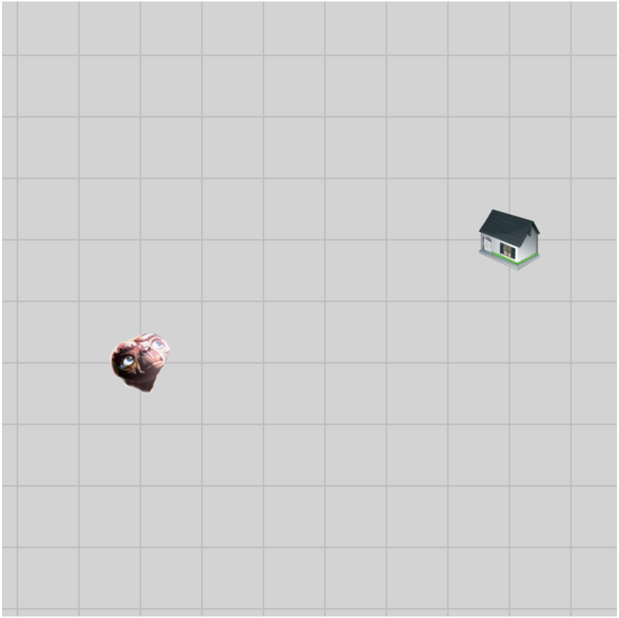
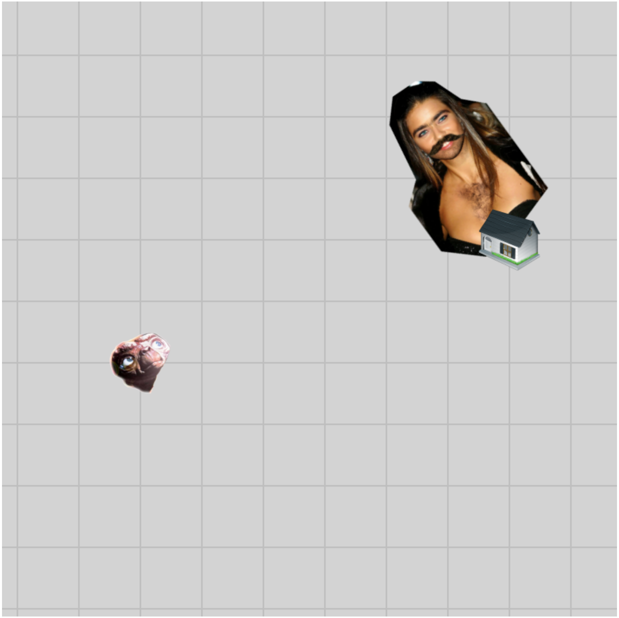
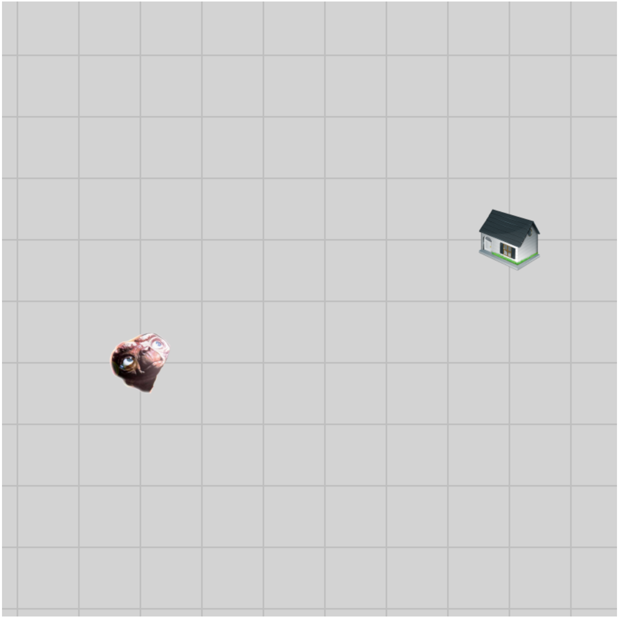
Optimal strategies for maintaining a chain of relays between an explorer and a base camp

