ANTS on a Plane

Jared Saia

Joint with Abhinav Aggarwal

ANTS on a Plane

Jared Saia

Joint with Abhinav Aggarwal

ANTS on a Plane

Jared Saia

Joint with Abhinav Aggarwal

ANTS on a Plane

Jared Saia

Joint with Abhinav Aggarwal

ANTS problem

N agents start at node (*nest)* on infinite grid

Target is placed on node at distance **L**

Goal: Find the target ASAP

ANTS problem

N agents start at node (*nest)* on infinite grid

Target is placed on node at distance **L**

Goal: Find the target ASAP

Synchronous agents; no communication
Advice: bits to encode #agents, roles, etc

[Feinerman and Korman, 2017]

ANTS Results

N agents start at node (*nest*) on infinite grid

Target is placed on node at distance **L**

Advice: bits to encode #agents, roles, etc

ANTS Results

N agents start at node (*nest*) on infinite grid

Target is placed on node at distance **L**

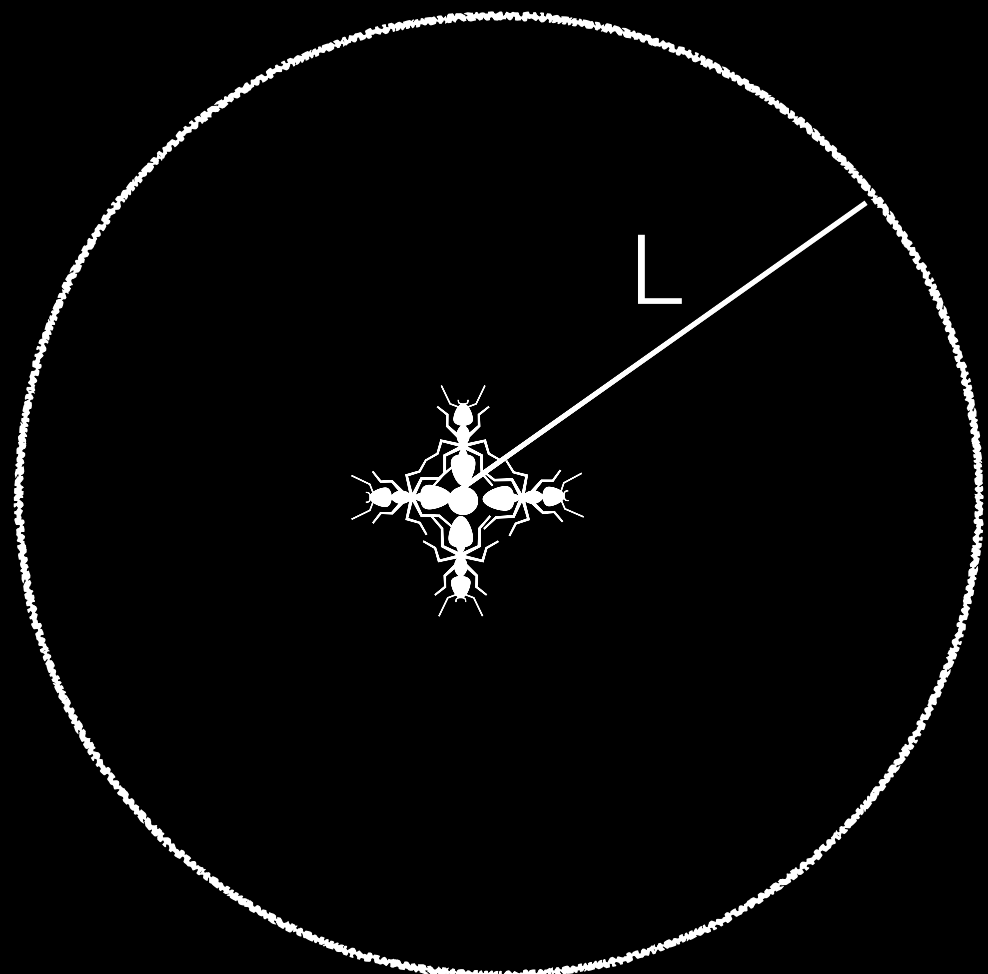
Advice: bits to encode #agents, roles, etc

$O(L + L^2/N)$ time with $O(\log \log N)$ bits advice

$O((L + L^2/N)\log^{1+\epsilon} N)$ time with no advice, for any fixed $\epsilon > 0$

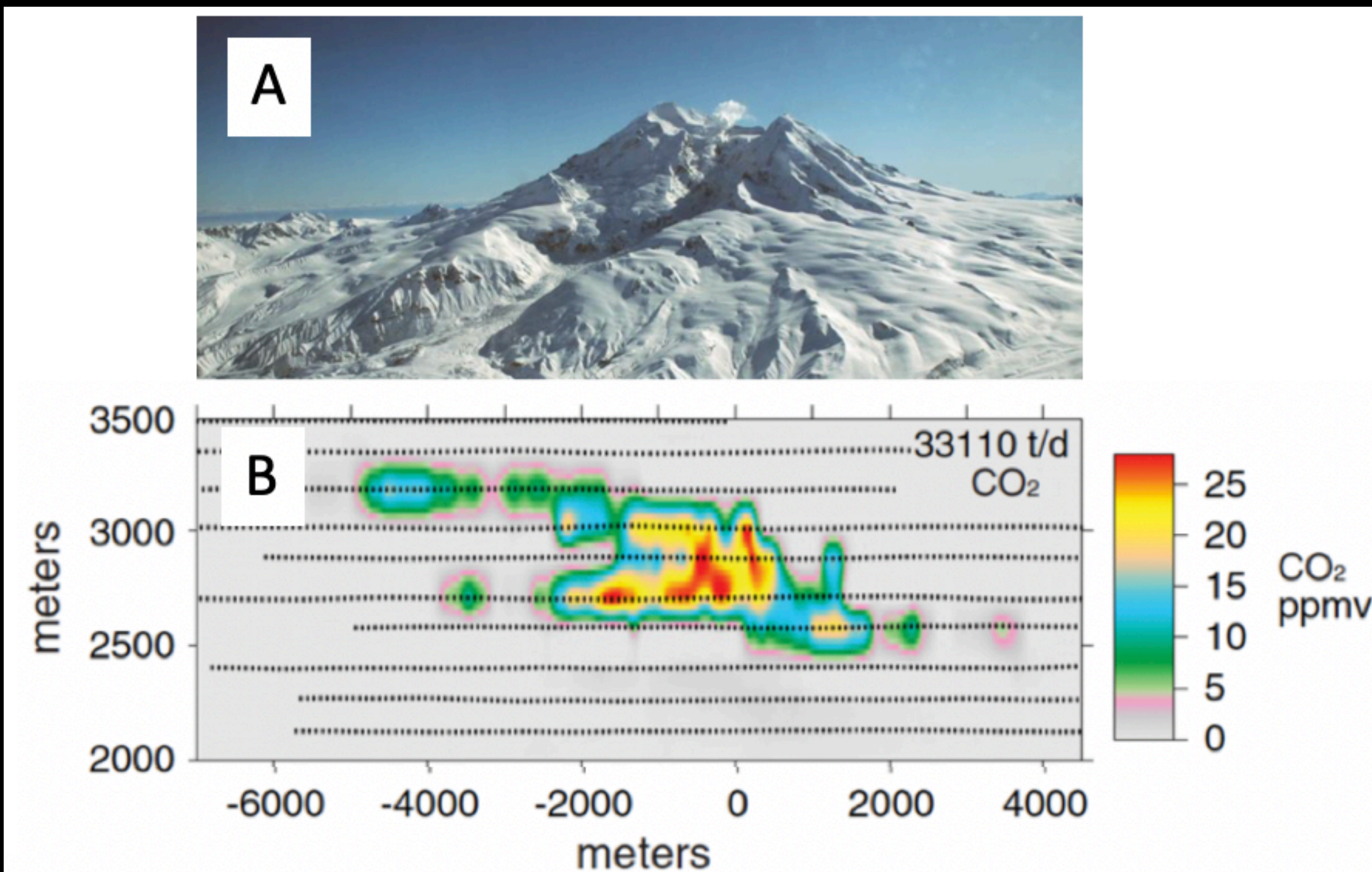
[Feinerman and Korman, 2017]

$O(L + L^2/N)$ is Optimal



Area to Search $\approx L^2$
N agents
Need $\approx L^2/N$ to search area
Plus L time to reach target

Motivation: Drones seek CO_2



Melanie Moses

moseslab.cs.unm.edu

Use grid to approximate plane?

Use grid to approximate plane?

Problems:

- Choosing grid granularity
  - Too low: May miss target
  - Too high: Computational load on agents
- Hard to handle different target shapes

Use grid to approximate plane?

Problems:

- Choosing grid granularity
  - Too low: May miss target
  - Too high: Computational load on agents
- Hard to handle different target shapes

Solution: Formulate problem on Euclidean plane

Target Shape

What target shape can we handle?

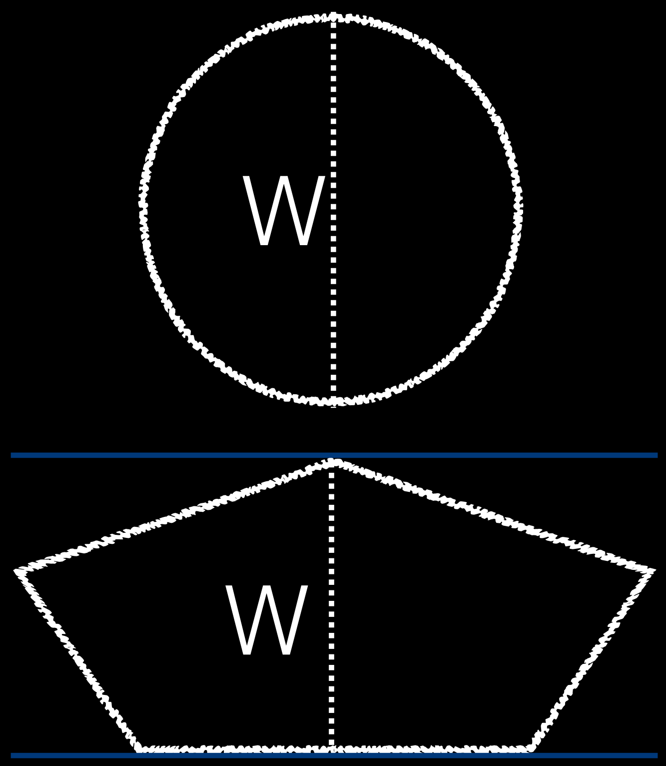
Target Shape

What target shape can we handle?

A convex shape

Width: Smallest distance between two parallel lines touching boundary but not interior

L is distance to segment W. Assume $W \leq L$.



Our Result - No Advice

$$O\left(\left(L + \left(\frac{L^2}{NW}\right)\right)\log L\right)$$ search time

[F&K '15] $O((L + L^2/N)\log^{1+\epsilon} N)$ time, for any $\epsilon > 0$

Our Result - No Advice

$$O\left(\left(L+\left(\frac{L^2}{NW}\right)\right)\log L\right)$$ search time

[F&K '15] $O((L+L^2/N)\log^{1+\epsilon} N)$ time, for any $\epsilon > 0$

If $t < N$ agents removed by adversary

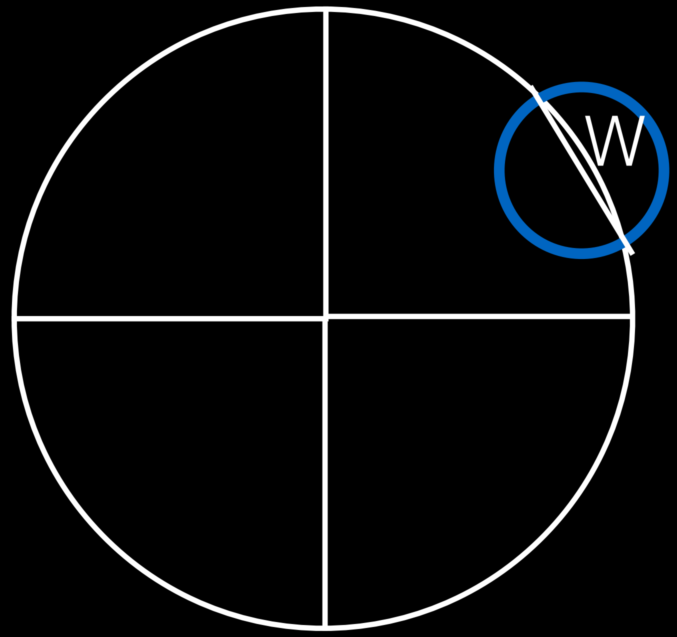
$$O\left(\left(L+\frac{L^2(t+1)}{NW}\right)\log L\right)$$ search time

Spokes

Spoke: line segment from nest and back

Say target is on circle of circumference 1

How many spokes are needed to find it?

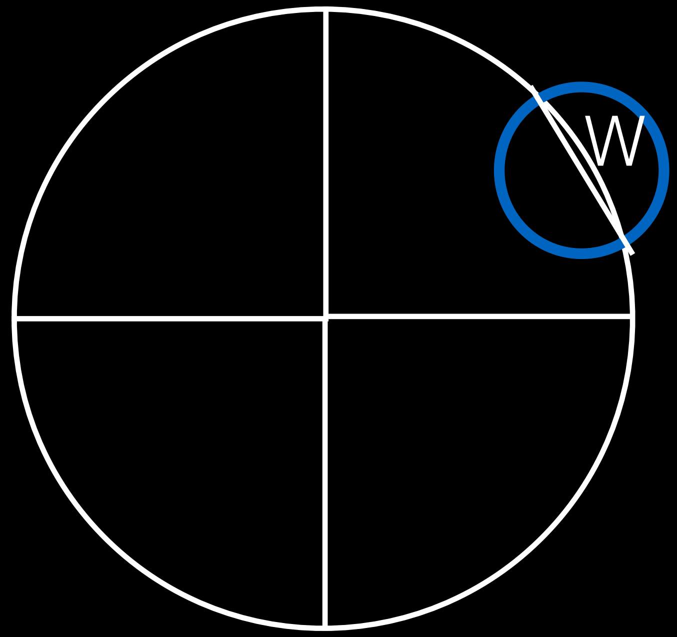


Spokes

Spoke: line segment from nest and back

Say target is on circle of circumference 1

How many spokes are needed to find it?



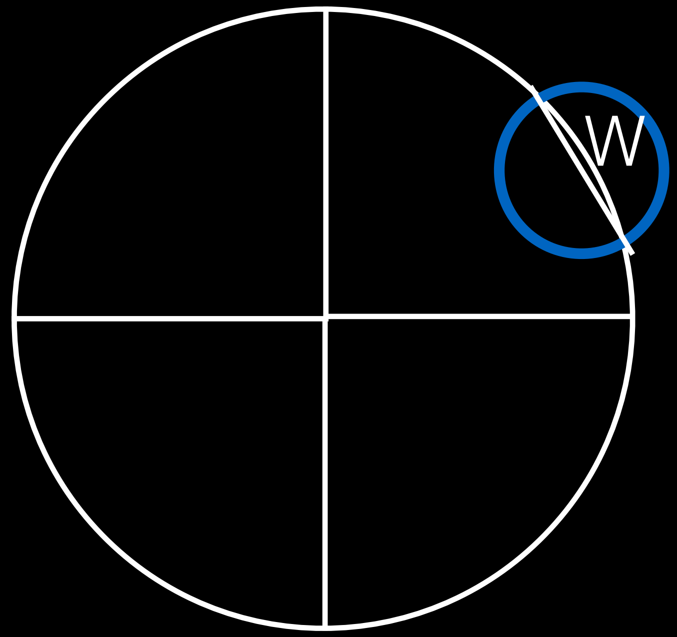
Suffices: 1/W evenly spaced spokes

Spokes

Spoke: line segment from nest and back

Say target is on circle of circumference 1

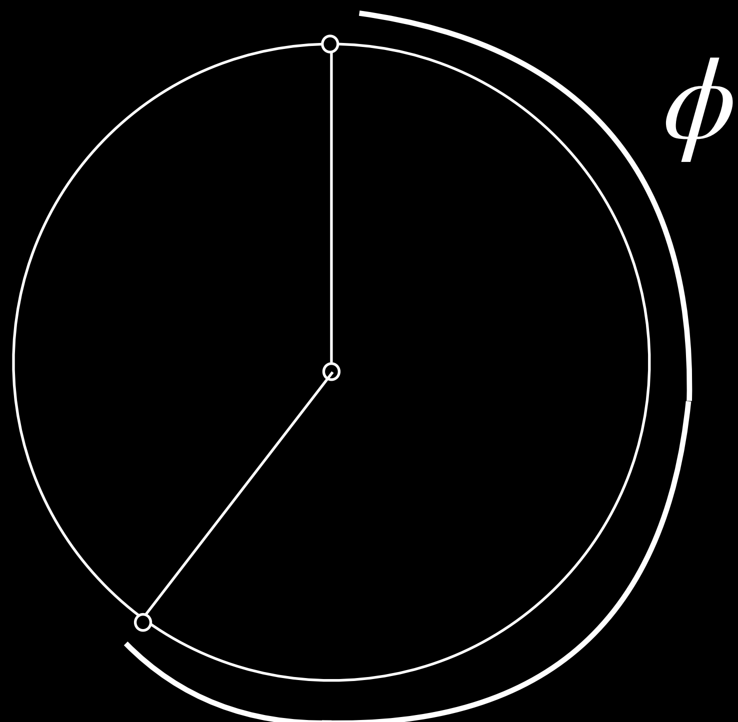
How many spokes are needed to find it?



