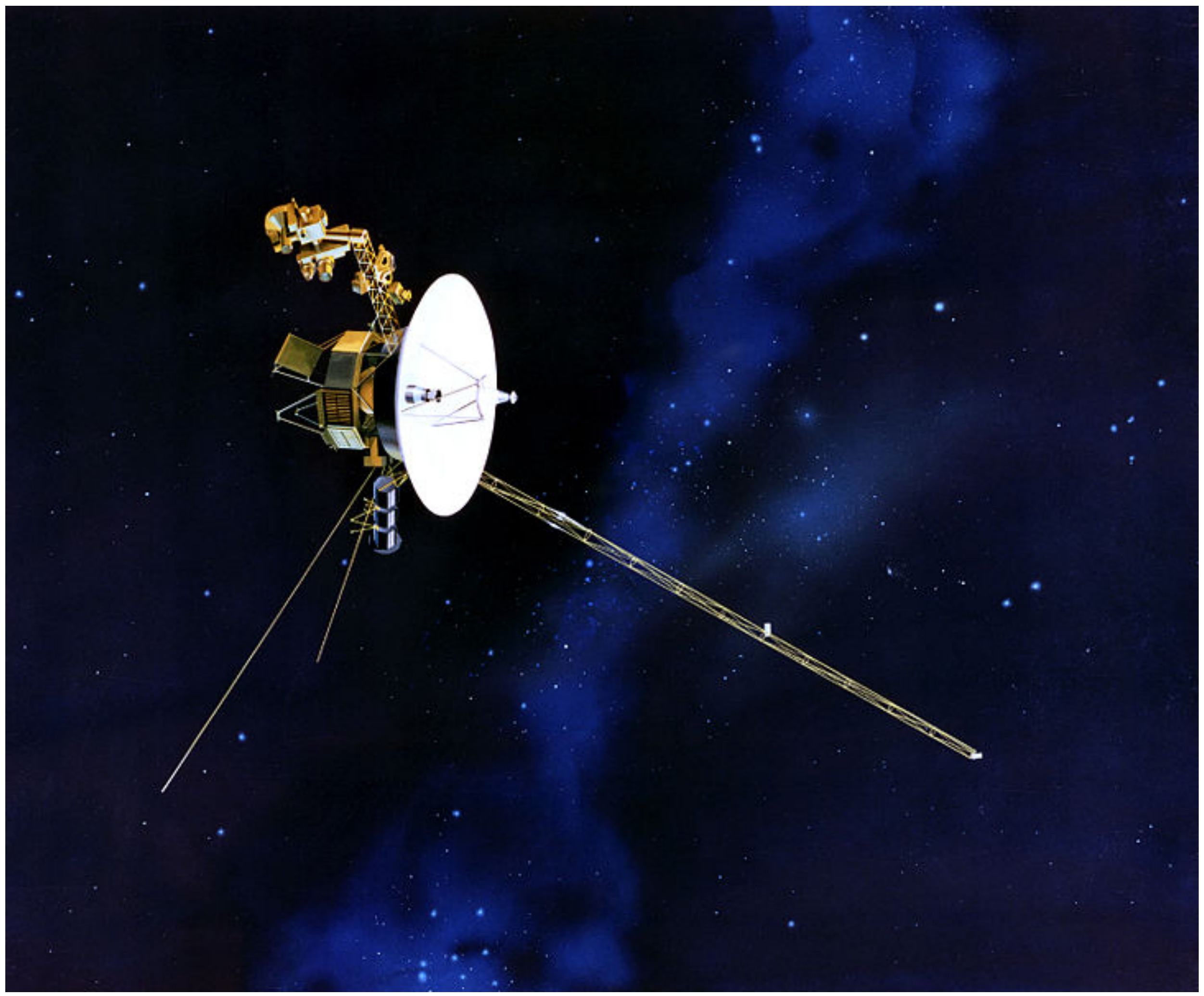
Institute of Operating Systems and Computer Networks