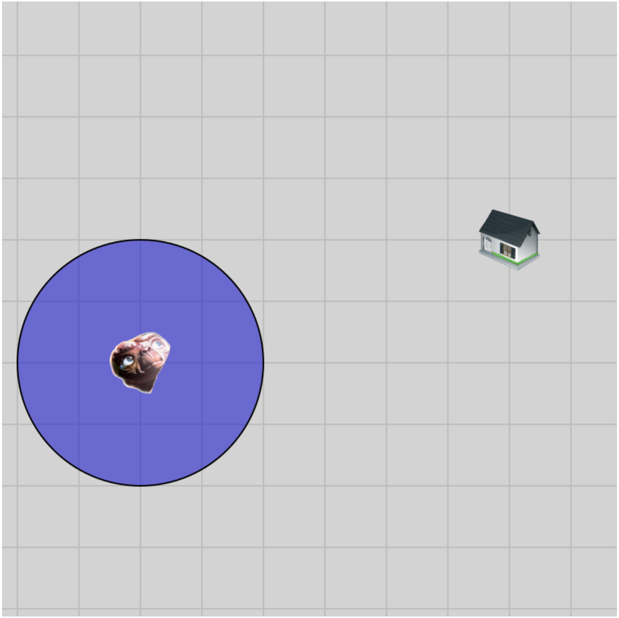
Lukas Humbel

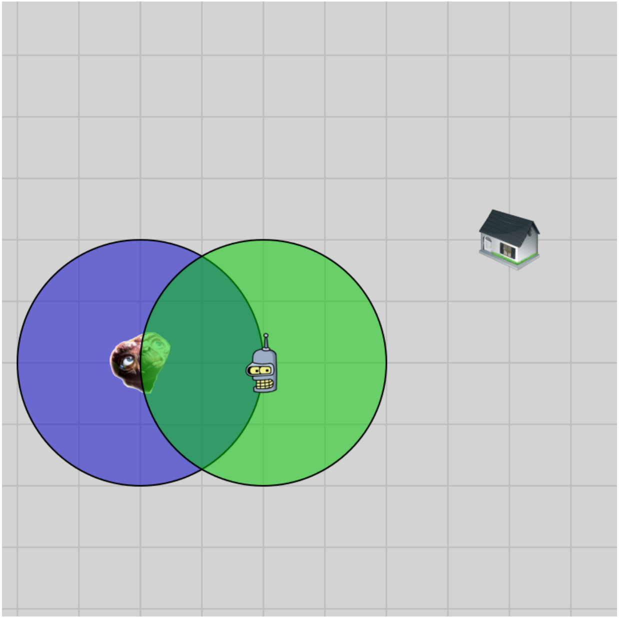
2. Mai 2012

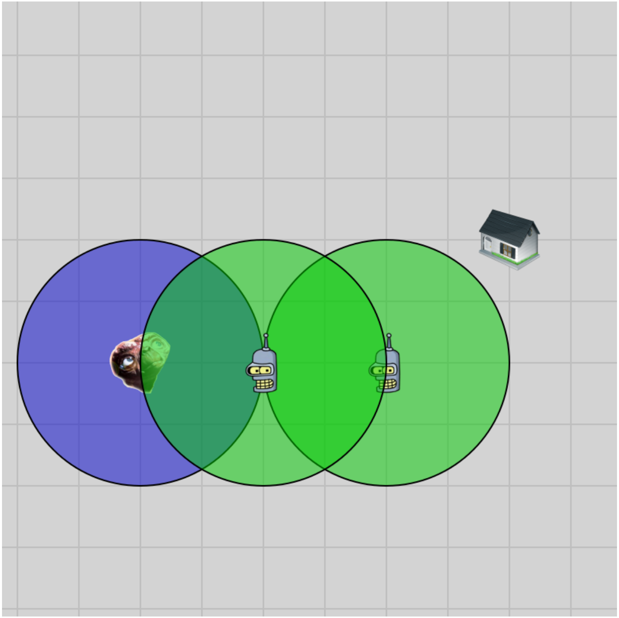


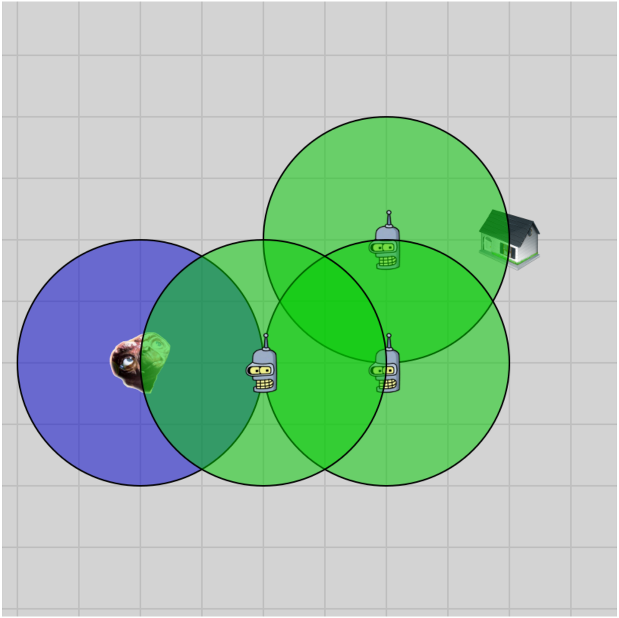


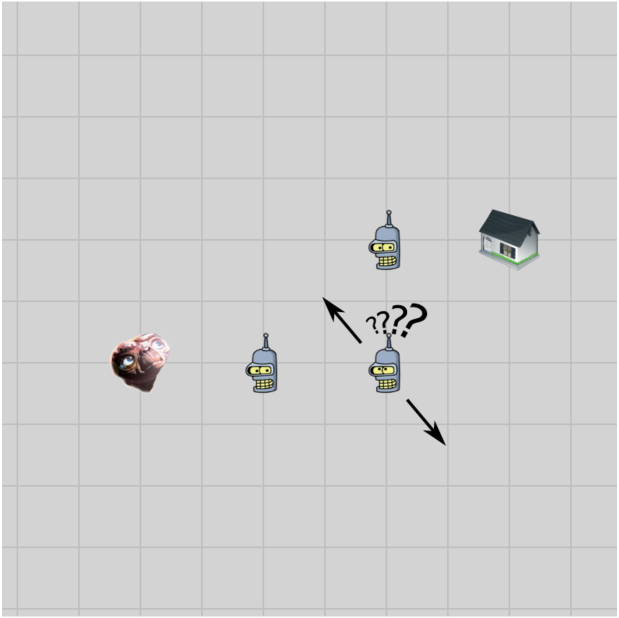










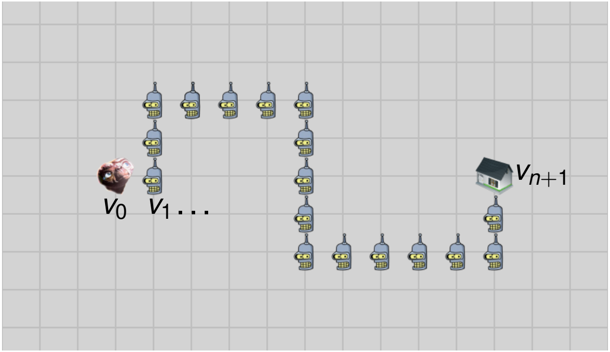


Outline

1. Model Definition
   - Problem Statement
   - Time/Relay Model
   - What to measure
2. Manhattan Hopper Strategy
   - Strategy Description
   - Static Scenario Performance
   - Dynamic Scenario Performance
3. Conclusion