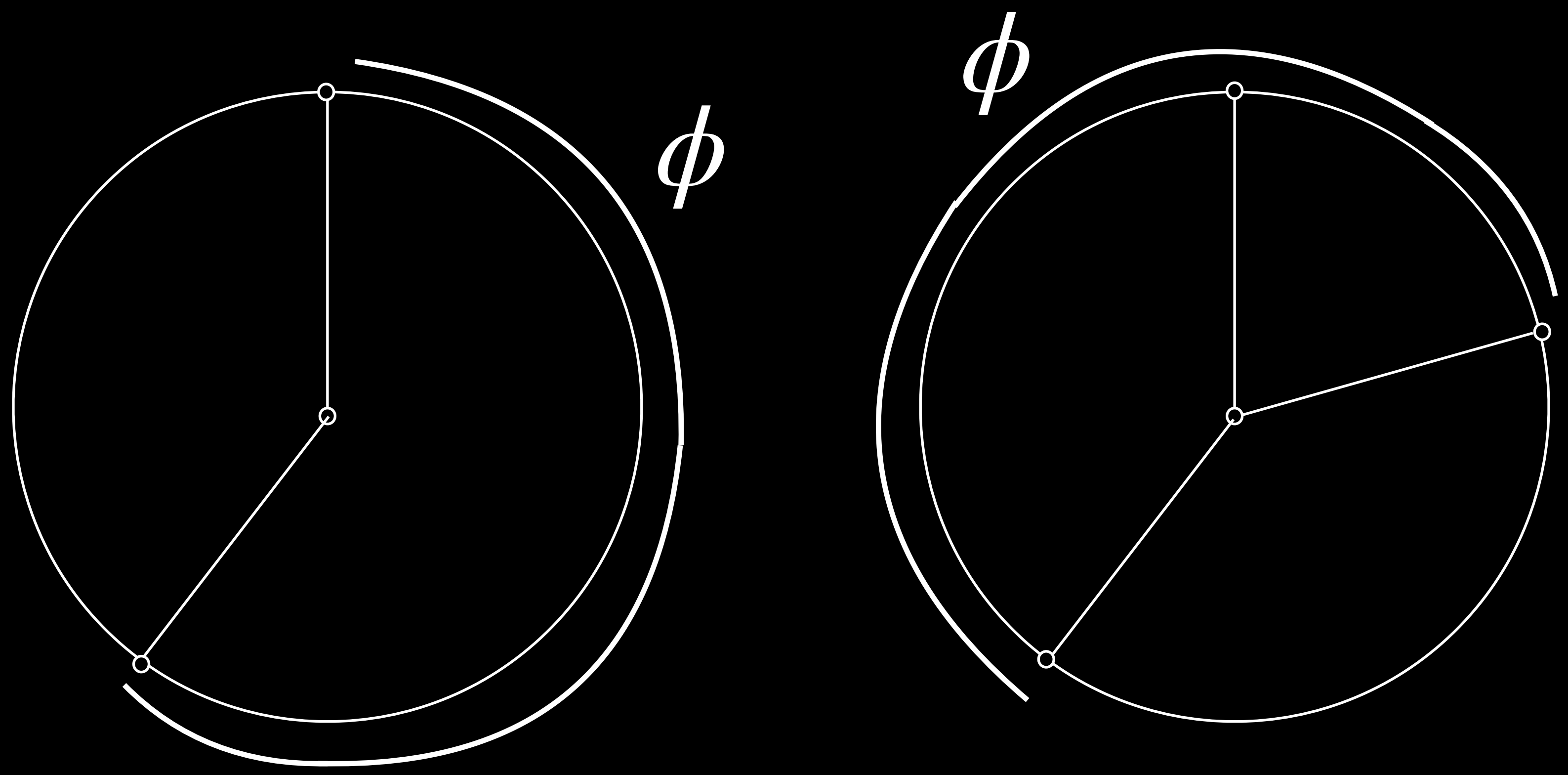
Suffices: 1/W evenly spaced spokes

How to get “evenly” space spokes, when don’t know W in advance?

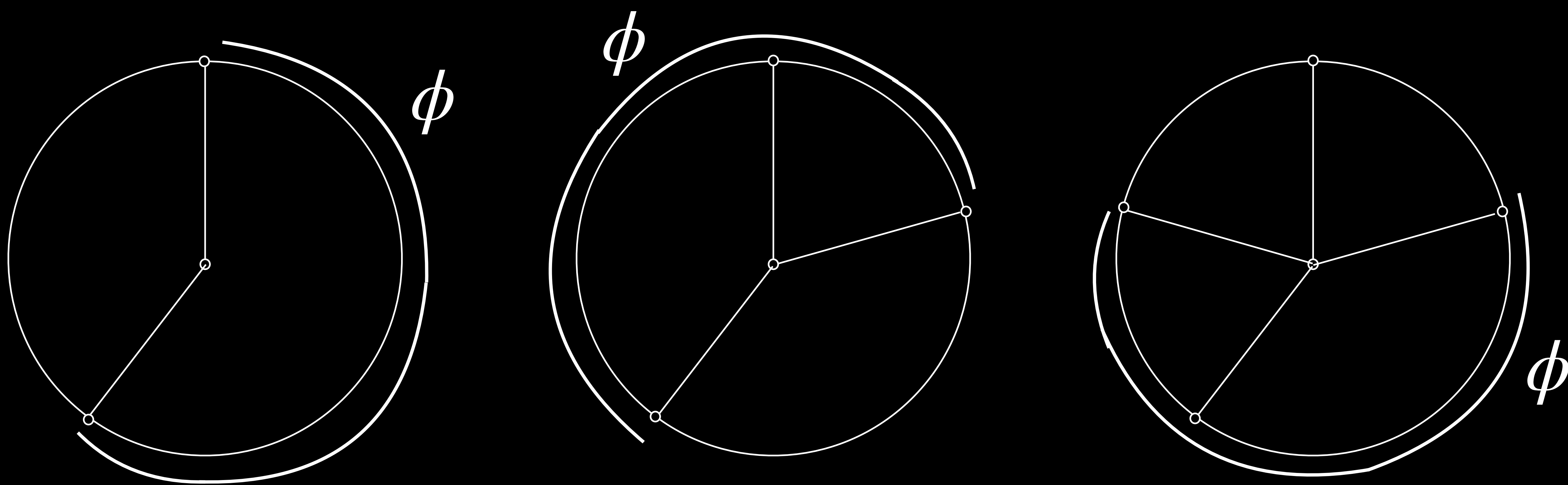
Idea: Use ϕ



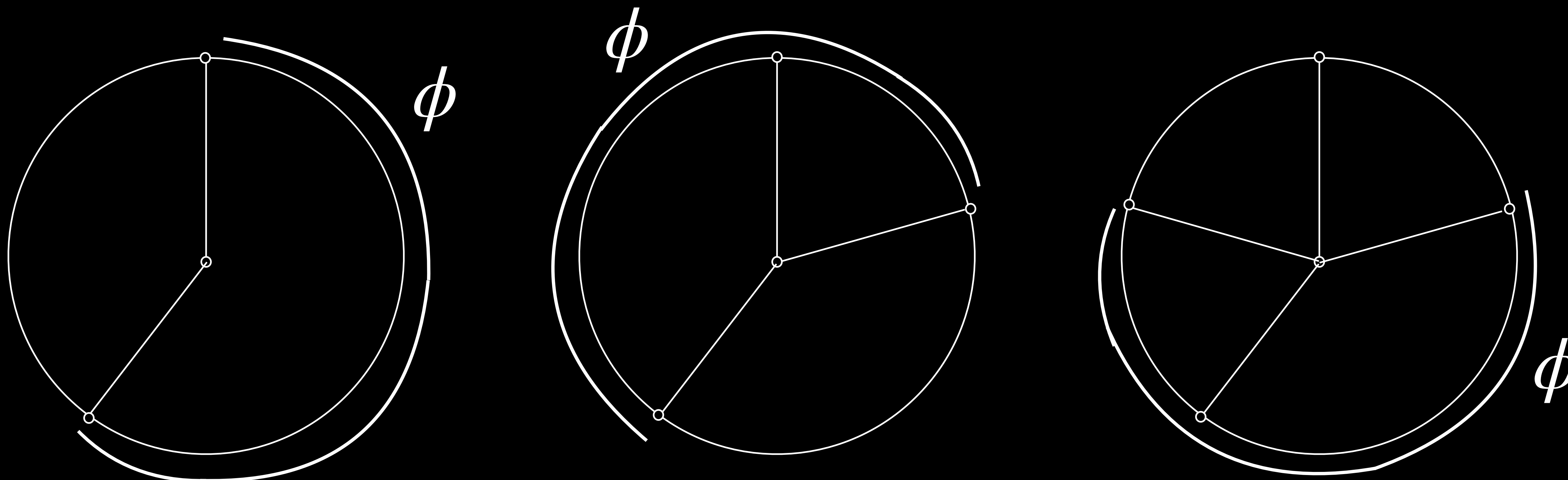
Idea: Use ϕ



Idea: Use ϕ



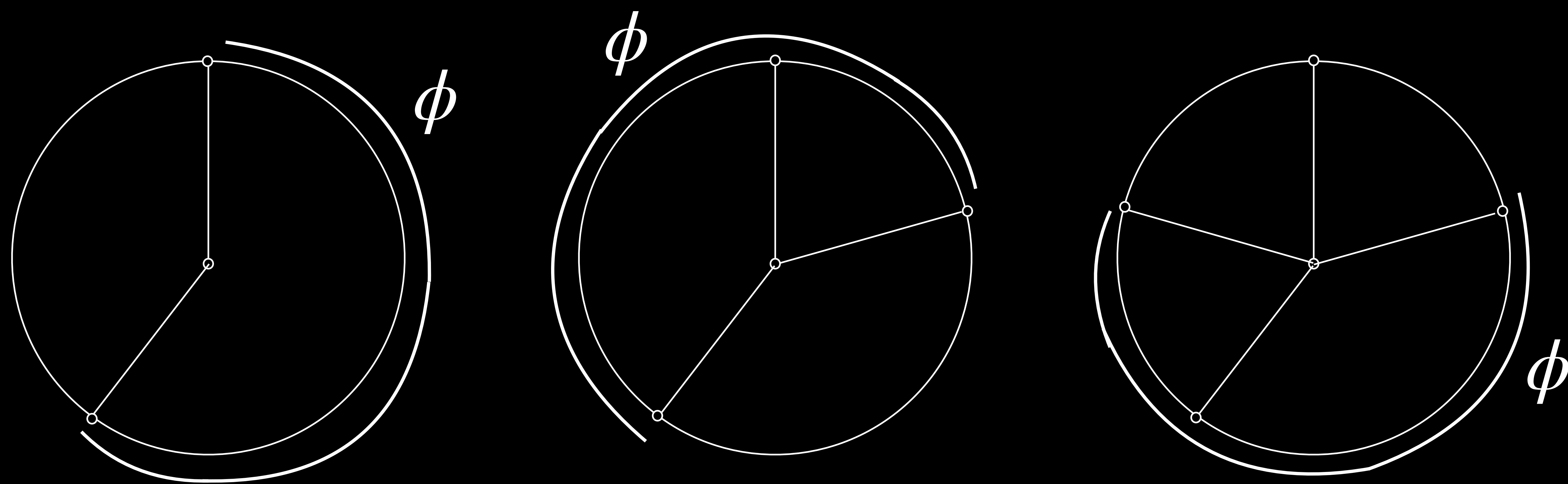
Idea: Use ϕ



[Swierczkowski, ’58] M points placed at arc distance ϕ → arc length between any neighboring points is O(1/M)

Moreover, ϕ minimizes hidden constant over all numbers

Idea: Use ϕ



arc lengths between neighboring spokes are $\approx 1/4$

[Swierczkowski, ’58] M points placed at arc distance $\phi \rightarrow$ arc length between any neighboring points is O(1/M)

Moreover, ϕ minimizes hidden constant over all numbers

Why does $\phi \to$ well-spread?

Why does $\phi \rightarrow$ well-spread?

Any number can be written as: $x_1 + \cfrac{1}{x_2 + \cfrac{1}{x_3 + \ldots}}$

where x_i are integers.

Why does $\phi \to$ well-spread?

Any number can be written as: $x_1 + \cfrac{1}{x_2 + \cfrac{1}{x_3 + \ldots}}$

where x_i are integers.

Small x_i means rational approximation hard

Why does $\phi \rightarrow$ well-spread?

Any number can be written as: $x_1 + \cfrac{1}{x_2 + \cfrac{1}{x_3 + \ldots}}$

where x_i are integers.

Small x_i means rational approximation hard

“Hardest to approximate” number → all $x_i = 1$

Why does $\phi \rightarrow$ well-spread?

Any number can be written as: $x_1 + \cfrac{1}{x_2 + \cfrac{1}{x_3 + \ldots}}$

where x_i are integers.

Small x_i means rational approximation hard

"Hardest to approximate" number → all $x_i = 1$

To get this, set $y = 1 + \frac{1}{y}$. Solving yields: $y = \phi$

Why does $\phi \to$ well-spread?

Any number can be written as: $x_1 + \cfrac{1}{x_2 + \cfrac{1}{x_3 + \ldots}}$

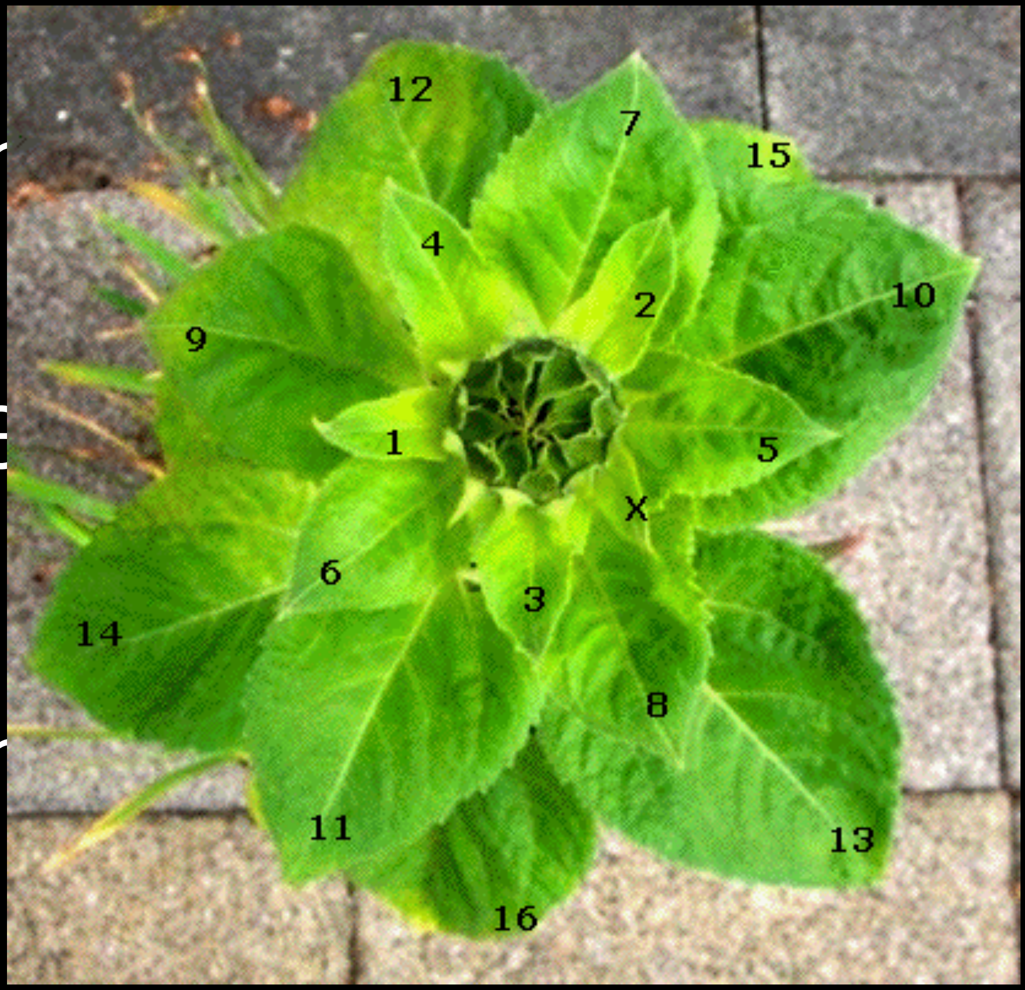
where x_i are integers.

"very irrational" →well-spread

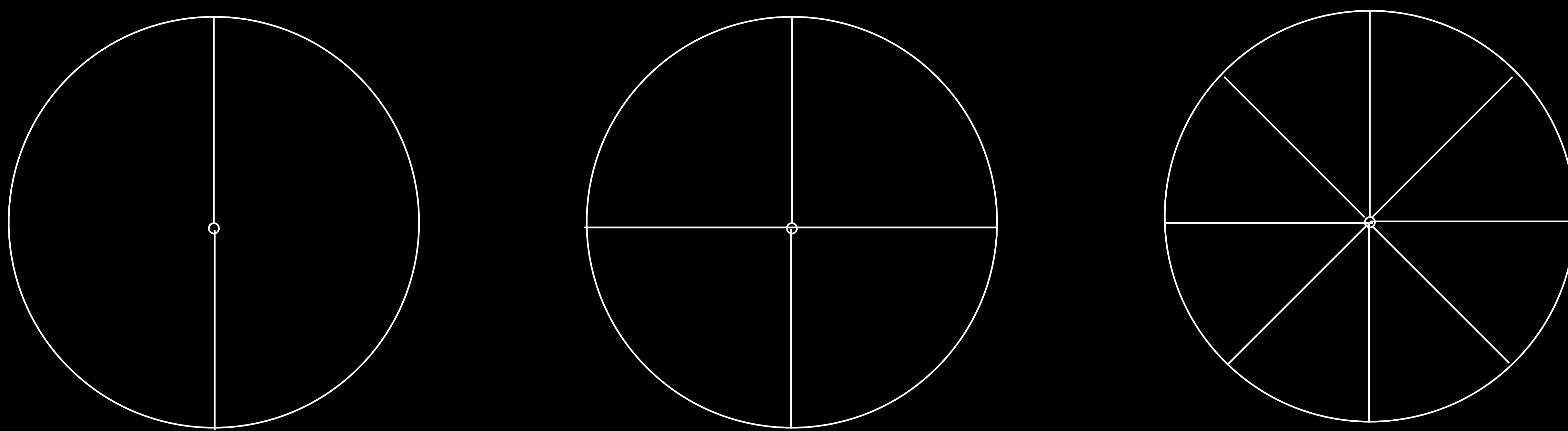
Small x_i means rational approxi

"Hardest to approximate" numbe

To get this, set $y = 1 + \frac{1}{y}$. Solvi

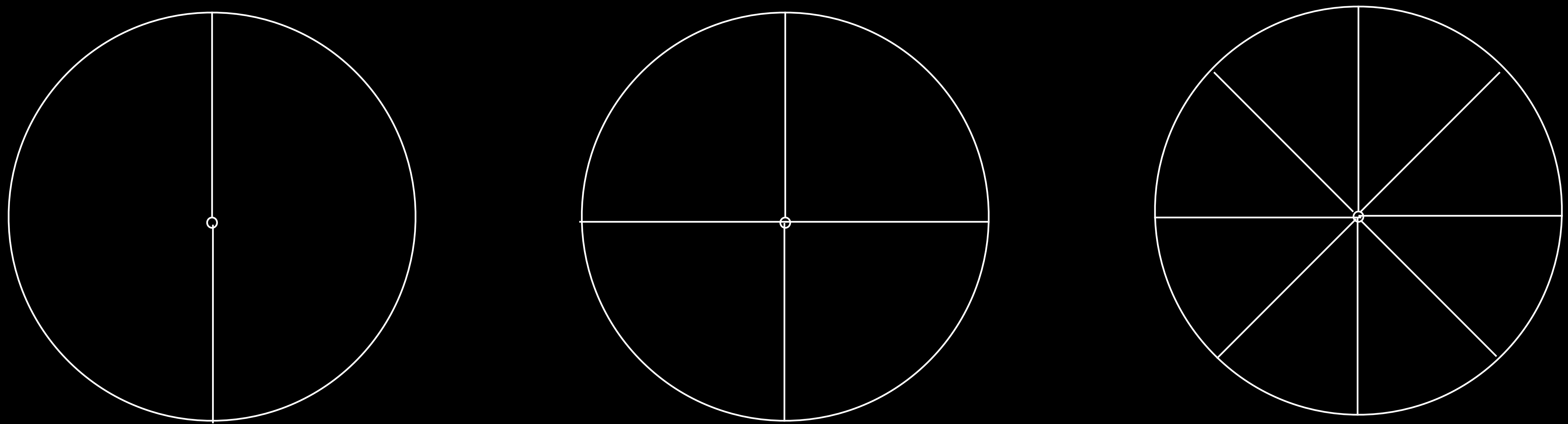


Why not powers of 2?