Beam It Up, Scotty: Angular Freeze-Tag with Directional Antennas

Sándor P. Fekete and Dominik Krupke

March 20, 2018

Motivation

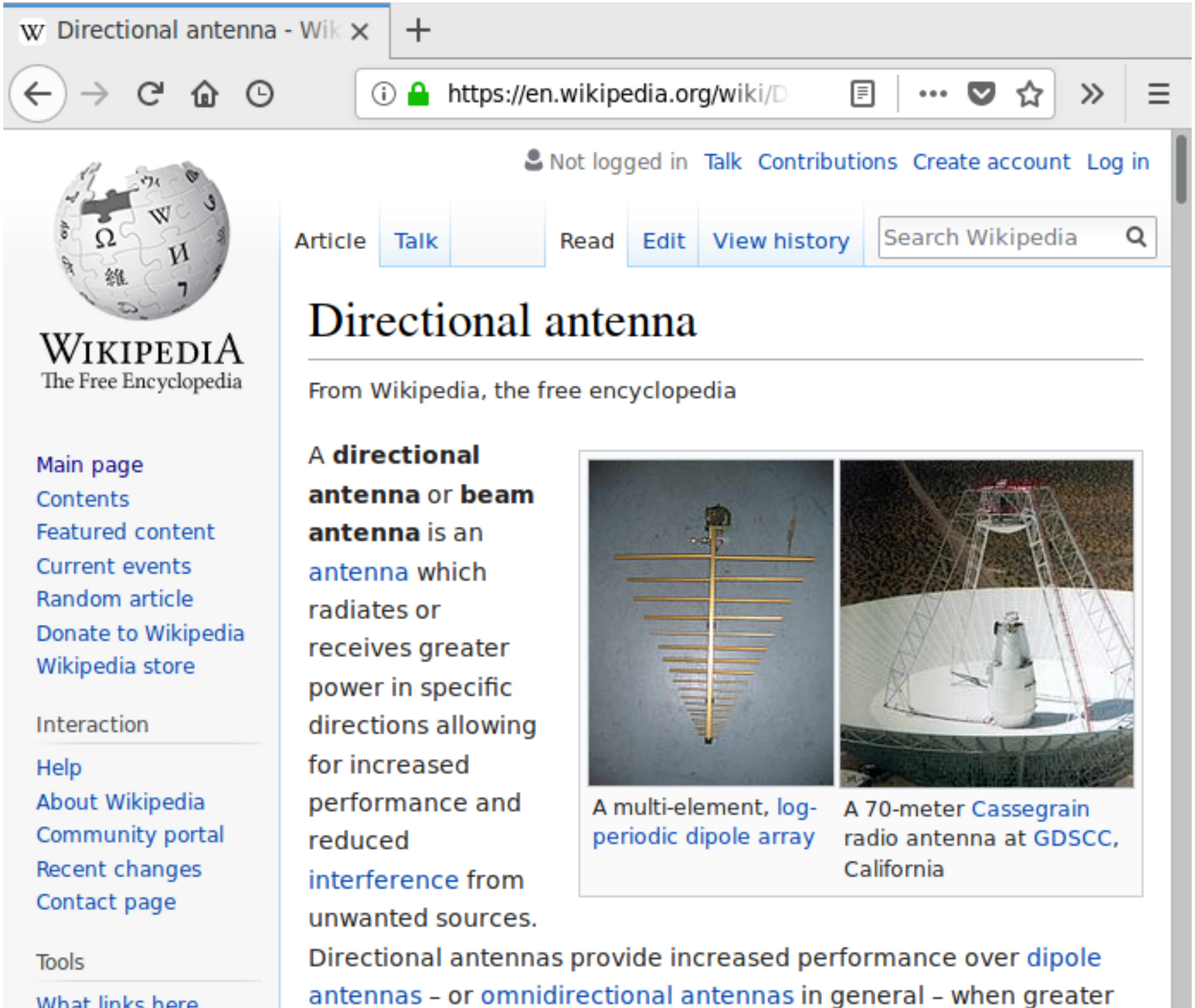


- Highly focused antennas.
- Expensive rotations.

How can we quickly distribute information from one satellite to all others?

d.krupke@tu-bs.de
Institute of Operating Systems and Computer Networks

Wikipedia



Directional antenna - Wik
https://en.wikipedia.org/wiki/D
Not logged in Talk Contributions Create account Log in
WIKIPEDIA
The Free Encyclopedia
Article Talk
Read Edit View history
Search Wikipedia
Directional antenna
From Wikipedia, the free encyclopedia
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page
Tools
What links here
A directional antenna or beam antenna is an antenna which radiates or receives greater power in specific directions allowing for increased performance and reduced interference from unwanted sources.
A multi-element, log-periodic dipole array
A 70-meter Cassegrain radio antenna at GDSCC, California
Directional antennas provide increased performance over dipole antennas – or omnidirectional antennas in general – when greater

Wikipedia

Directional antenna - Wik

https://en.wikipedia.org/wiki/D

the antenna but for small antennas can be increased by adding a ferrite rod), and efficiency (again, affected by size, but also resistivity of the materials used and impedance matching). These factors are easy to improve without adjusting other features of the antennas or coincidentally improved by the same factors that increase directivity, and so are typically not emphasized.