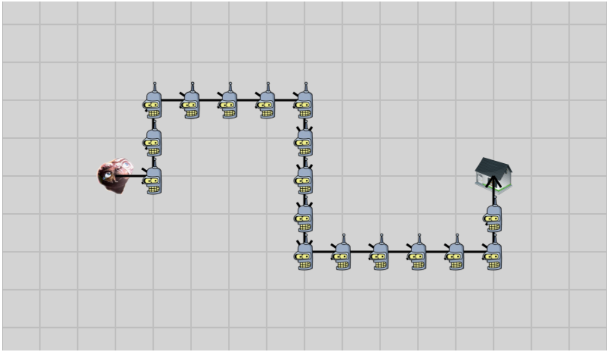
Problem Statement

Problem Statement



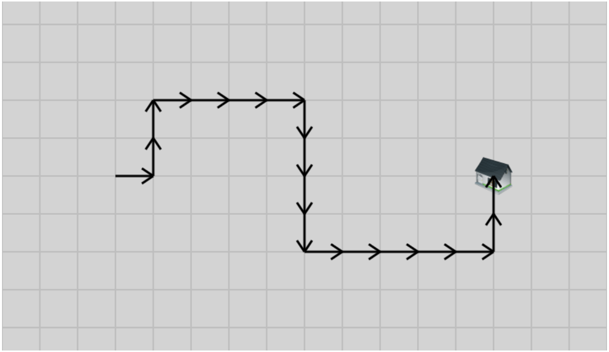
- Grid size: 0.5
- Transmission distance: 1

Problem Statement



- Grid size: 0.5
- Transmission distance: 1

Problem Statement



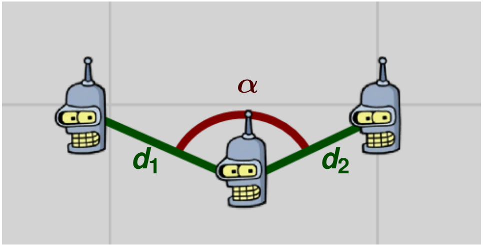
- Grid size: 0.5
- Transmission distance: 1

Time/Relay Model

Time Model

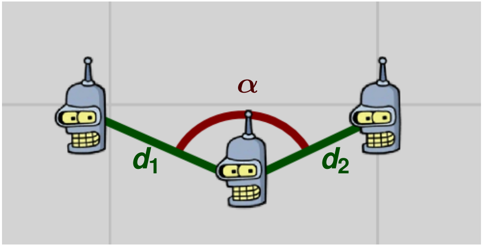
- Synchronized
- Look – Compute – Move

Relay Model - Sensory Input



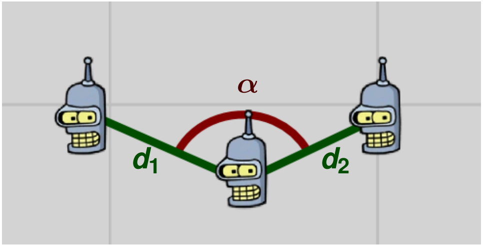
- Sees its chain neighbors
- Memoryless
- No communication

Relay Model - Sensory Input



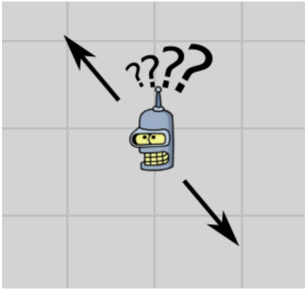
- Sees its chain neighbors
- Memoryless
- No communication

Relay Model - Sensory Input



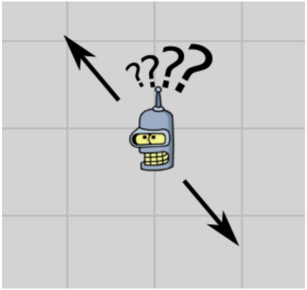
- Sees its chain neighbors
- Memoryless
- No communication
- . . . must sense when predecessor has stepped

Relay Model - Movement



- Moves with constant speed

Relay Model - Movement



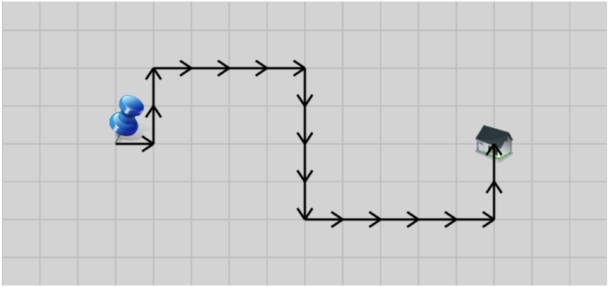
- Moves with constant speed
- Can be removed everywhere
- Inserted only at home

Chain Attributes

- *Valid* condition
- *Optimal* condition

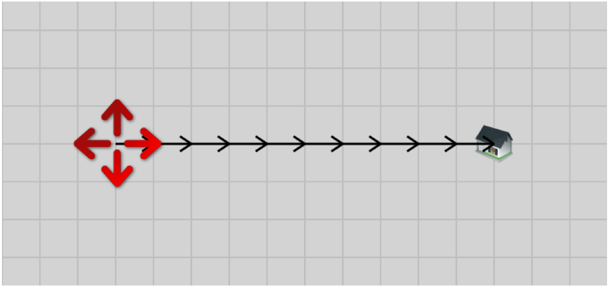
What to measure

Static Scenario



- Explorer fixed
- Quality measurement: Time to *optimal* chain

Dynamic Scenario



- Chain in optimal condition
- Explorer moving
- Quality measurement:
  - Possible speed of explorer
  - Maximal chain length

What can we expect?

- Dynamic Scenario
  - Explorer can move as fast as a relay
  - *constant*

What can we expect?

- Dynamic Scenario
  - Explorer can move as fast as a relay
  - *constant*
  - Chain length? O(minimal length)

What can we expect?