Q: Why not have 2^i evenly space spokes in iteration i?

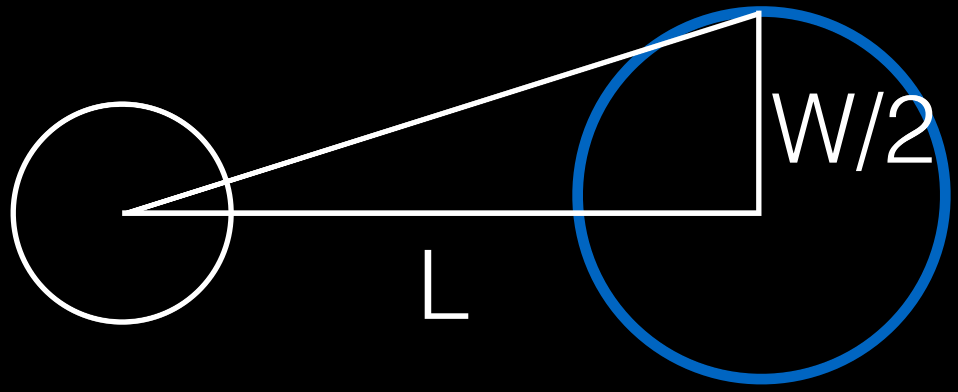
Why not powers of 2?



Q: Why not have 2^i evenly space spokes in iteration i?

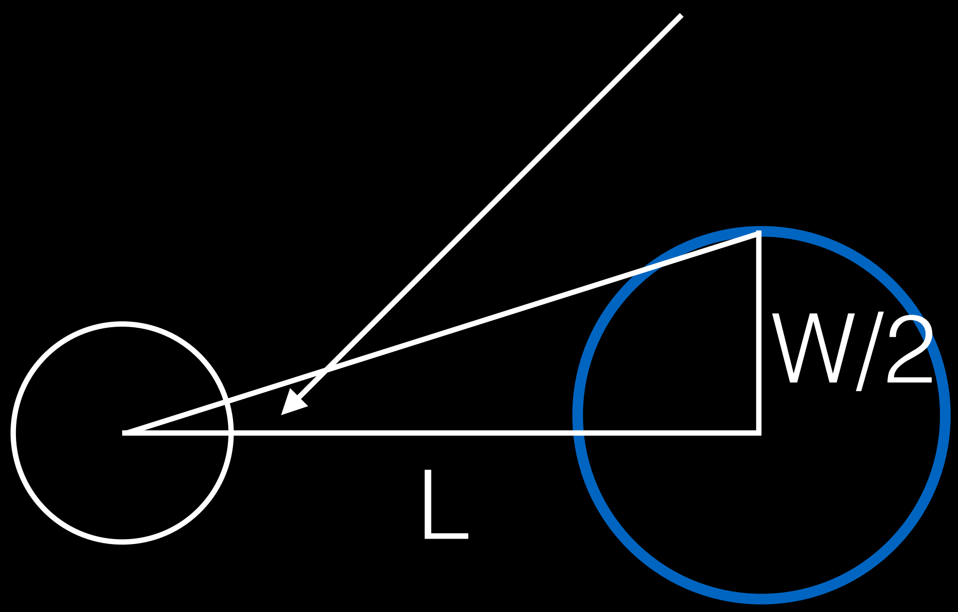
A1: Off from optimal number of spokes by $\leq$ factor of 2
A2: Requires memory for the counter, and also adds algorithmic complexity.

How many spokes?



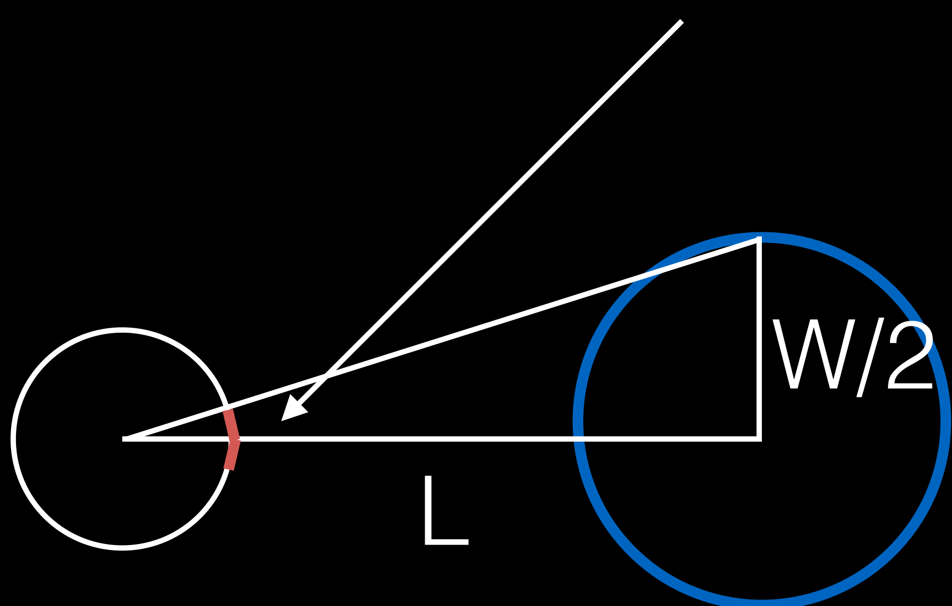
How many spokes?

$$\theta = \sin^{-1} \frac{W}{2L}$$



How many spokes?

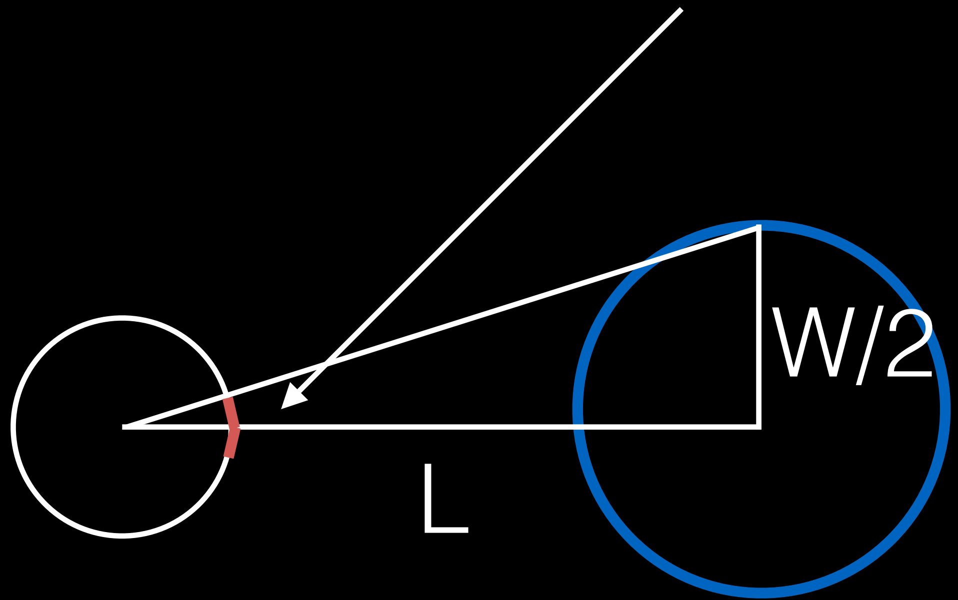
$$\theta = \sin^{-1} \frac{W}{2L}$$



Let α be arc length on unit circle

How many spokes?

$$\theta = \sin^{-1}\frac{W}{2L}$$

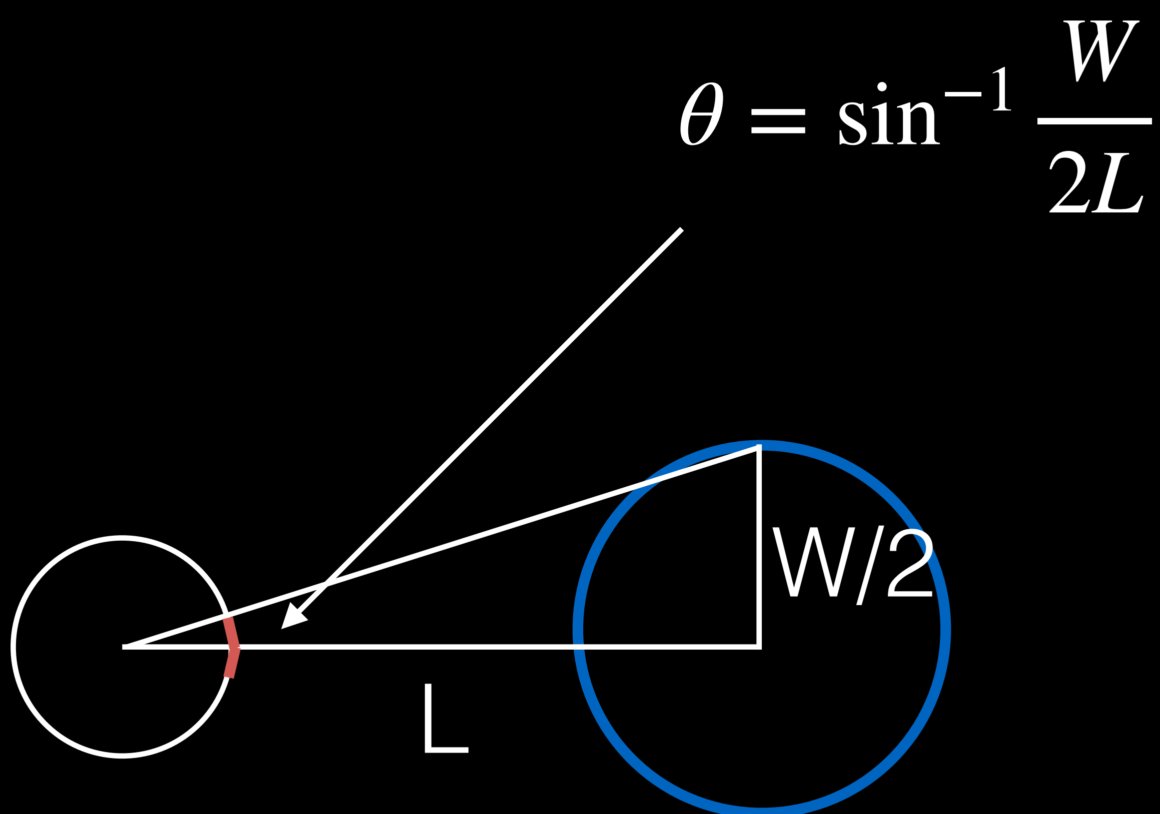


Let α be arc length on unit circle

$$\alpha = \frac{1}{\pi}\sin^{-1}\frac{W}{2L}$$

Power Series: $\sin^{-1} x \geq x$, for $|x| \leq 1$

How many spokes?



Let α be arc length on unit circle

$$\alpha = \frac{1}{\pi}\sin^{-1}\frac{W}{2L}$$

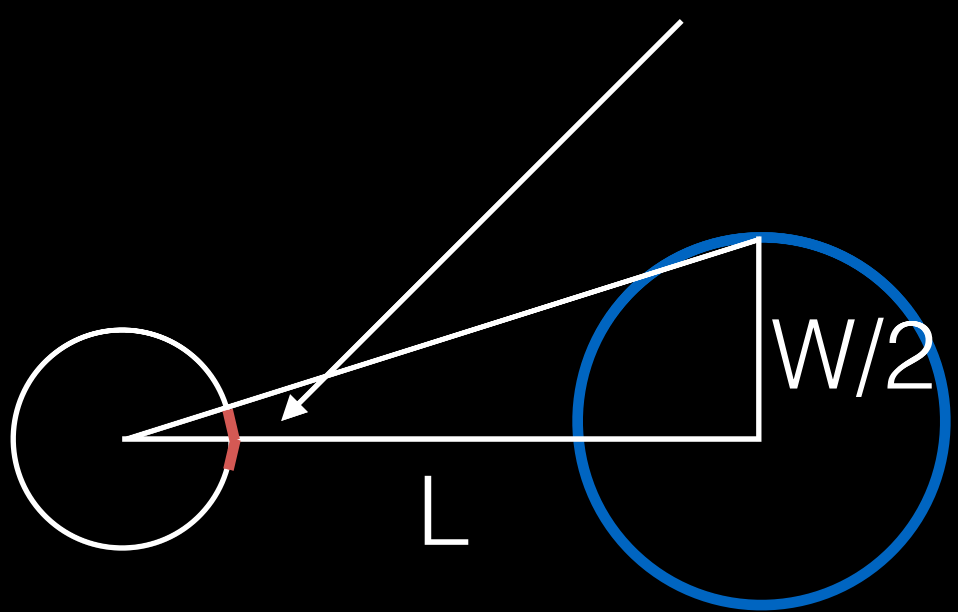
Power Series: $\sin^{-1} x \geq x$, for $|x| \leq 1$

Thus, $\alpha \geq \dfrac{W}{2\pi L}$

So by [Swierczkowski, '58], need O(L/W) spokes

How many spokes?

$\theta = \sin^{-1} \frac{W}{2L}$



$O(L^2/W)$ search time!
If know L in advance

Let α be arc length on unit circle

$$\alpha = \frac{1}{\pi} \sin^{-1} \frac{W}{2L}$$

Power Series: $\sin^{-1} x \geq x$, for $|x| \leq 1$

Thus, $\alpha \geq \frac{W}{2\pi L}$

So by [Swierczkowski, '58], need O(L/W) spokes

How to handle unknown L?

Spoke Length

spokes

	1	2	4	8	16
1					
2					
4					
8					
16					

How to handle unknown L?

# spokes \ Spoke Length	1	2	4	8	16
1					
2					
4					
8					
16					

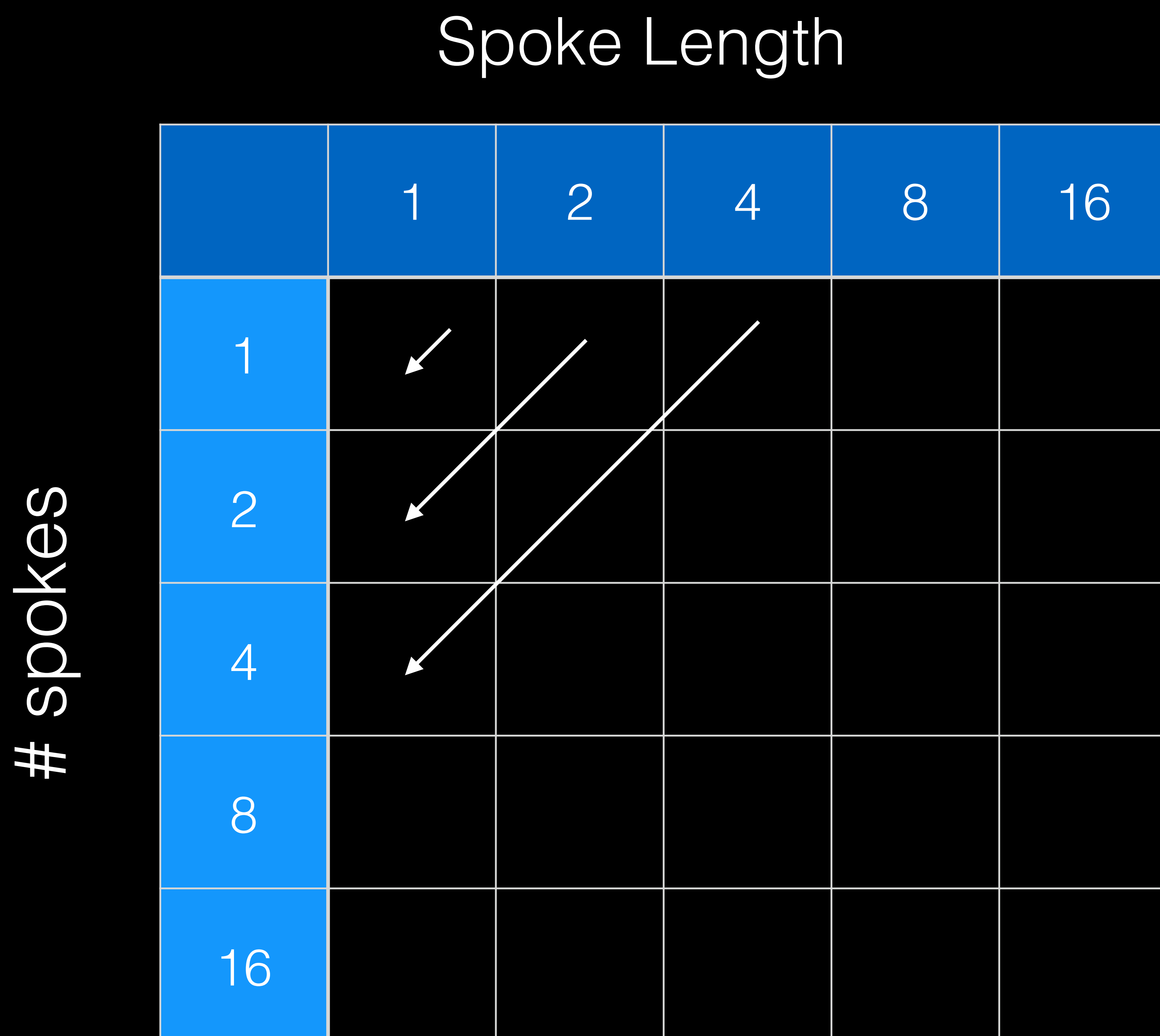
How to handle unknown L?

Spoke Length

spokes

	1	2	4	8	16
1					
2					
4					
8					
16					

How to handle unknown L?