Applications [edit]

High gain antennas are typically the largest component of deep space probes, and the highest gain radio antennas are physically enormous structures, such as the Arecibo Observatory. The Deep Space Network uses 35 m dishes at about 1 cm wavelengths. This combination gives the antenna gain of about 100,000,000 (or 80 dB, as normally measured), making the transmitter appear about 100 million times stronger, and a receiver about 100 million times more sensitive, *provided the target is within the beam*. This beam can cover at most one hundred millionth (10^{-8}) of the sky, so very accurate pointing is required.

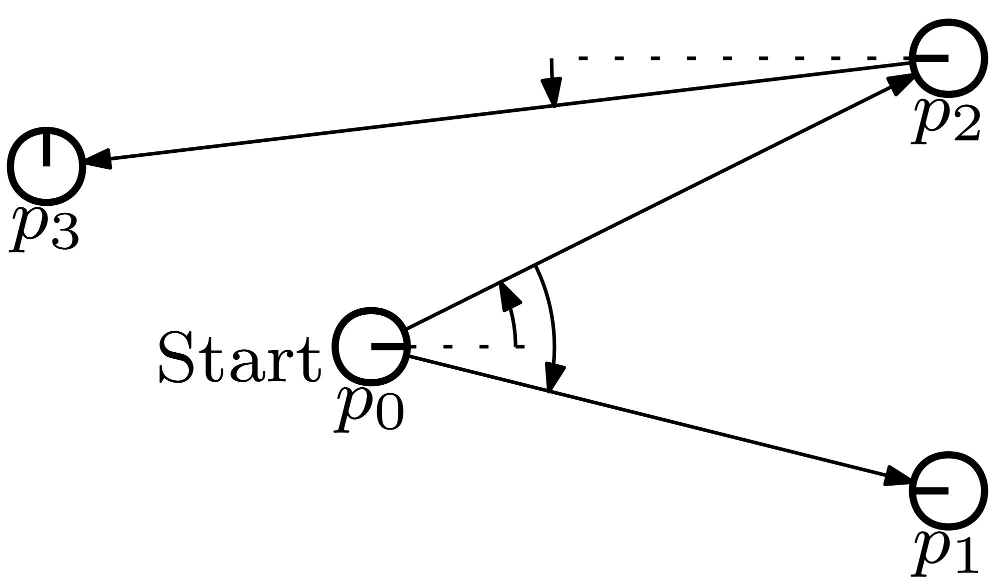
Gallery [edit]

Problem Description

Distribute data from one satellite to all other with minimal makespan.
Informed/activated satellites can participate in the distribution.

Some simplifications:

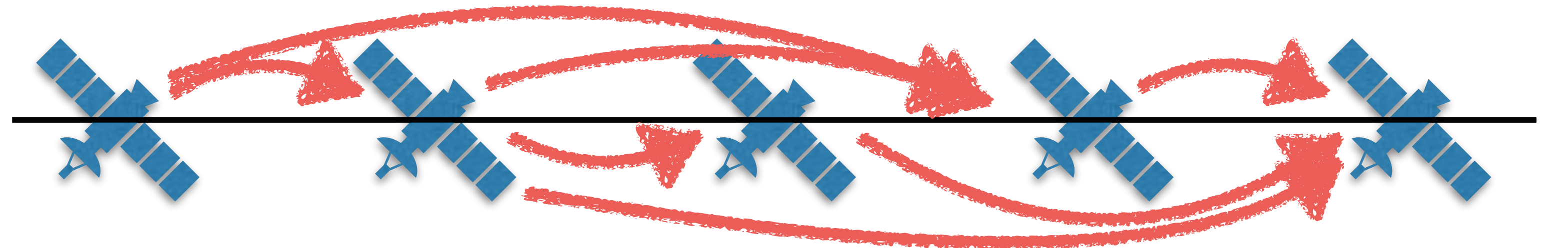
1. Only sender has to adjust.
2. Exact adjustment, i.e. beam is a ray.
3. Fixed positions.
4. Geometric plane.
5. Negligible transmission time.
6. Rotation time equal rotation angle.