- Dynamic Scenario
  - Explorer can move as fast as a relay
  - *constant*
  - Chain length? O(minimal length)
- Static Scenario
  - There are cases where a (constant speed moving) relay needs n timesteps to get close to the direct line.

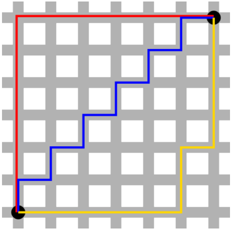
Strategy Description

Manhattan Hopper

- All stations move on a grid
- Chain remains valid
- Relays move at most constant distance
- Uses Manhattan distance

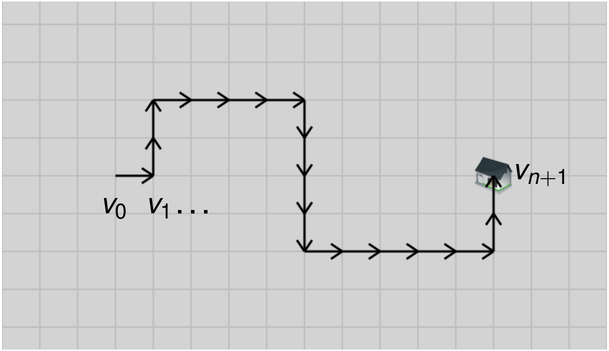
Manhattan Hopper

- All stations move on a grid
- Chain remains valid
- Relays move at most constant distance
- Uses Manhattan distance



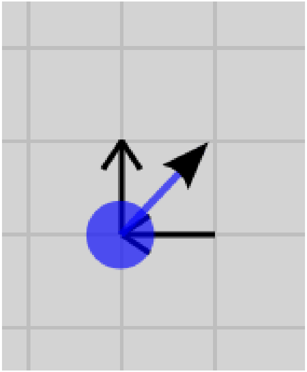
- $d = \Delta_x + \Delta_y$

Manhattan Hopper Description

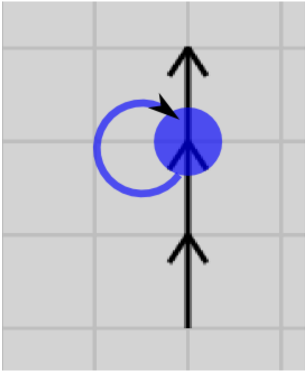


- Executed sequentally. v_{i+1} moves after v_i
- One sequence is called a *run*

Manhattan Hopper Description

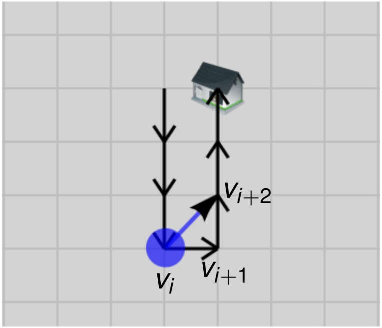


Neighbors not in line $\rightarrow$ move



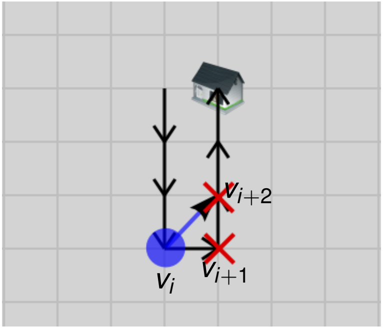
Neighbors in line $\rightarrow$ stay

Manhattan Hopper Description



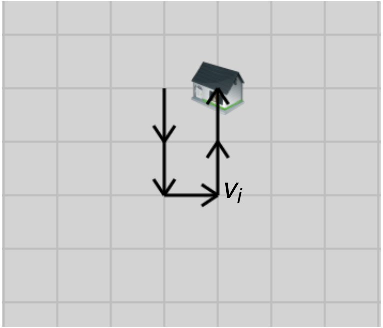
- If v_i moves to v_{i+2}. v_{i+1} and v_{i+2} are removed.

Manhattan Hopper Description



- If v_i moves to v_{i+2}. v_{i+1} and v_{i+2} are removed.
- v_{i+1} and v_{i+2} are removed.

Manhattan Hopper Description



- If v_i moves to v_{i+2}. v_{i+1} and v_{i+2} are removed.
- v_{i+1} and v_{i+2} are removed.
- A remove operation ends the run.

A little example

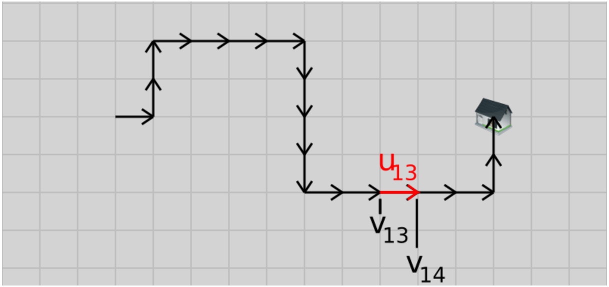
Static Scenario Performance

Static Scenario

Theorem 1

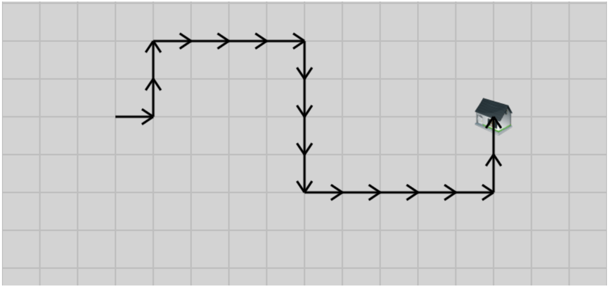
- *After n runs, the chain has optimal length*