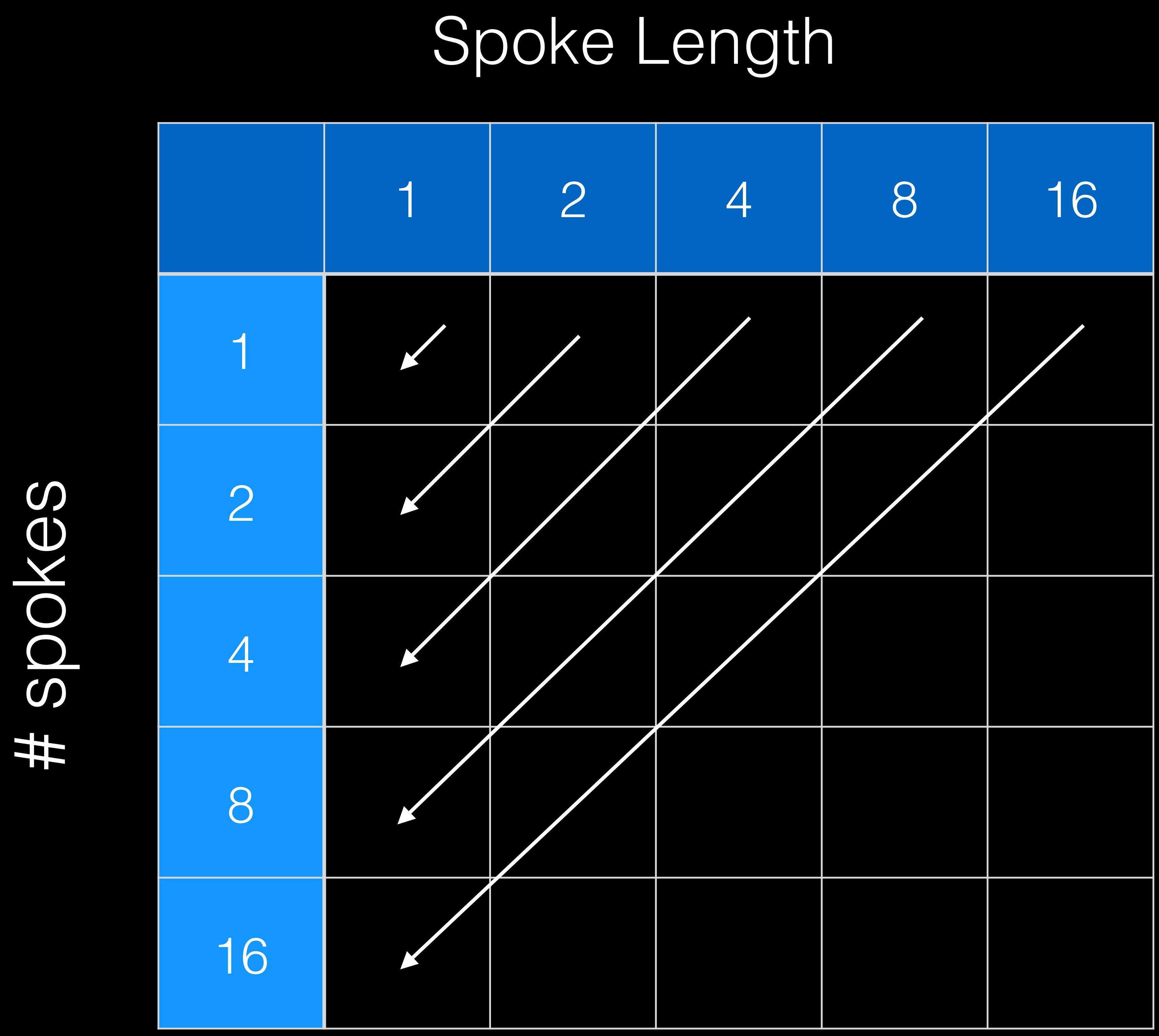
How to handle unknown L?

Spoke Length

spokes

	1	2	4	8	16
1					
2					
4					
8					
16					

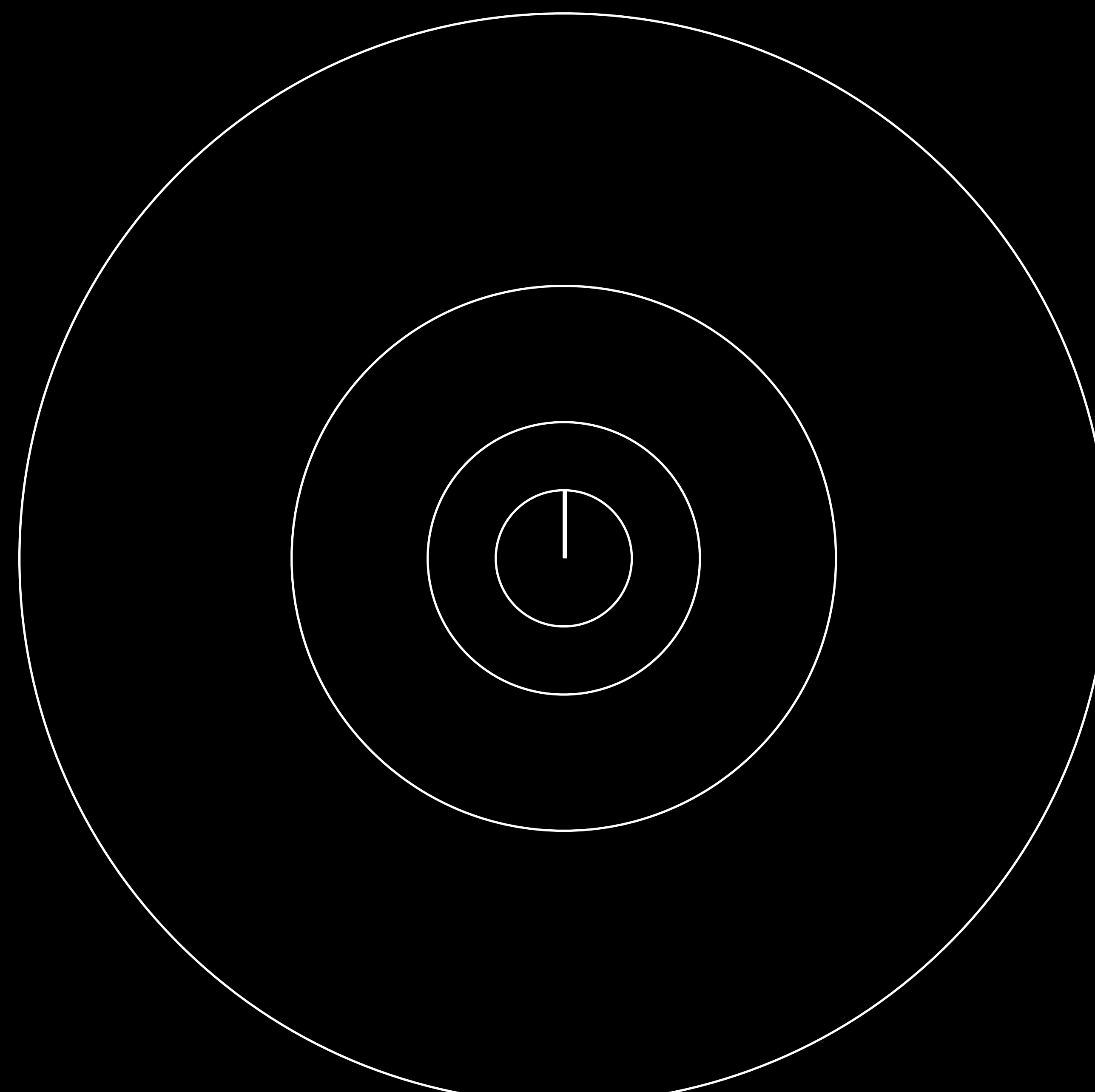
How to handle unknown L?



Spoke Length

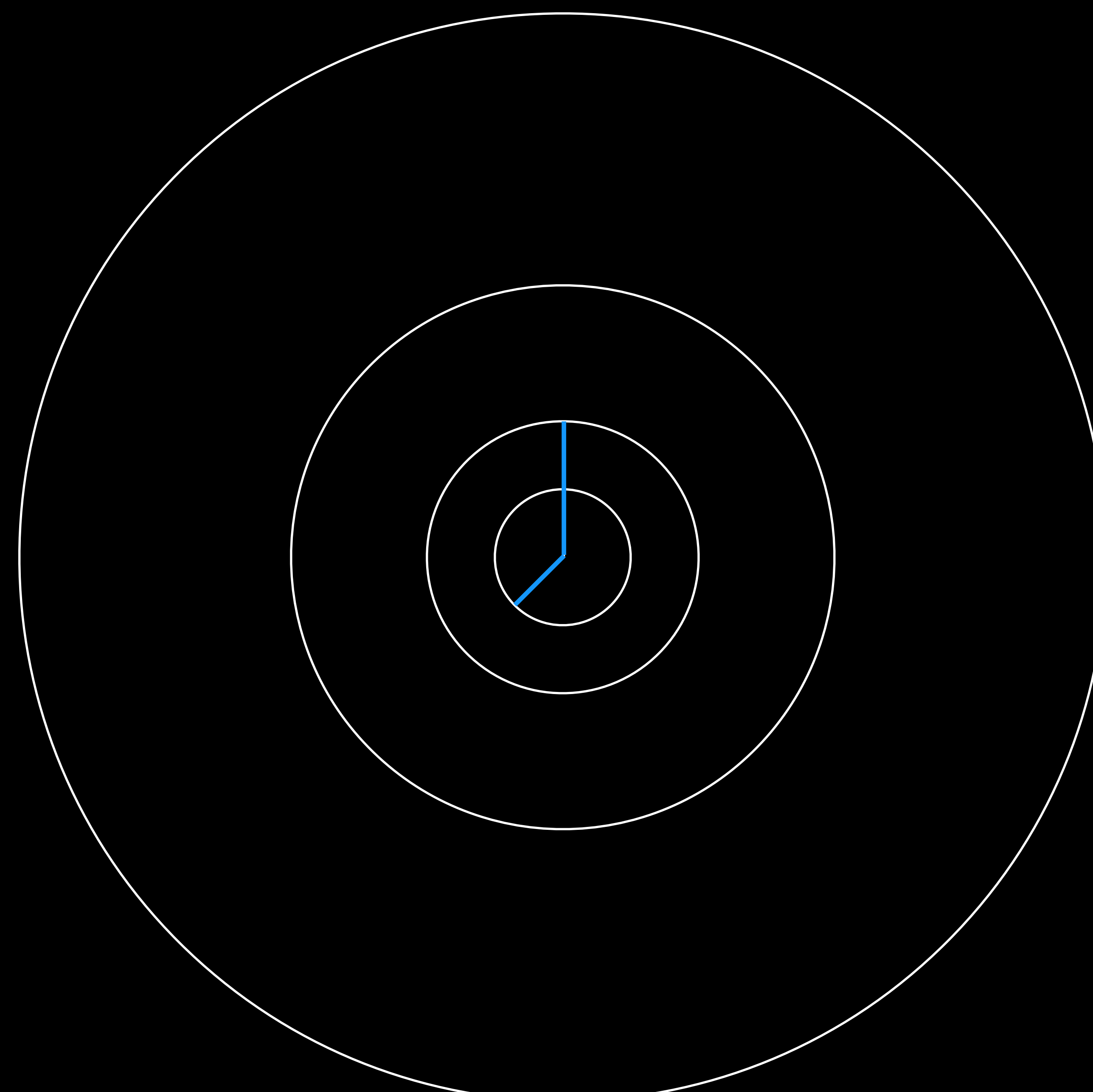
spokes

	1	2	4	8	16
1	↙				
2					
4					
8					
16					



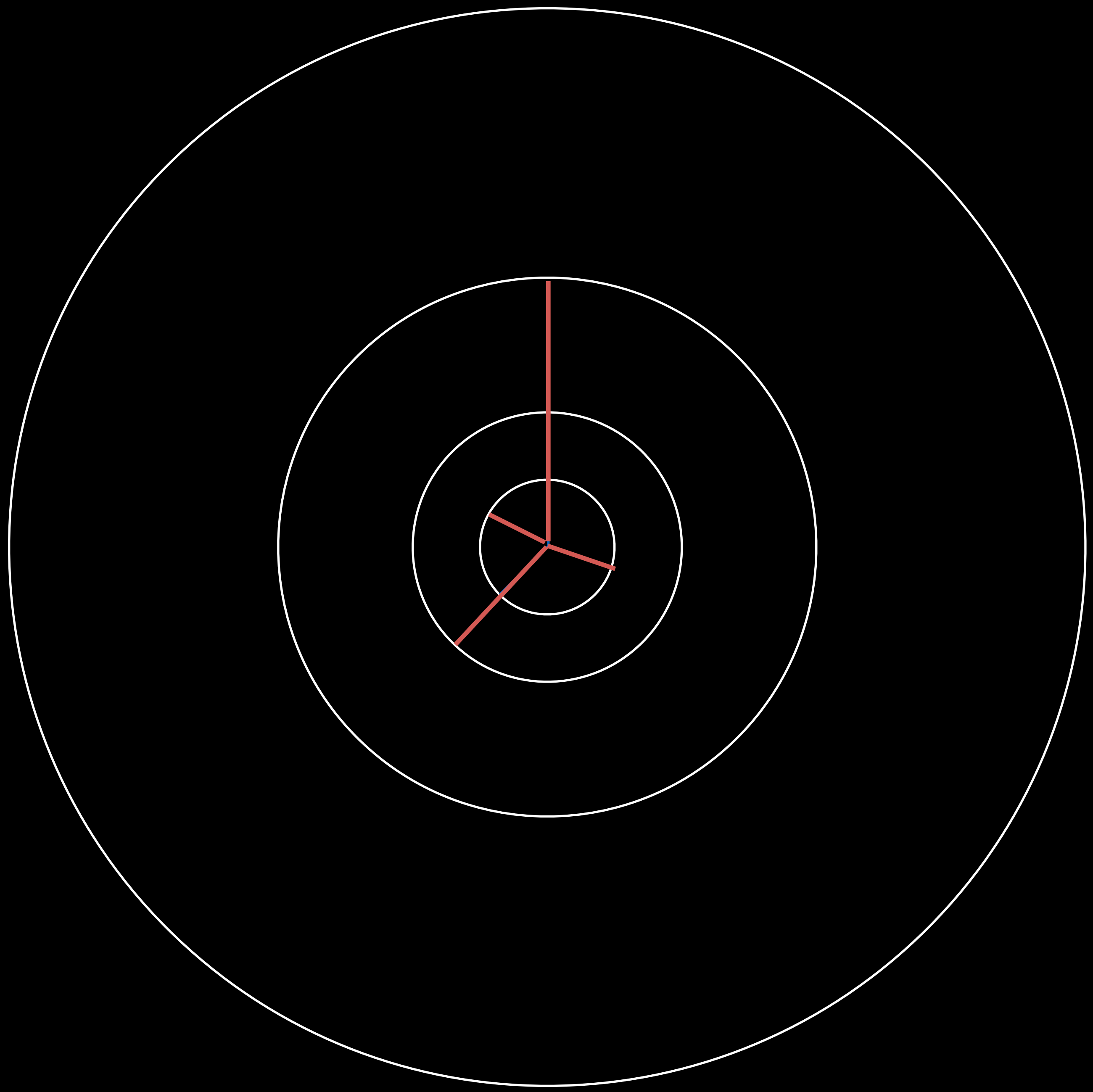
Spoke Length

# spokes	1	2	4	8	16
1					
2					
4					
8					
16					



Spoke Length

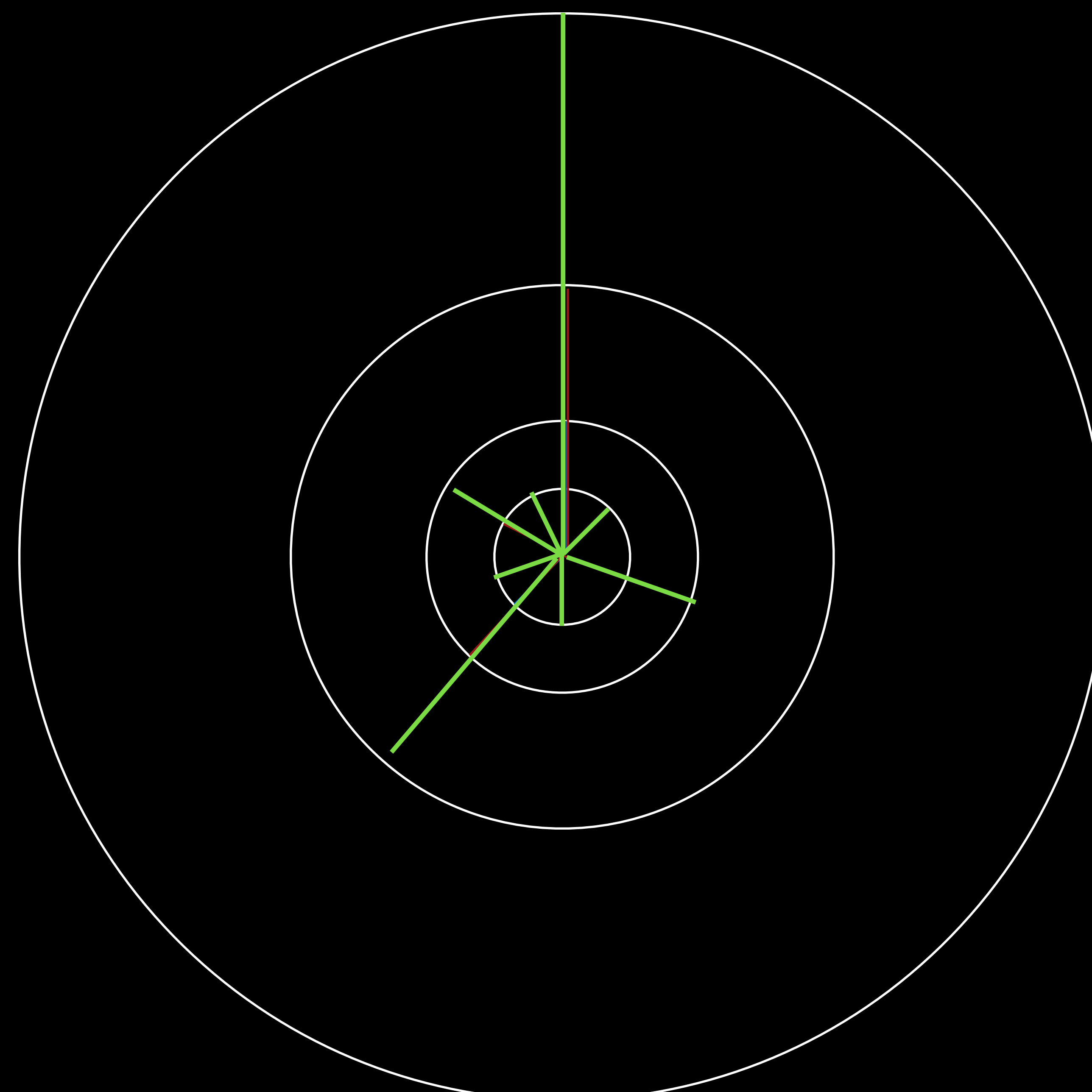
# spokes	1	2	4	8	16
1					
2					
4					
8					
16					