d.krupke@tu-bs.de
Institute of Operating Systems and Computer Networks

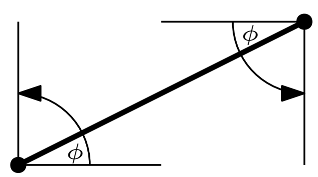
Summary

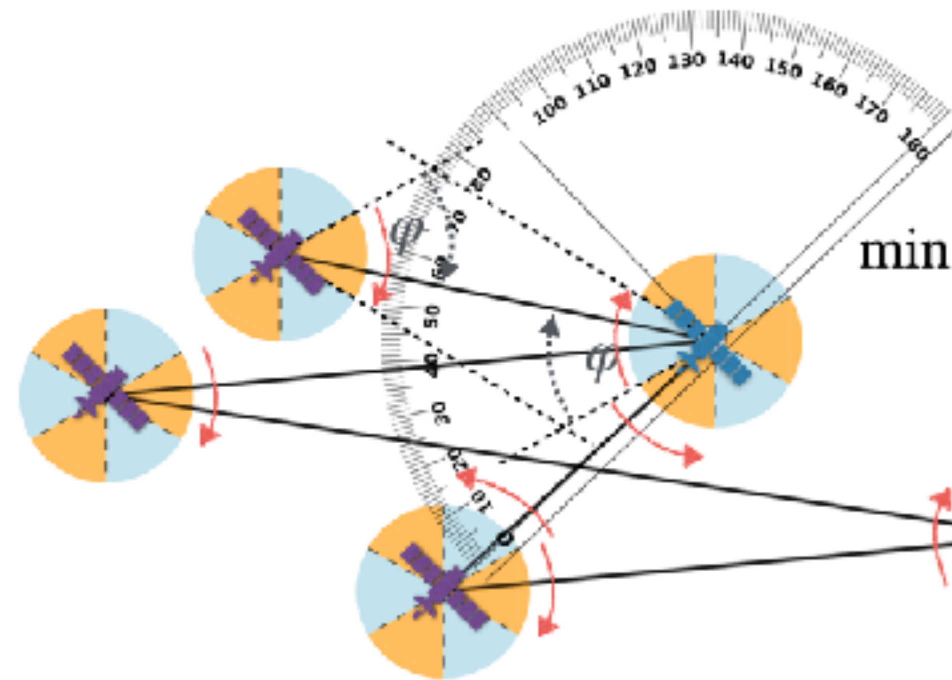
1D



$\Theta(\lceil \log_2 \chi(G) \rceil)$

2D

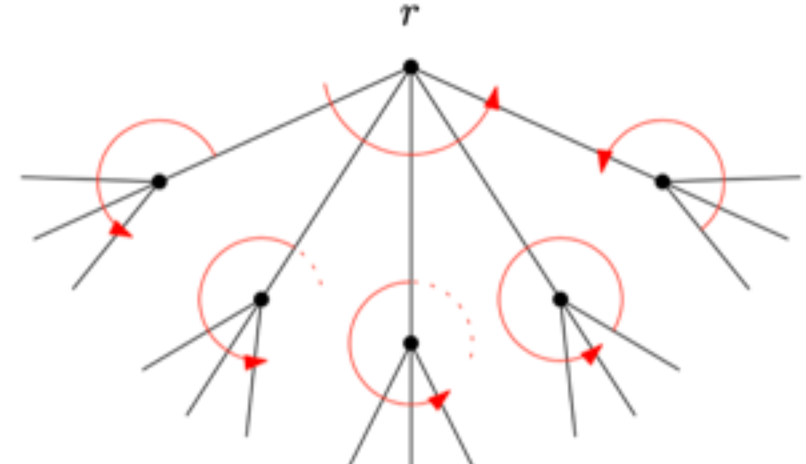




$\min \frac{360^\circ}{2s} \geq 55^\circ \Rightarrow 60^\circ$

$\Theta(\lceil \log_2 \chi(G) \rceil)$

3D & Abstract



$\Omega(\lceil \log_2 \chi(G) \rceil)$

Thank you!