Configuration



- $\vec{u}_i = position(v_{i+1}) - position(v_i)$

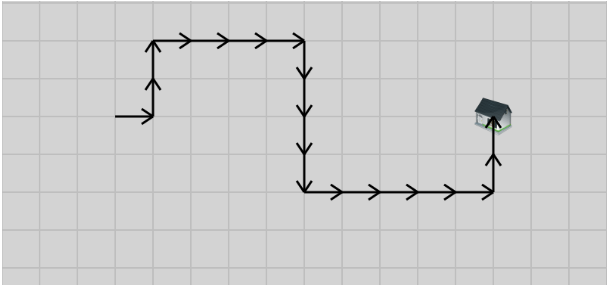
Configuration



- $\vec{u}_i = position(v_{i+1}) - position(v_i)$
-

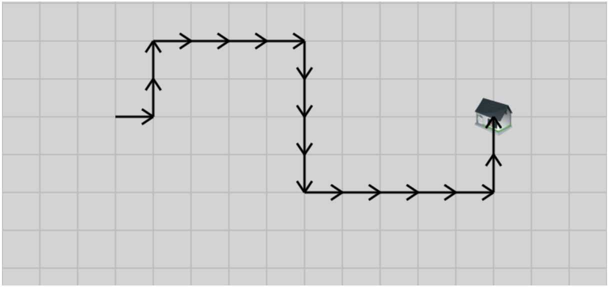
$$C = (\Rightarrow, \Uparrow, \Uparrow, \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \Uparrow, \Uparrow)$$
$$= (\vec{u_0}, \vec{u_1}, \ldots, \vec{u_k})$$

Configuration



- $\vec{u}_i = position(v_{i+1}) - position(v_i)$
-
$$\begin{aligned} C &= (\Rightarrow, \Uparrow, \Uparrow, \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \Uparrow, \Uparrow) \\ &= (\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_k) \end{aligned}$$
- $\vec{u}_i$ and $\vec{u}_j$ are *oppositional* $\leftrightarrow \vec{u}_i = -\vec{u}_j$

Configuration



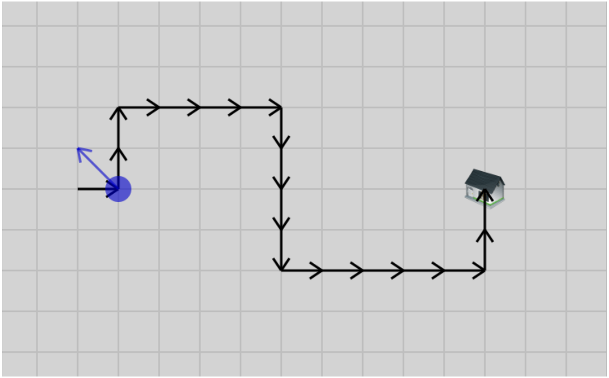
- $\vec{u}_i = position(v_{i+1}) - position(v_i)$
- $$C = (\Rightarrow, \Uparrow, \Uparrow, \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \Uparrow, \Uparrow)$$
$$= (\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_k)$$
- $\vec{u}_i$ and $\vec{u}_j$ are *oppositional* $\leftrightarrow \vec{u}_i = -\vec{u}_j$
- Optimal (Manhattan) length configuration?

Static Scenario - Strategy Effects On Configuration

Lemma 2

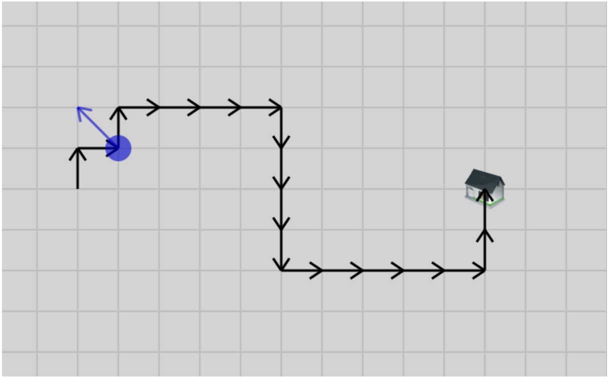
- *Let* $C = (\vec{u_0}, \vec{u_1}, \vec{u_2} \ldots, \vec{u_k})$.
- *Assume a run finishes without removing any relay.*
- $C' = (\vec{u_1}, \vec{u_2}, \ldots, \vec{u_k}, \vec{u_0})$ *is the configuration after the run.*
- *Also afterwards* $\vec{u_0}$ *is not oppositional to any other.*

Static Scenario - Strategy Effects On Configuration



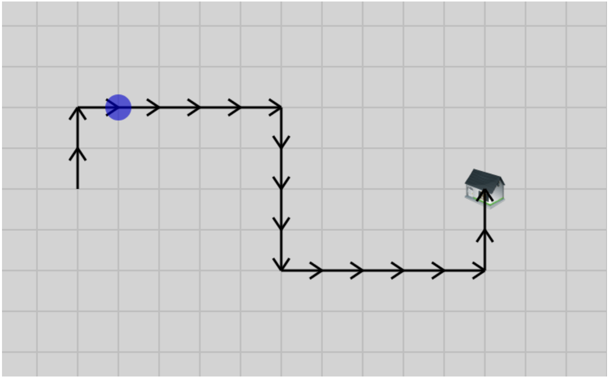
$$C = (\underbrace{\Rightarrow, \Uparrow}, \Uparrow, \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \Uparrow, \Uparrow)$$

Static Scenario - Strategy Effects On Configuration



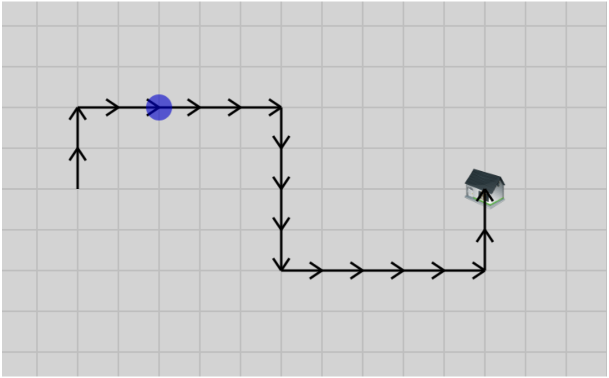
$$C = (\Uparrow, \underbrace{\Rightarrow, \Uparrow,} \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \Uparrow, \Uparrow)$$