Spoke Length

spokes

	1	2	4	8	16
1					
2					
4					
8					
16					



Stream Problem

	1	2	4	8	16
1					
2					
4					
8					
16	○				

1) Target hidden in a cell

2) Algorithm chooses a stream of cells

3) Game ends when algorithm finds target

Total cost is sum of costs of cells searched

Cell (x,y) costs $x \cdot y$

Stream Problem

	1	2	4	8	16
1					
2					
4					
8					
16					

1) Target hidden in a cell

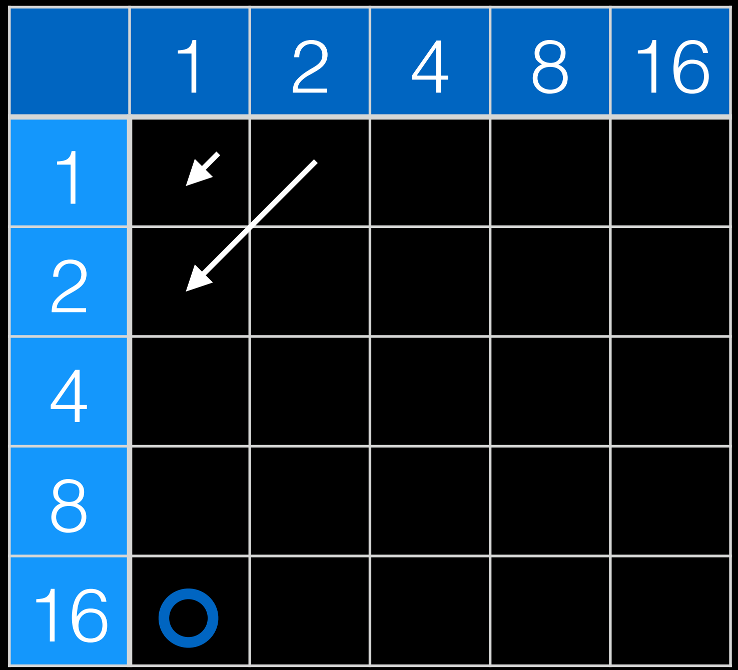
2) Algorithm chooses a stream of cells

3) Game ends when algorithm finds target

Total cost is sum of costs of cells searched

Cell (x,y) costs $x \cdot y$

Stream Problem



1) Target hidden in a cell

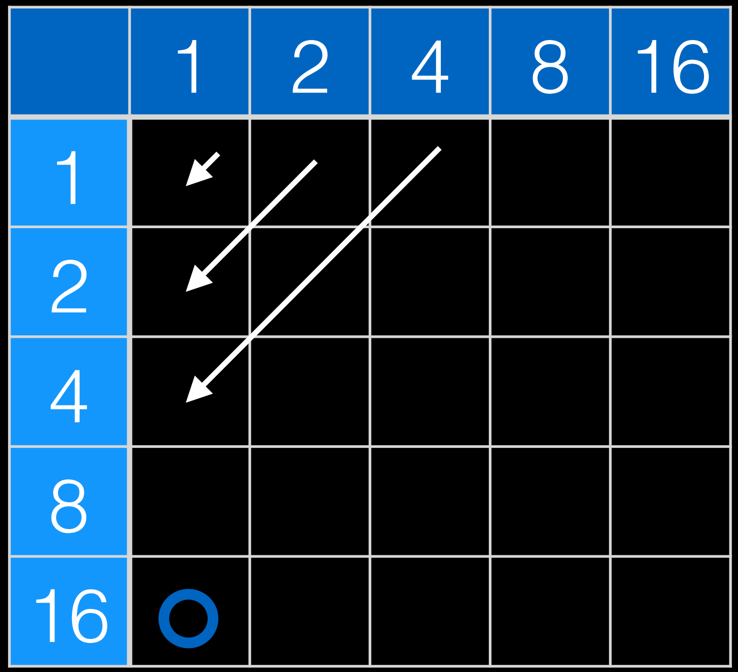
2) Algorithm chooses a stream of cells

3) Game ends when algorithm finds target

Total cost is sum of costs of cells searched

Cell (x,y) costs $x \cdot y$

Stream Problem



1) Target hidden in a cell

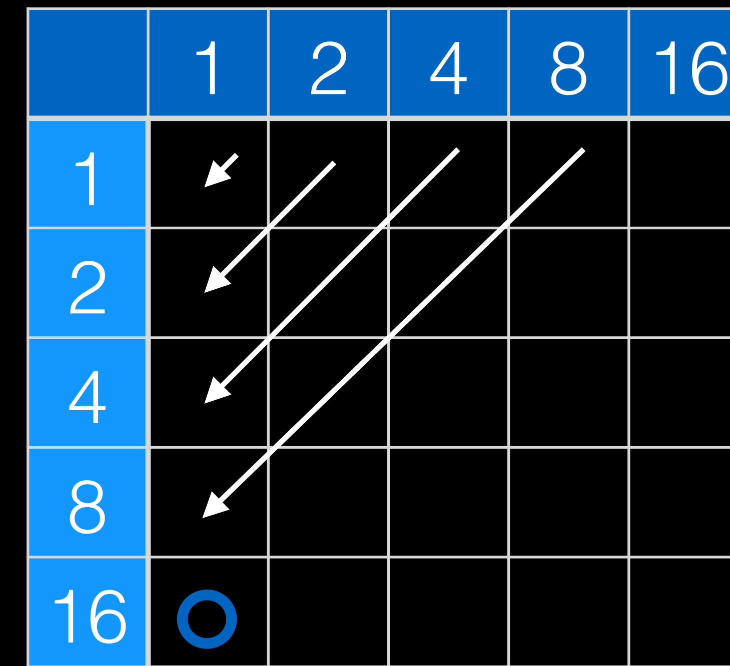
2) Algorithm chooses a stream of cells

3) Game ends when algorithm finds target

Total cost is sum of costs of cells searched

Cell (x,y) costs $x \cdot y$

Stream Problem



1) Target hidden in a cell

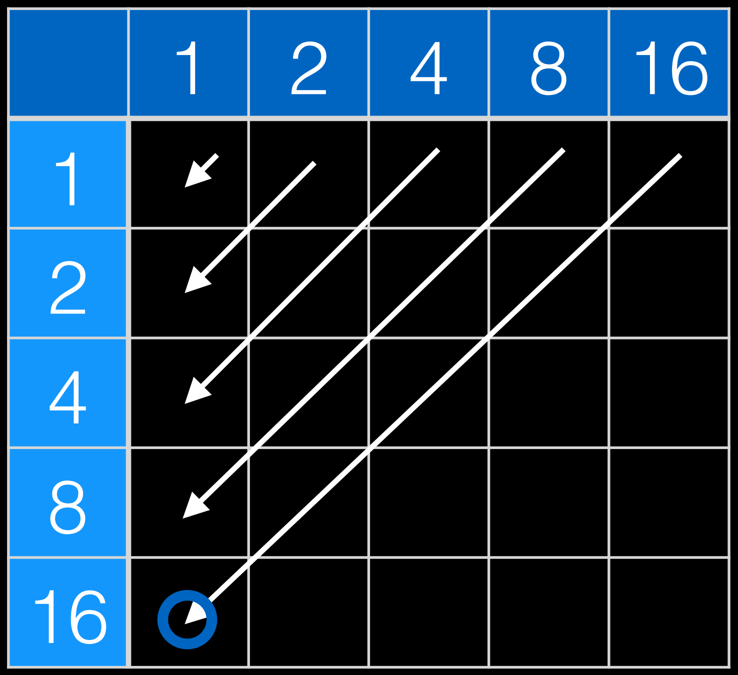
2) Algorithm chooses a stream of cells

3) Game ends when algorithm finds target

Total cost is sum of costs of cells searched

Cell (x,y) costs $x \cdot y$

Stream Problem



1) Target hidden in a cell

2) Algorithm chooses a stream of cells

3) Game ends when algorithm finds target

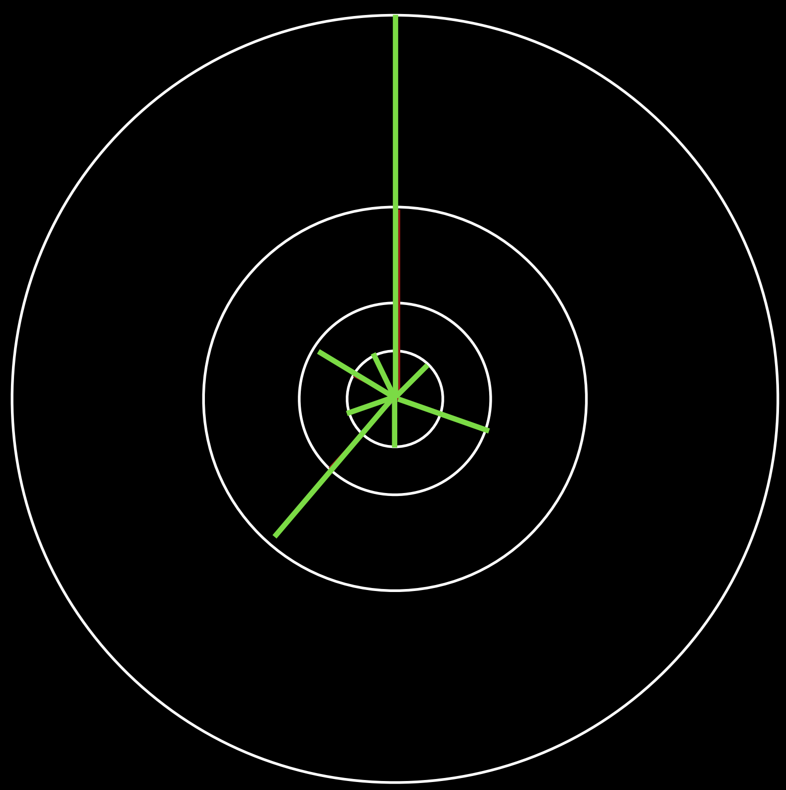
Total cost is sum of costs of cells searched

Cell (x,y) costs $x \cdot y$

GoldenFA

For epoch i = 1 to ∞,
  For each $1 \le x \le i$,
    Make 2^{i-x} spokes of length 2^x,
    rotated by ϕ

	1	2	4	8	16
1					
2					
4					
8					
16					



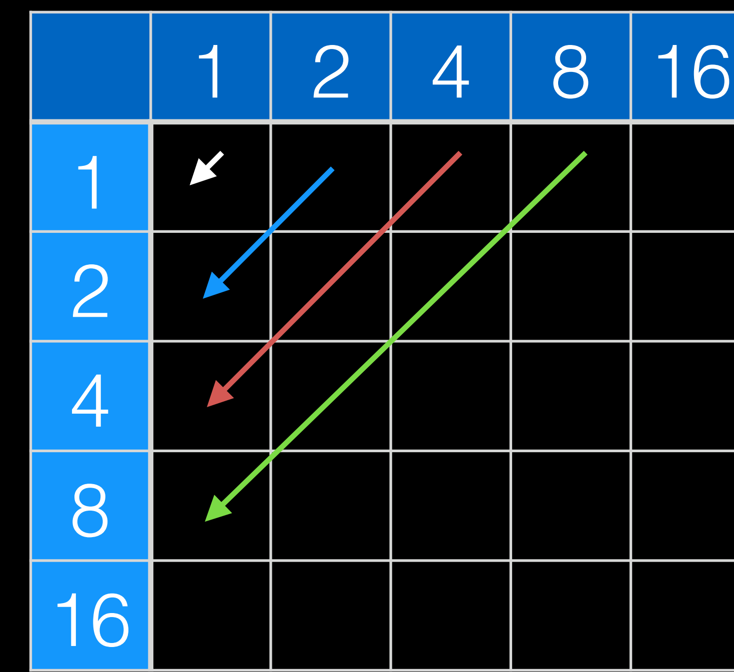
GoldenFA

For epoch i = 1 to ∞,

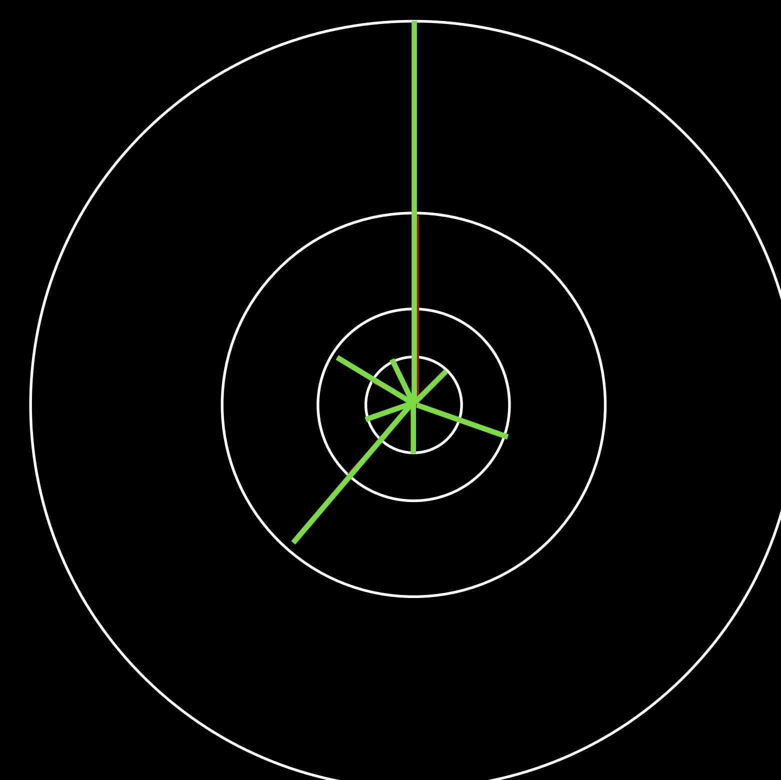
  For each $1 \le x \le i$,

    Make 2^{i-x} spokes of length 2^x,

    rotated by ϕ



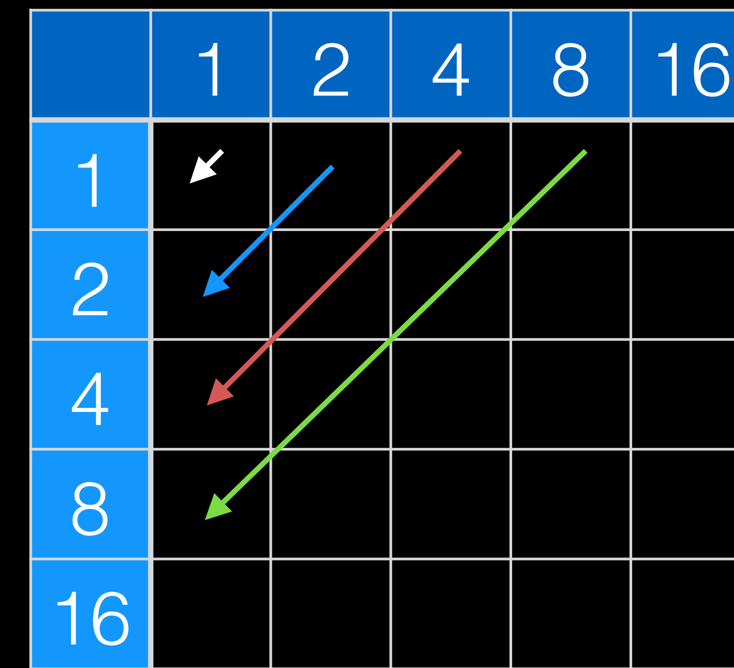
Number of epochs before reaching distance L: O(log L)

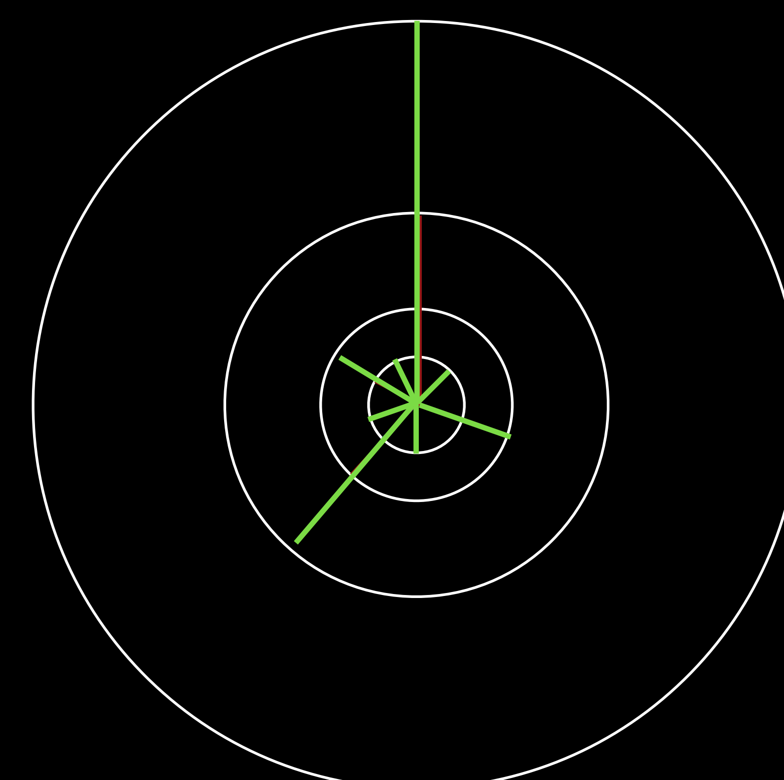


GoldenFA

For epoch i = 1 to ∞,
  For each $1 \le x \le i$,
    Make 2^{i-x} spokes of length 2^{x},
    rotated by ϕ

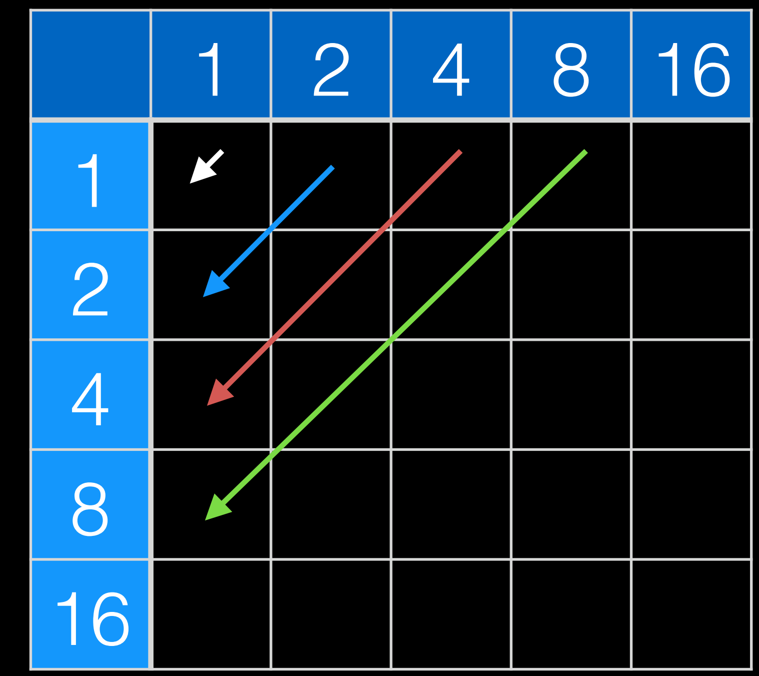
Number of epochs before reaching distance L: O(log L)





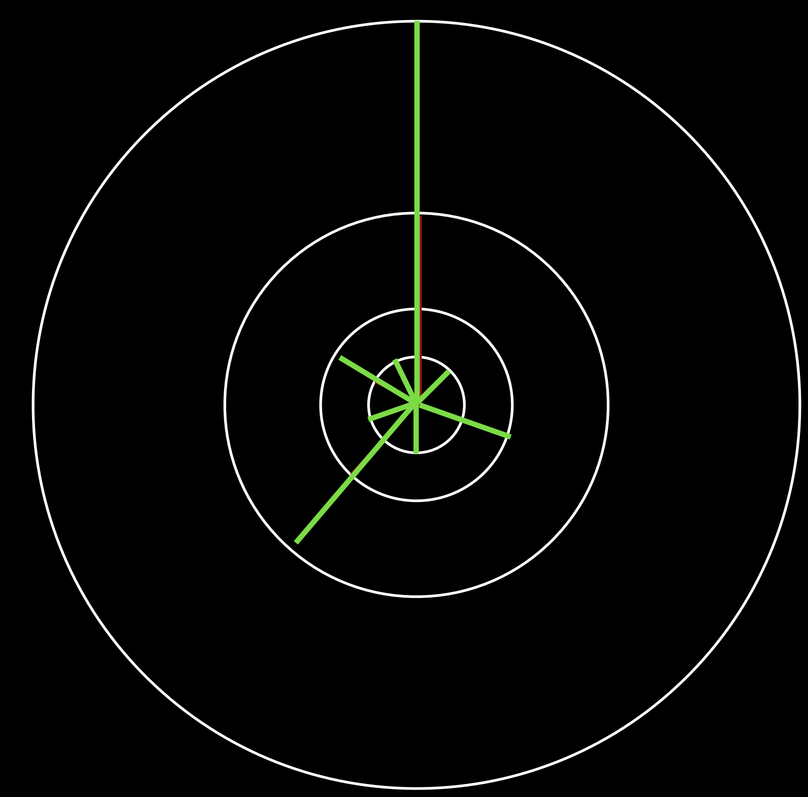
GoldenFA

For epoch i = 1 to ∞,
  For each $1 \le x \le i$,
   Make 2^{i-x} spokes of length 2^x,
   rotated by ϕ



Number of epochs before reaching distance L: O(log L)

Number of epochs before the spokes are sufficiently close: O(log (L/W))



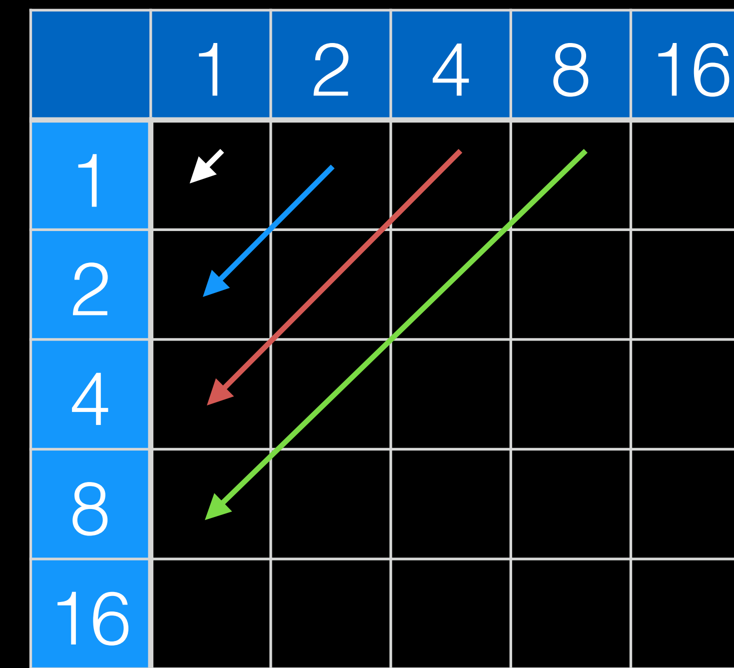
GoldenFA

For epoch i = 1 to ∞,

For each $1 \leq x \leq i$,

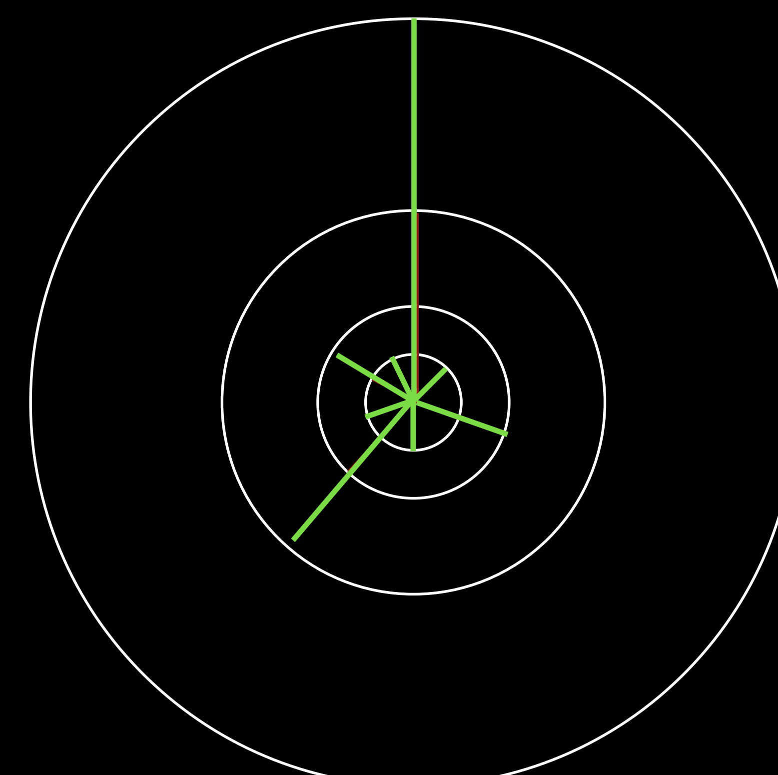
Make 2^{i-x} spokes of length 2^x,

rotated by ϕ



Number of epochs before reaching distance L: O(log L)

Number of epochs before the spokes are sufficiently close: O(log (L/W))



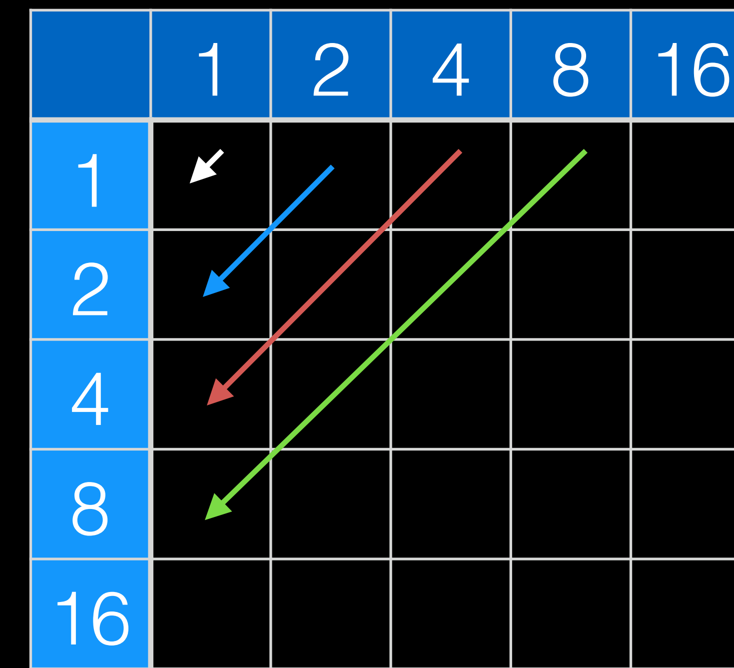
GoldenFA

For epoch i = 1 to ∞,

For each $1 \le x \le i$,

Make 2^{i-x} spokes of length 2^x,

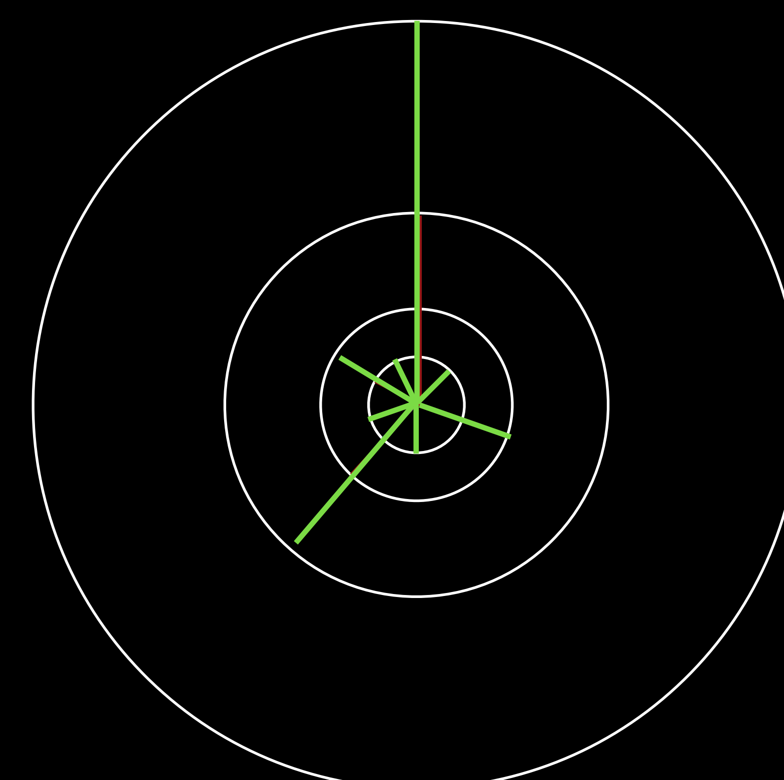
rotated by ϕ



Number of epochs before reaching distance L: O(log L)

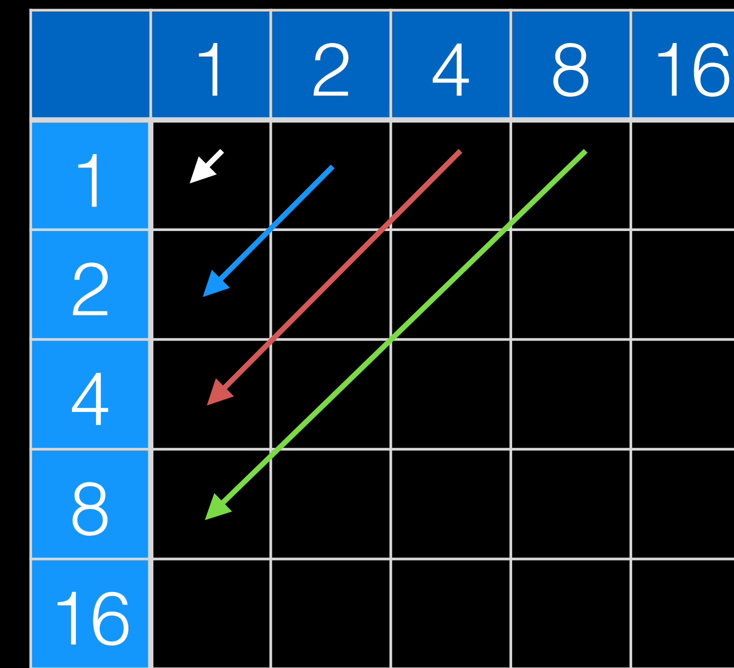
Number of epochs before the spokes are sufficiently close: O(log (L/W))

Cost for epoch i: $2^i \cdot i$



GoldenFA

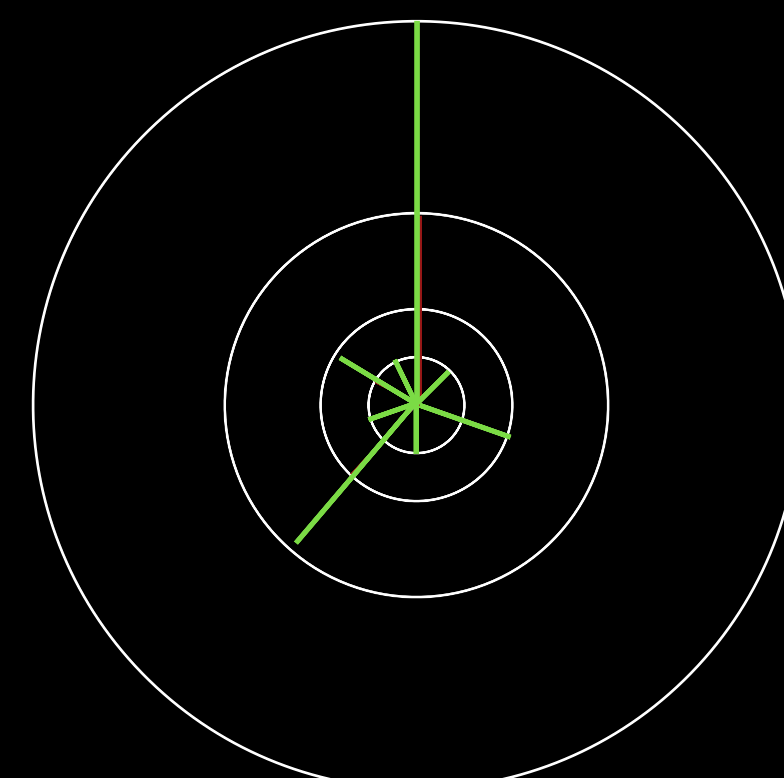
For epoch i = 1 to ∞,
  For each $1 \leq x \leq i$,
    Make 2^{i-x} spokes of length 2^x,
    rotated by ϕ



Number of epochs before reaching distance L: O(log L)

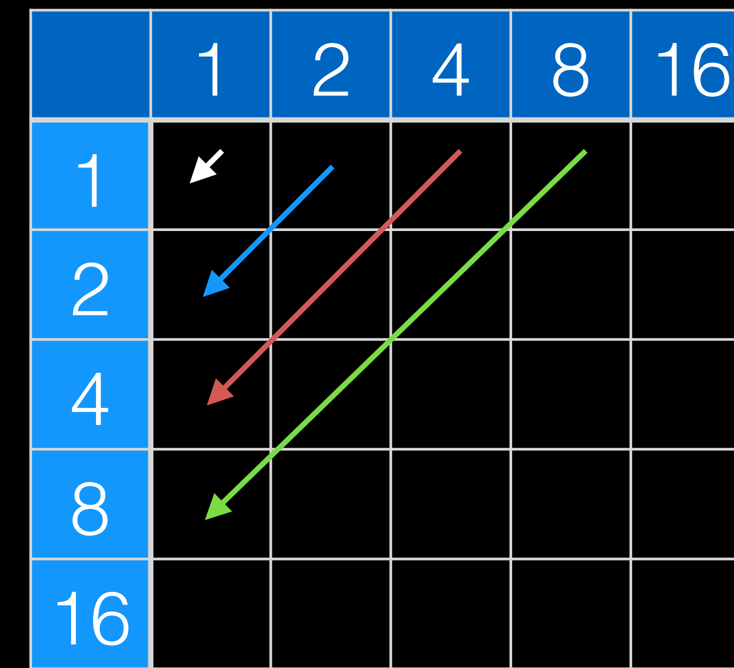
Number of epochs before the spokes are sufficiently close: O(log (L/W))

Cost for epoch i: $2^i \cdot i$



GoldenFA

For epoch i = 1 to ∞,
  For each $1 \leq x \leq i$,
    Make 2^{i-x} spokes of length 2^x,
    rotated by ϕ

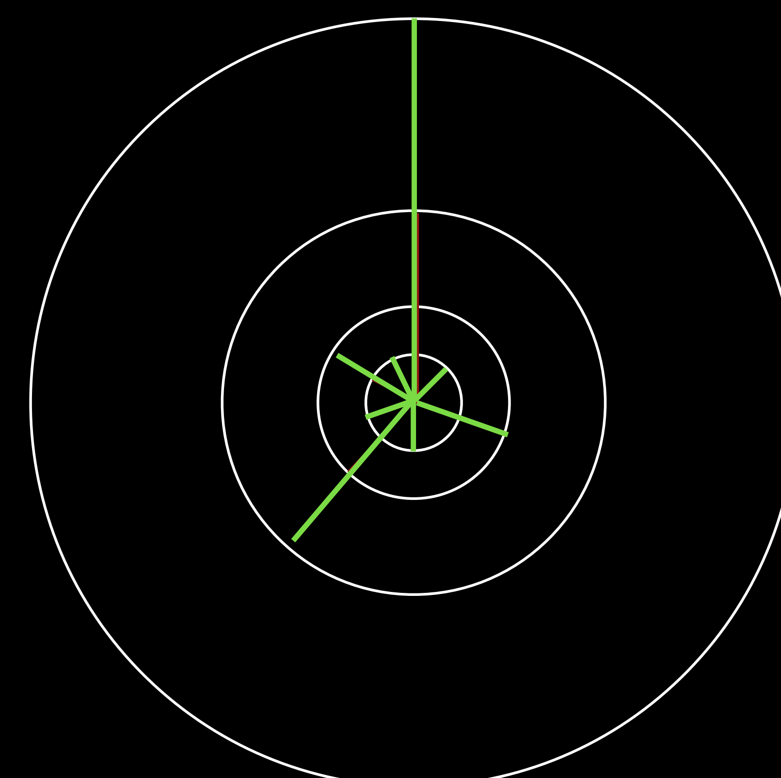


Number of epochs before reaching distance L: O(log L)

Number of epochs before the spokes are sufficiently close: O(log (L/W))