Static Scenario - Strategy Effects On Configuration



$$C = (\Uparrow, \Uparrow, \underbrace{\Rightarrow, \Rightarrow}, \Rightarrow, \ldots, \Rightarrow, \Uparrow, \Uparrow)$$

Static Scenario - Strategy Effects On Configuration



$$C = (\Uparrow, \Uparrow, \Rightarrow, \underbrace{\Rightarrow, \Rightarrow}, \ldots, \Rightarrow, \Uparrow, \Uparrow)$$

Static Scenario - Strategy Effects On Configuration

- If $\vec{u}_0$ is oppositional to any other $\vec{u}_i$, $\vec{u}_0$ will meet it at some point

$$C = (\ldots, \underbrace{\Rightarrow, \Leftarrow}, \ldots)$$

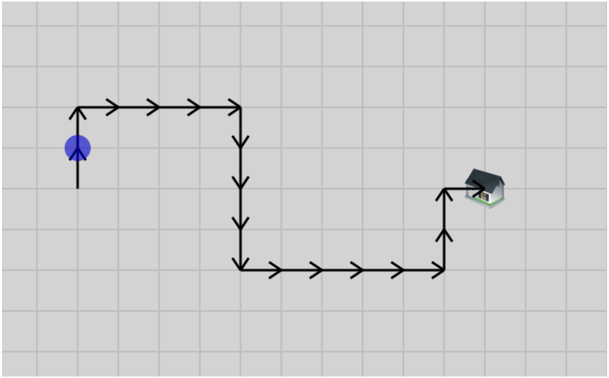
- triggers a removal

Static Scenario - Strategy Effects On Configuration

Lemma 3

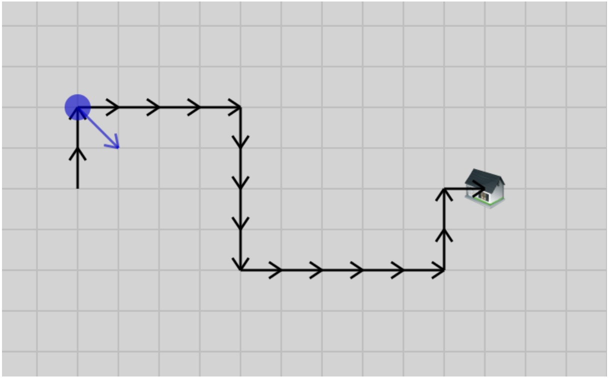
- *Let* $C = (\vec{u_0}, \vec{u_1}, \vec{u_2} \ldots, \vec{u_k})$.
- *The run finishes with removing* v_i *and* v_{i+1} *if and only if* $\vec{u_{i+1}}$ *is the first vector oppositional to* $\vec{u_0}$.
- $C' = (\vec{u_1}, \vec{u_2}, \ldots, \vec{u_i}, \vec{u_{i+2}}, \ldots \vec{u_k})$ *is the configuration after the run.*

Static Scenario - Strategy Effects On Configuration



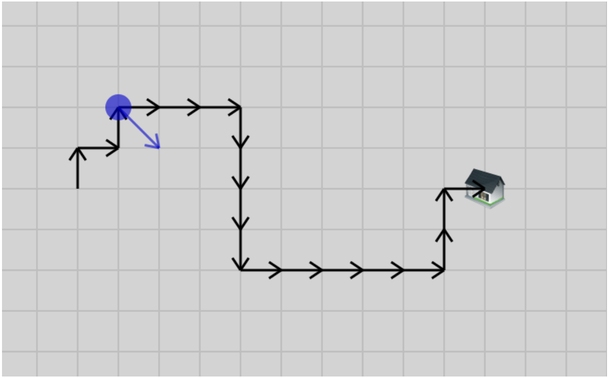
$$C = (\Uparrow, \Uparrow, \Rightarrow, \Rightarrow, \Rightarrow, \Downarrow, \ldots, \Uparrow, \Uparrow, \Rightarrow)$$

Static Scenario - Strategy Effects On Configuration



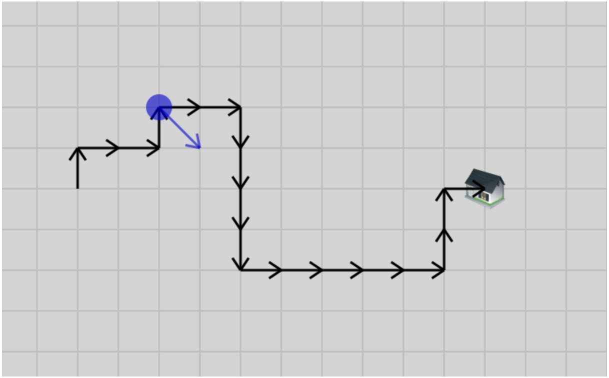
$$C = (\Uparrow, \underbrace{\colorbox{yellow}{$\Uparrow$}, \Rightarrow}, \Rightarrow, \Rightarrow, \Downarrow, \ldots, \Uparrow, \Uparrow, \Rightarrow)$$

Static Scenario - Strategy Effects On Configuration



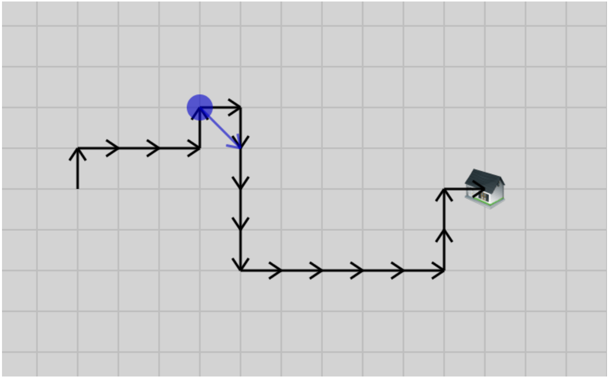
$$C = (\Uparrow, \Rightarrow, \underbrace{\Uparrow, \Rightarrow}, \Rightarrow, \Downarrow, \ldots, \Uparrow, \Uparrow, \Rightarrow)$$