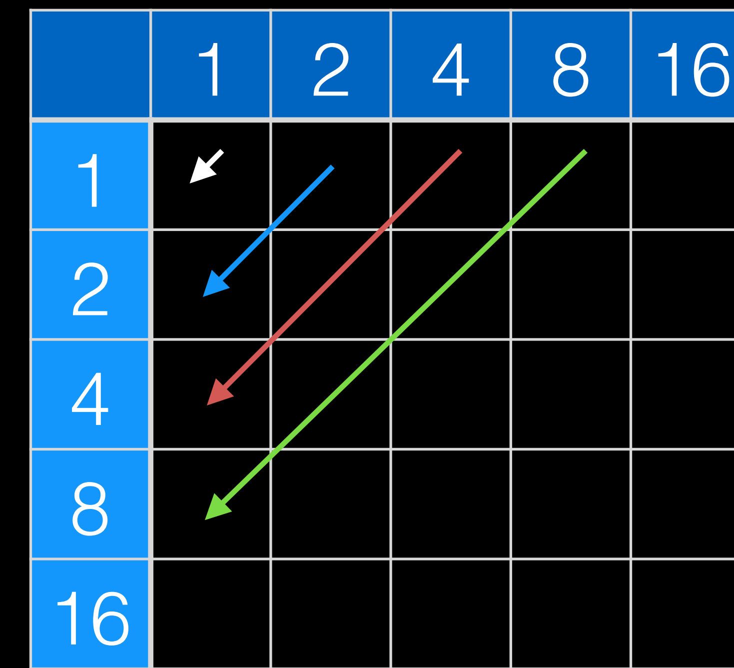
Cost for epoch i: $2^i \cdot i$

Total cost dominated by last epoch:

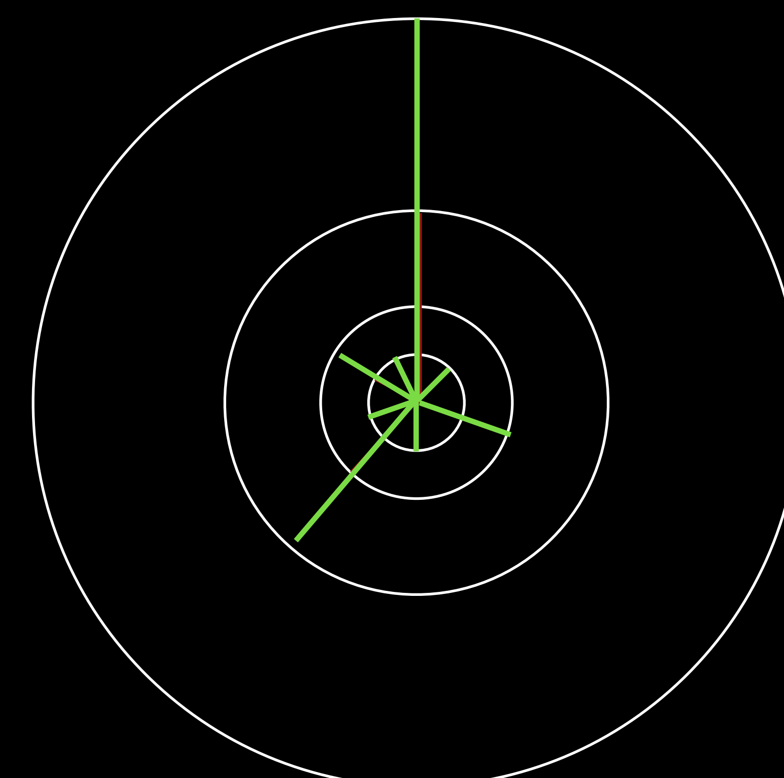


GoldenFA

For epoch i = 1 to ∞,
  For each $1 \leq x \leq i$,
    Make 2^{i-x} spokes of length 2^x,
    rotated by ϕ



Number of epochs before reaching distance L: O(log L)



Number of epochs before the spokes are sufficiently close: O(log (L/W))

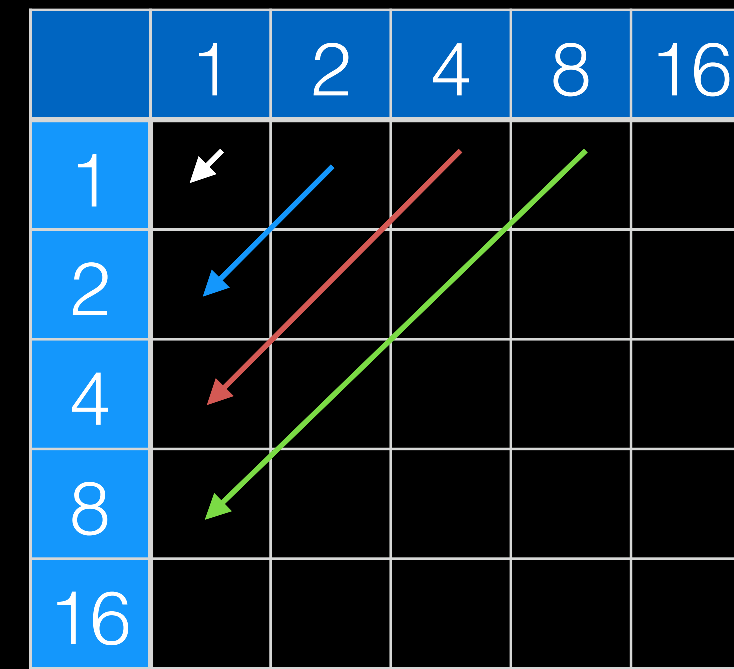
Cost for epoch i: $2^i \cdot i$

Total cost dominated by last epoch:

$$O\left(\frac{L^2}{W}\log\frac{L^2}{W}\right) = O\left(\frac{L^2}{W}\log L\right)$$

GoldenFA

For epoch i = 1 to ∞,
  For each $1 \leq x \leq i$,
    Make 2^{i-x} spokes of length 2^x,
    rotated by ϕ



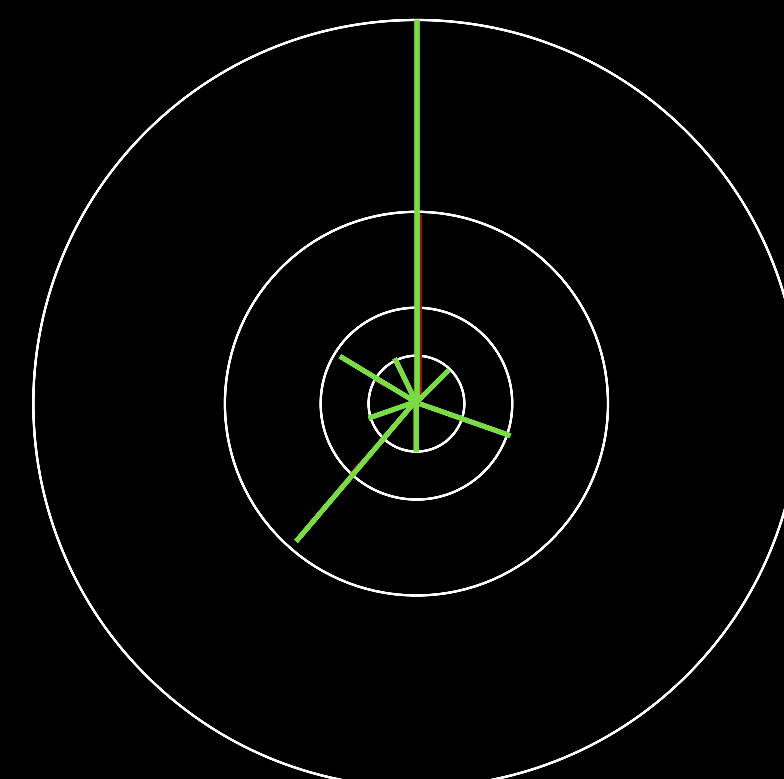
Number of epochs before reaching distance L: O(log L)

Number of epochs before the spokes are sufficiently close: O(log (L/W))

Cost for epoch i: $2^i \cdot i$

Total cost dominated by last epoch:

$$O\left(\frac{L^2}{W}\log\frac{L^2}{W}\right) = O\left(\frac{L^2}{W}\log L\right)$$



Multiple Searchers?

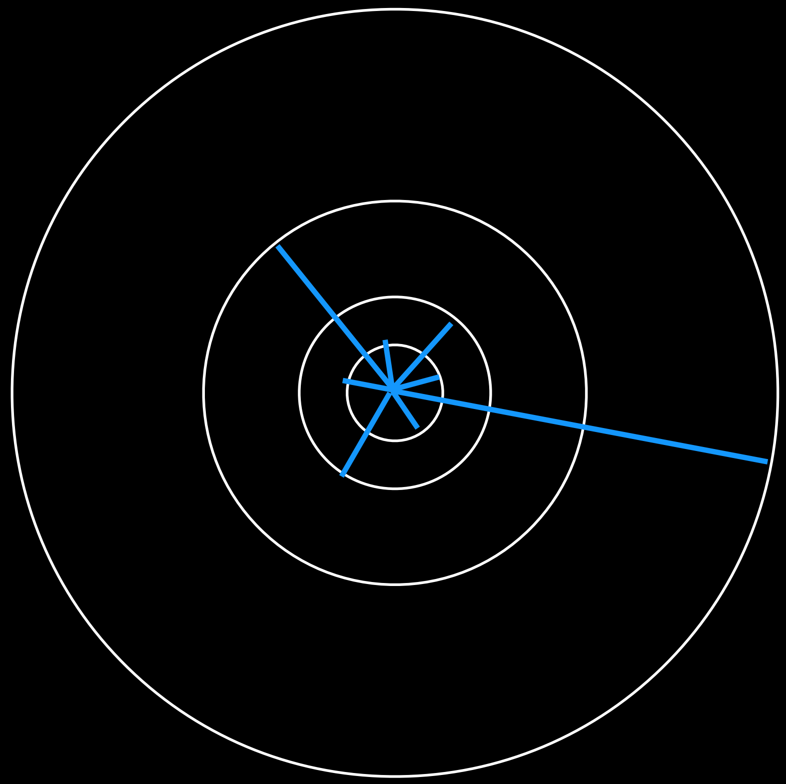
Multiple Searchers?

Random Initial Orientation!

N Agents

Each agent chooses a random initial heading
  In epoch i = 1 to ∞,
    For each $1 \le x \le i$,
      Make 2^{i-x} spokes of length 2^x, rotated by ϕ

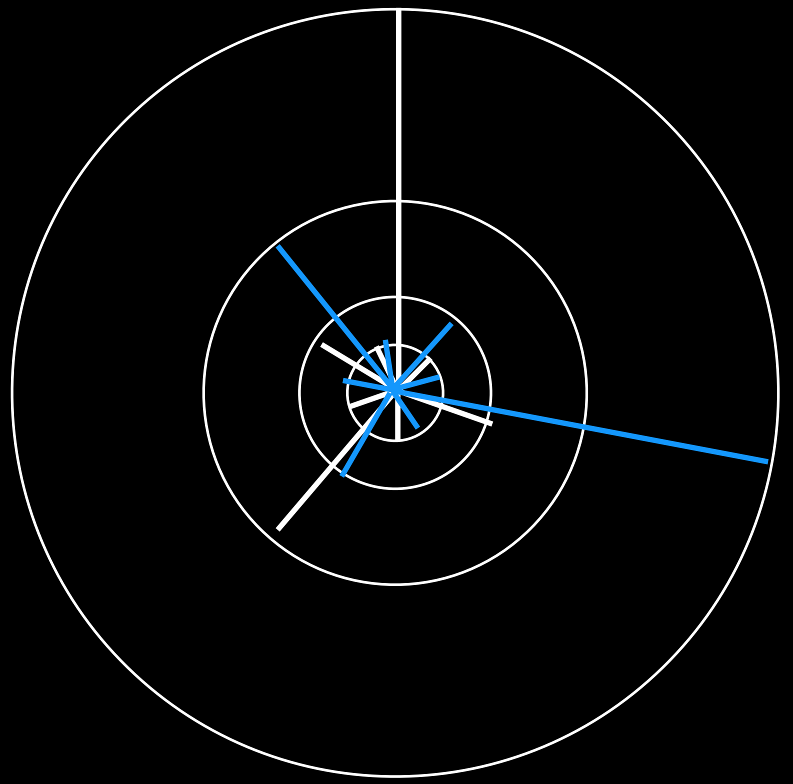
N = 3



N Agents

Each agent chooses a random initial heading
  In epoch i = 1 to ∞,
    For each $1 \leq x \leq i$,
      Make 2^{i-x} spokes of length 2^x, rotated by ϕ

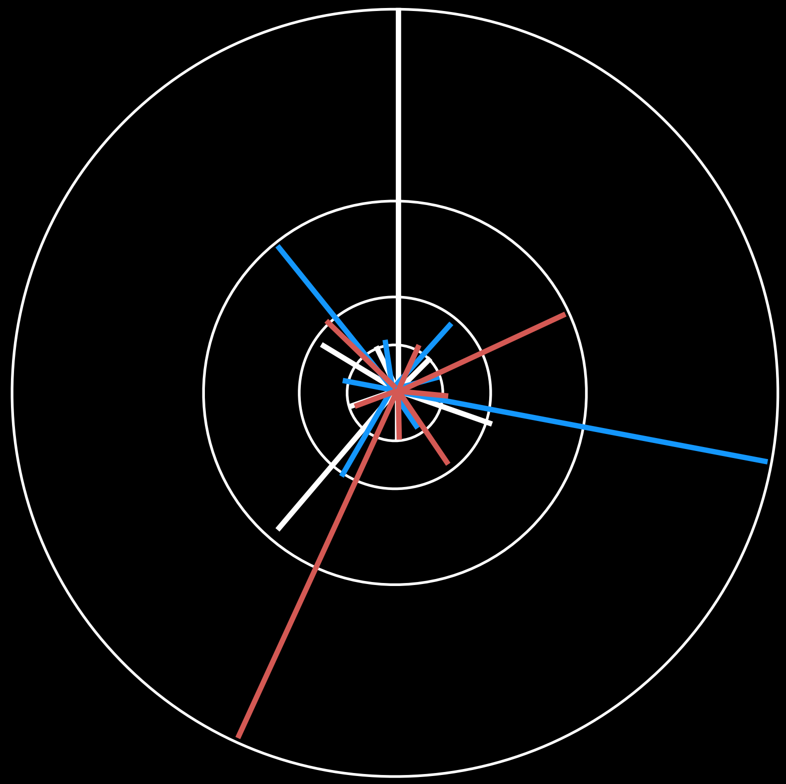
N = 3



N Agents

Each agent chooses a random initial heading
  In epoch i = 1 to ∞,
    For each $1 \leq x \leq i$,
      Make 2^{i-x} spokes of length 2^x, rotated by ϕ

N = 3



N Agents

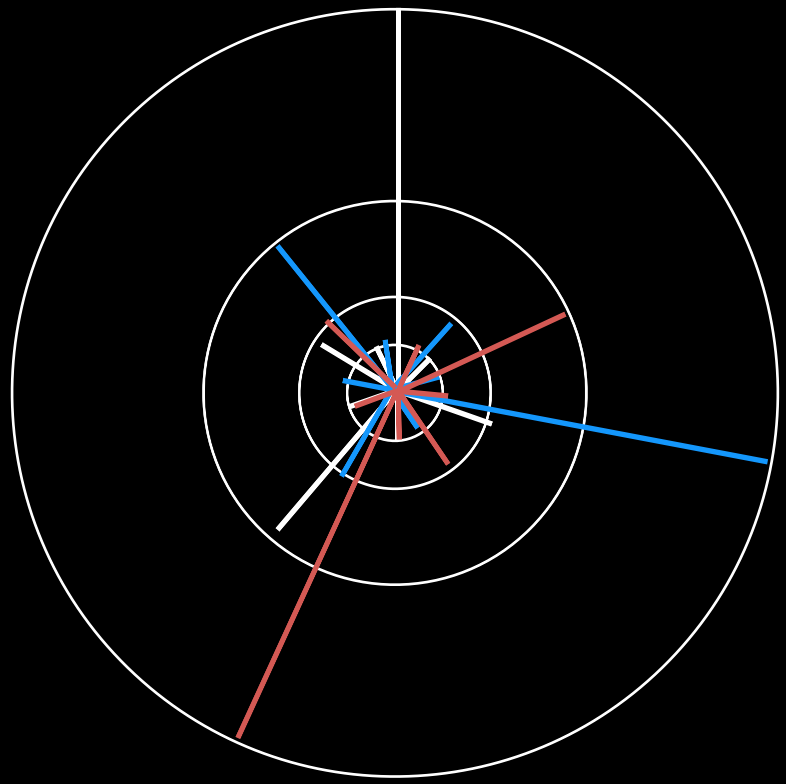
Each agent chooses a random initial heading