Static Scenario - Strategy Effects On Configuration



$$C = (\Uparrow, \Rightarrow, \Rightarrow, \underbrace{\Uparrow, \Rightarrow}, \Downarrow, \ldots, \Uparrow, \Uparrow, \Rightarrow)$$

Static Scenario - Strategy Effects On Configuration



$$C = (\Uparrow, \Rightarrow, \Rightarrow, \Rightarrow, \underbrace{\Uparrow, \Rightarrow}, \Downarrow, \quad \ldots, \Uparrow, \Uparrow, \Rightarrow)$$

$$C' = (\Uparrow, \Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow, \underbrace{\Uparrow, \Downarrow}, \quad \ldots, \Uparrow, \Uparrow, \Rightarrow)$$

$$C'' = (\Uparrow, \Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow, \quad \ldots, \Uparrow, \Uparrow, \Rightarrow)$$

Static Scenario - Strategy Effects On Configuration

Lemma 3

- *Let* $C = (\vec{u_0}, \vec{u_1}, \vec{u_2} \ldots, \vec{u_k})$.
- *The run finishes with removing* v_i *and* v_{i+1} *if and only if* $\vec{u_{i+1}}$ *is the first vector oppositional to* $\vec{u_0}$.
- $C' = (\vec{u_1}, \vec{u_2}, \ldots, \vec{u_i}, \vec{u_{i+2}}, \ldots \vec{u_k})$ *is the configuration after the run.*

Static Scenario - Some Observations

- Vectors are never created, label them uniquely

$$C_1 = (\vec{a_0}, \vec{a_1}, \ldots, \vec{a_k})$$

Static Scenario - Some Observations

- Vectors are never created, label them uniquely

$$C_1 = (\vec{a}_0, \vec{a}_1, \ldots, \vec{a}_k)$$

- In every run $\vec{u}_i$ ($i \neq 0$) reduces its position at least by one
  - Case 1: No removal
  - Case 2: Removal happens and $\vec{u}_i$ is before the removal
  - Case 3: Removal happens and $\vec{u}_i$ is after the removal

Static Scenario

- Assume after n runs, there is an oppositional pair $\vec{u_p}$ and $\vec{u_q}$ with $p < q$.

$$C = (\ldots, \underbrace{u_p, \ldots, u_n}_{\text{Distance: } n-p})$$

- At most $n - p + 1$ runs earlier, $\vec{u_p}$ was at position 0
- and hence would have been removed.

Static Scenario

- Assume after n runs, there is an oppositional pair $\vec{u_p}$ and $\vec{u_q}$ with $p < q$.

$$C = (\dots, \underbrace{u_p, \dots, u_n}_{\text{Distance: } n-p})$$

- At most $n - p + 1$ runs earlier, $\vec{u_p}$ was at position 0
- and hence would have been removed.
- After n rounds, there are no more oppositional pairs.

Static Scenario

- It takes n rounds to reach minimal length. Timesteps?

Static Scenario

- It takes *n* rounds to reach minimal length. Timesteps?
- Pipeline! Start new run every 3 time steps.

Static Scenario

- It takes n rounds to reach minimal length. Timesteps?
- Pipeline! Start new run every 3 time steps.
- After $3n + n = 4n$ time steps the chain is optimal

Dynamic Scenario Performance

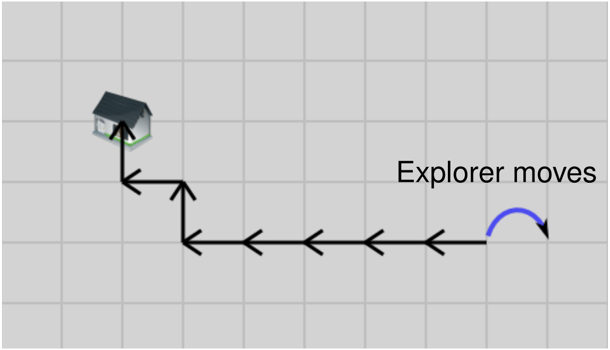
Dynamic Scenario

- Must handle explorer moves

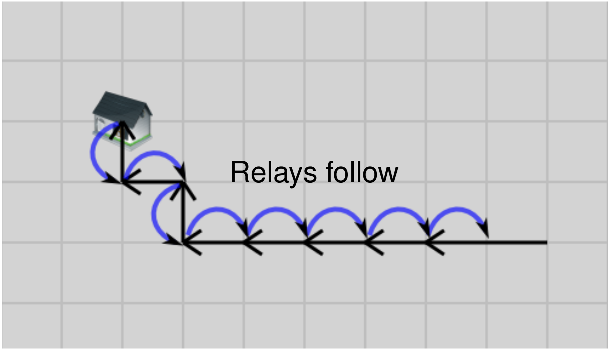
Dynamic Scenario

- Must handle explorer moves
- Perform *Follow* run
- Then perform *Hopper* run
  - The *Hopper* run is what we have seen before

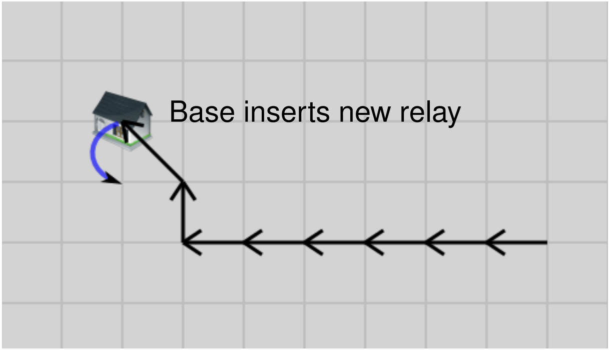
Follow Run



Follow Run



Follow Run



Follow Run

Dynamic Scenario Performance

Lemma 4

Let the chain have optimal length prior to the explorer's movement. Then after the explorer's movement, the Hopper and Follow run bring the chain to an optimal length.