In epoch i = 1 to ∞,

For each $1 \leq x \leq i$,

Make 2^{i-x} spokes of length 2^x, rotated by ϕ

N = 3



$t < N$ faults

Expected search time:

$$O\left(\left(L + \frac{L^2(t+1)}{NW}\right)\log L\right)$$

N Agents; $t < N$ faults

N Agents; $t < N$ faults

Expected search time:

$$O\left(\left(L + \frac{L^2(t+1)}{NW}\right)\log L\right)$$

N Agents; $t < N$ faults

Expected search time:

$$O\left(\left(L + \frac{L^2(t+1)}{NW}\right)\log L\right)$$

Lower bound on expected search time for "spoke-based":

$$\Omega\left(L + \left(\frac{L^2(t+1)}{NW}\right)\log L\right)$$

"Spoke-based": All search via line segments from nest

N Agents; $t < N$ faults

Compute expected # agents finding target in each epoch

Expected search time:

$$O\left(\left(L + \frac{L^2(t+1)}{NW}\right)\log L\right)$$

Lower bound on expected search time for "spoke-based":

$$\Omega\left(L + \left(\frac{L^2(t+1)}{NW}\right)\log L\right)$$

"Spoke-based": All search via line segments from nest

N Agents; $t < N$ faults

Expected search time:

$$O\left(\left(L + \frac{L^2(t+1)}{NW}\right)\log L\right)$$

Compute expected # agents finding target in each epoch

Use this expectation to bound probability $\leq t$ agents find target

Lower bound on expected search time for “spoke-based”:

$$\Omega\left(L + \left(\frac{L^2(t+1)}{NW}\right)\log L\right)$$

“Spoke-based”: All search via line segments from nest

N Agents; $t < N$ faults

Expected search time:

$$O\left(\left(L + \frac{L^2(t+1)}{NW}\right)\log L\right)$$

Compute expected # agents finding target in each epoch

Use this expectation to bound probability $\leq t$ agents find target

Lower bound on expected search time for "spoke-based":

$$\Omega\left(L + \left(\frac{L^2(t+1)}{NW}\right)\log L\right)$$

Yao's Lemma
on Stream Problem

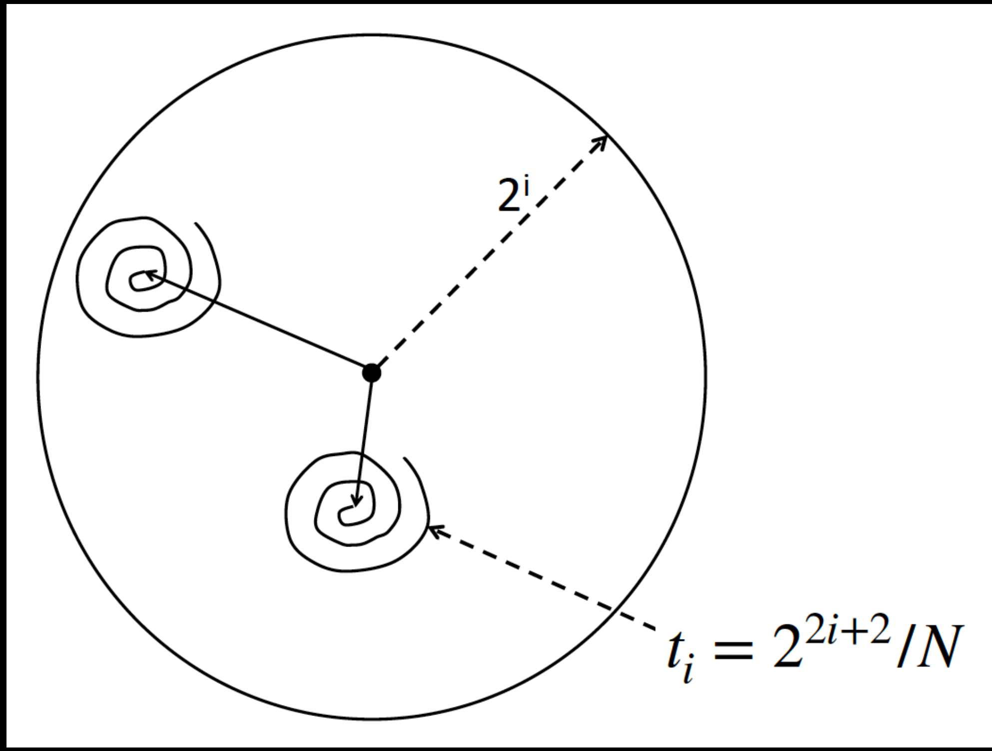
"Spoke-based": All search via line segments from nest

Experiments

F&K Advice

Each agent does the following:

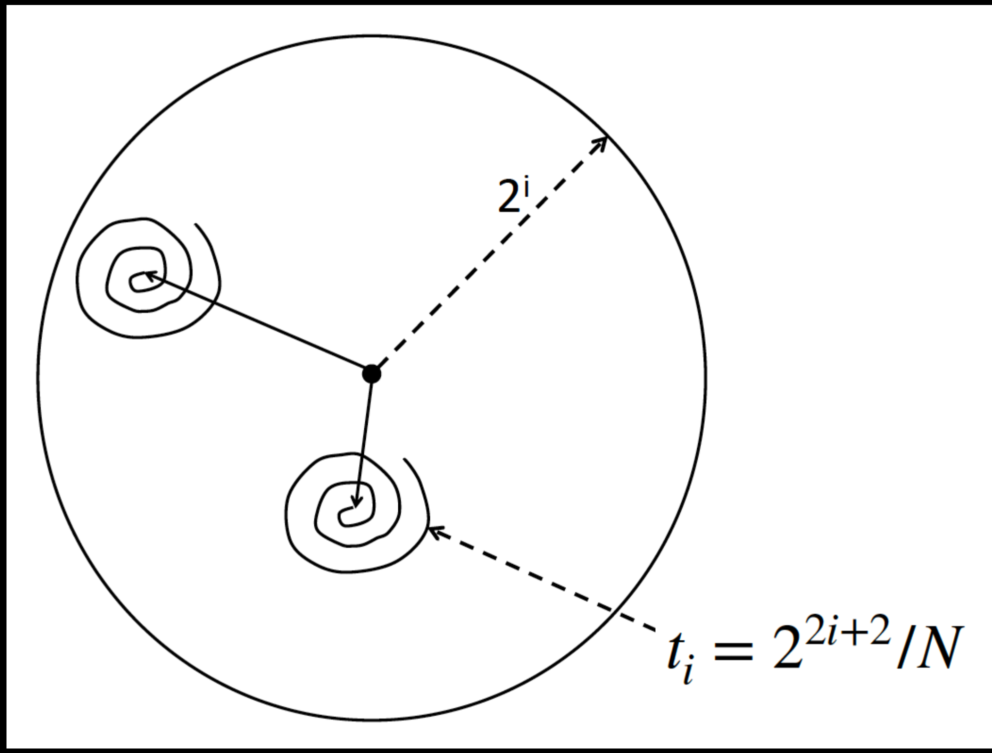
For stage j = 1 to ∞
    For phase i = 1 to j
        Go to a random point at distance $\leq 2^i$
        Spiral search for time $2^{2i+2}/N$
        Return to nest



F&K Advice

Each agent does the following:

For stage j = 1 to ∞
    For phase i = 1 to j
        Go to a random point at distance $\leq 2^i$
        Spiral search for time $2^{2i+2}/N$
        Return to nest



$\log N$ bits of advice to know N

$\log \log N$ bits of advice to know 2-approximation to N

F&K NoAdvice (fix $\epsilon > 0$)

Each agent does the following

For epoch $\ell = 0$ to ∞

For stage i = 0 to ℓ

For phase j = 0 to i

Go to a random point at distance $\leq \sqrt{\dfrac{2^{i+j}}{\lceil \log^{1+\epsilon} 2^j \rceil}}$

Spiral search for time $\dfrac{2^{2i+2}}{\lceil \log^{1+\epsilon} 2^j \rceil}$

Return to nest

Algorithms Tested

Algorithm	**Advice (bits)**	**Robustness**	**Runtime**
F&K (advice)	$O(\log\log N)$	Not Robust	$O\left(L + \frac{L^2}{N}\right)$ for $W = \Theta(1)$
F&K (no advice)	0	Not Robust	$O\left(\left(L + \frac{L^2}{N}\right)\log^{1+\varepsilon} N\right)$ for fixed $\varepsilon > 0$ and $W = \Theta(1)$
GoldenFA	0	$t < N$	$O\left(\left(L + \frac{L^2(t+1)}{NW}\right)\log L\right)$

Algorithms Tested

Algorithm	Advice (bits)	Robustness	Runtime
F&K (advice)	$O(\log\log N)$	Not Robust	$O\left(L + \frac{L^2}{N}\right)$ for $W = \Theta(1)$
F&K (no advice)	0	Not Robust	$O\left(\left(L + \frac{L^2}{N}\right)\log^{1+\varepsilon} N\right)$ for fixed $\varepsilon > 0$ and $W = \Theta(1)$
GoldenFA	0	$t < N$	$O\left(\left(L + \frac{L^2(t+1)}{NW}\right)\log L\right)$

GoldenFA-Heuristic:

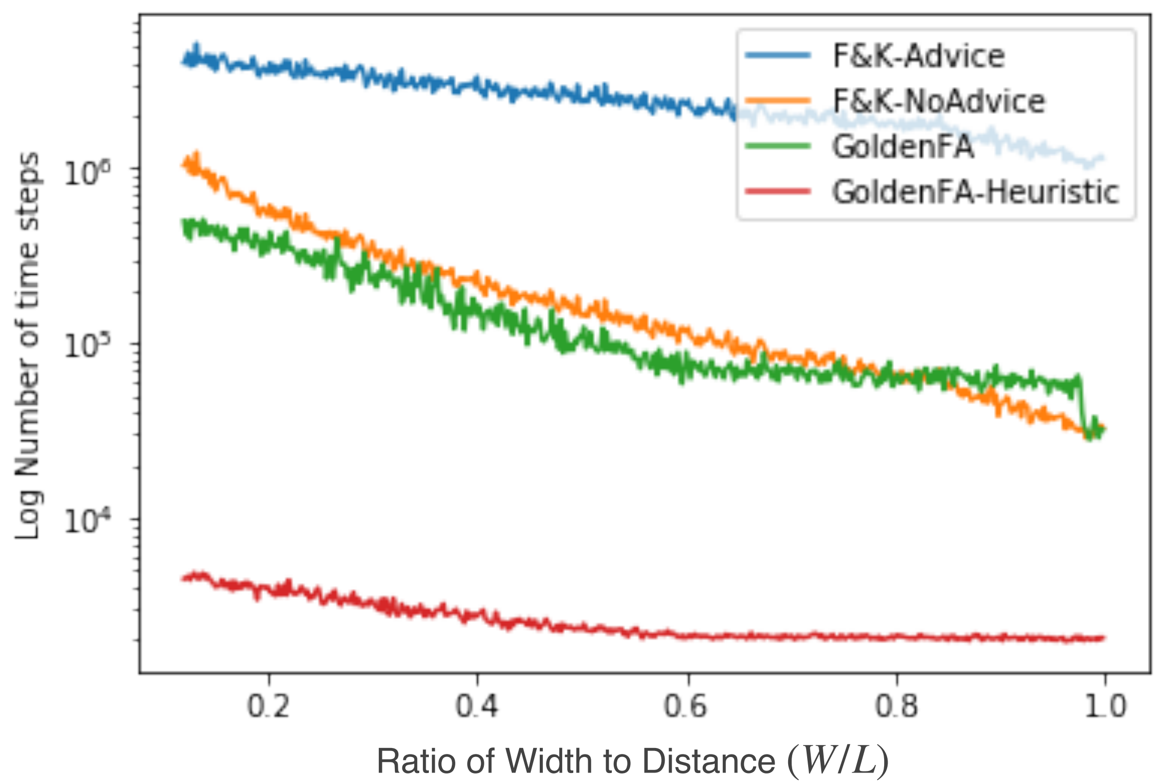
In epoch i, make $\lceil c(1+\alpha) \rceil$ spokes of length $(1+\alpha)^i$

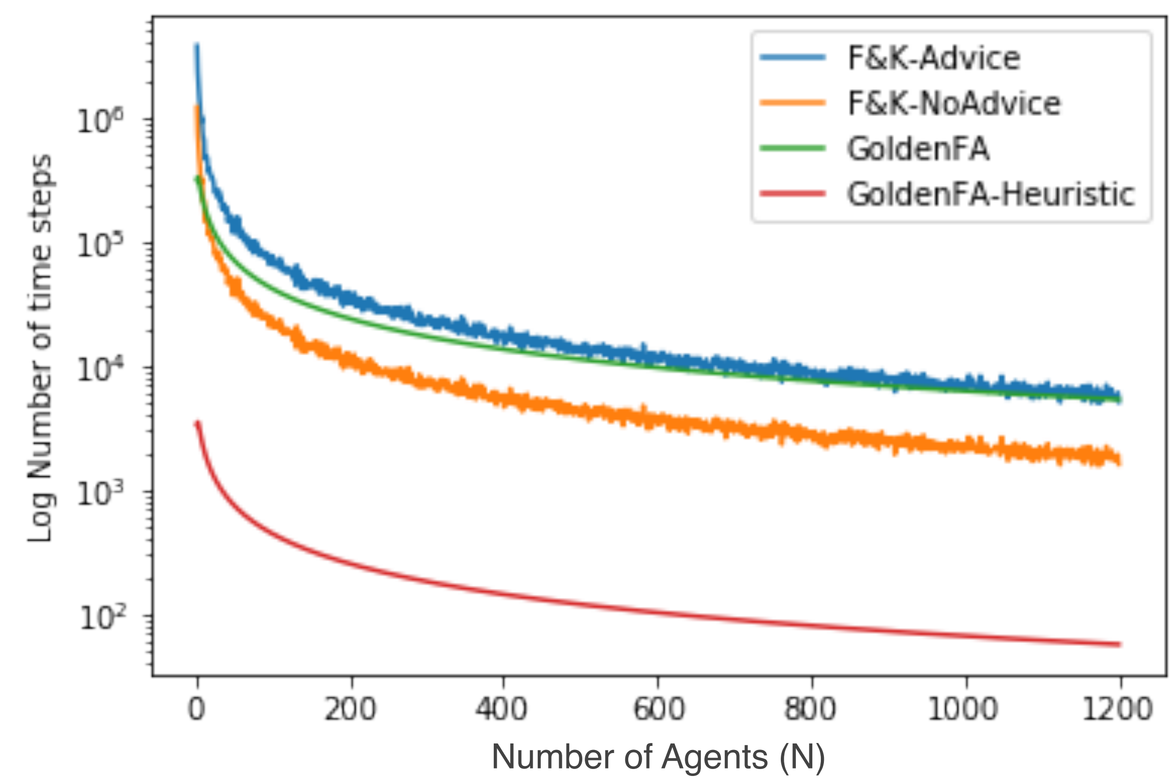
$c \leftarrow 1.9; \alpha \leftarrow 7$

F&K-NoAdvice:

$\epsilon \leftarrow .01$

Varying W; Varying N

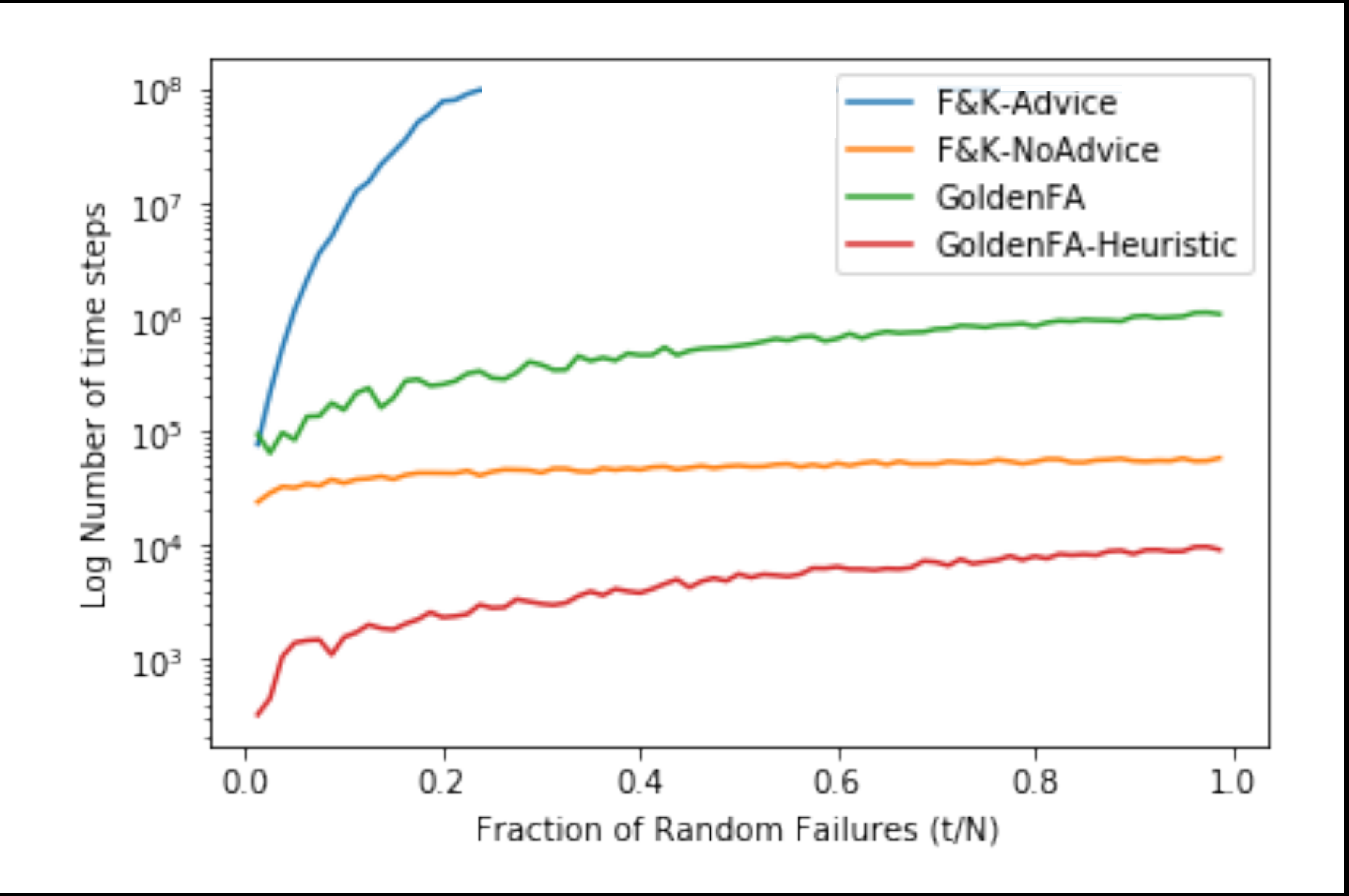




$L = 500; N = 1$

$L = 500; W = 4$

Faults



$L = 500; D = 4; N = 100$

Conclusion


THAT'S IT
I'VE HAD IT WITH THESE
%$@#!@$%!
ANTS
ON THIS
%$@#!@$%!
PLANE

Results Recap

L = target distance; W = target width;
N = # agents; t = # faults

Results Recap

L = target distance; W = target width;
N = # agents; t = # faults

Expected search time:

$$O\left(\left(L + \frac{L^2(t+1)}{NW}\right)\log L\right)$$

Results Recap

L = target distance; W = target width;
N = # agents; t = # faults

Expected search time:

$$O\left(\left(L + \frac{L^2(t+1)}{NW}\right)\log L\right)$$

Lower bound on expected search time for “spoke-based”:

$$\Omega\left(L + \left(\frac{L^2(t+1)}{NW}\right)\log L\right)$$

Future Work

Get the %$@#%! ANTS off the %$@#%! plane

Get the %$@#%! ANTS off the %$@#%! plane

Theoretical Problem: Search in $\mathbb{R}^3$

Get the %$@#%! ANTS off the %$@#%! plane

Theoretical Problem: Search in $\mathbb{R}^3$

Practical Problem: Many searches have properties that simplify search along third dimension

Target Density

Target Density

Assume: Agent can sense local target density

Target Density

Assume: Agent can sense local target density

General Problem: Order statistics

Target Density

Assume: Agent can sense local target density

General Problem: Order statistics

Problem 1: Efficiently estimate target mass

Target Density

Assume: Agent can sense local target density

General Problem: Order statistics

Problem 1: Efficiently estimate target mass

Problem 2: Find max target density

Target Density

Assume: Agent can sense local target density

General Problem: Order statistics

Problem 1: Efficiently estimate target mass

Problem 2: Find max target density

Gradient Descent

Questions?