Dynamic Scenario Performance

Lemma 4

Let the chain have optimal length prior to the explorer's movement. Then after the explorer's movement, the Hopper and Follow run bring the chain to an optimal length.

Proof.

- Let C be the configuration before the movement and C' after the *follow* run.

Dynamic Scenario Performance

Lemma 4

Let the chain have optimal length prior to the explorer's movement. Then after the explorer's movement, the Hopper and Follow run bring the chain to an optimal length.

Proof.

- Let C be the configuration before the movement and C' after the *follow* run.
- No pair of oppositional vectors in C

Dynamic Scenario Performance

Lemma 4

Let the chain have optimal length prior to the explorer's movement. Then after the explorer's movement, the Hopper and Follow run bring the chain to an optimal length.

Proof.

- Let C be the configuration before the movement and C' after the *follow* run.
- No pair of oppositional vectors in C
- At most one pair of oppositional in C'

Dynamic Scenario Performance

Lemma 4

Let the chain have optimal length prior to the explorer's movement. Then after the explorer's movement, the Hopper and Follow run bring the chain to an optimal length.

Proof.

- Let C be the configuration before the movement and C' after the *follow* run.
- No pair of oppositional vectors in C
- At most one pair of oppositional in C'
- One *Hopper* removes the first pair of oppositional vectors

Dynamic Scenario Performance

Lemma 4

Let the chain have optimal length prior to the explorer's movement. Then after the explorer's movement, the Hopper and Follow run bring the chain to an optimal length.

Proof.

- Let C be the configuration before the movement and C' after the *follow* run.
- No pair of oppositional vectors in C
- At most one pair of oppositional in C'
- One *Hopper* removes the first pair of oppositional vectors
- Hence there is no pair at the end and hence the chain has optimal length

□

Dynamic Scenario Performance

- $d_r :=$ (Manhattan) distance between explorer and home at beginning of round r.

Dynamic Scenario Performance

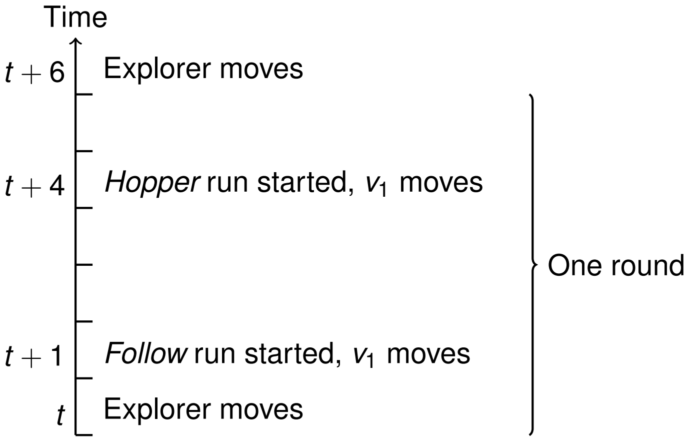
- $d_r :=$ (Manhattan) distance between explorer and home at beginning of round r.
- $d_r = 4.5$
- Number of relays $= 9$
- Optimal chain: Number of relays = $2d_r$

Dynamic Scenario Performance

- Explorer speed?

Dynamic Scenario Performance

- Explorer speed?
- Must pipeline



Dynamic Scenario Performance

Theorem 5

Assume we start with an optimal chain. Then, the chain maintained by the strategy has the following properties before each round r.

1. *The chain remains connected*
2. *The explorer may move a distance of $\frac{1}{2}$ every round, i.e. every 6th time step*
3. *Relays move at most constant distance per round*
4. *The number of relays used in the chain is at most $3d_r + 2$*

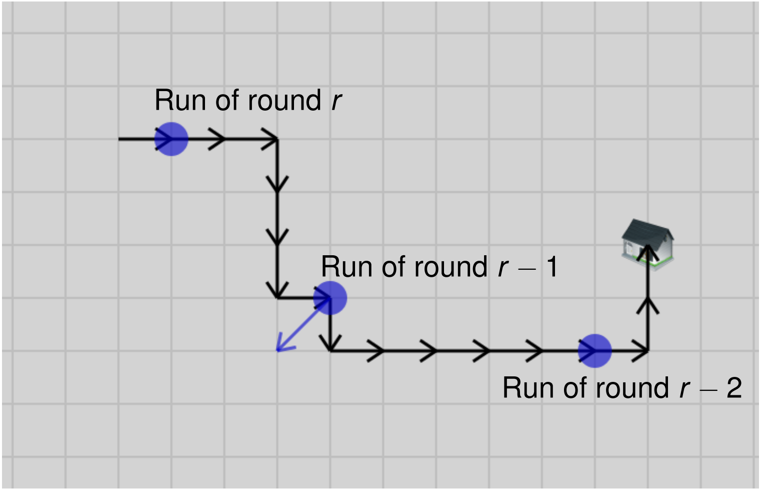
Dynamic Scenario Performance - Number Of Relays

- Each *Hopper* run operates on an optimal chain.
  - Chain has $2d_r$ relays.
  - Run takes at most $2d_r + 2$ time steps.

Dynamic Scenario Performance - Number Of Relays

- Each *Hopper* run operates on an optimal chain.
  - Chain has $2d_r$ relays.
  - Run takes at most $2d_r + 2$ time steps.
- Fix round r
- Number of relays $\leq 2d_r + 2$ (number of unfinished *Hopper* runs)

Dynamic Scenario Performance - Number Of Relays



How Many Unfinished Hopper runs Are There?

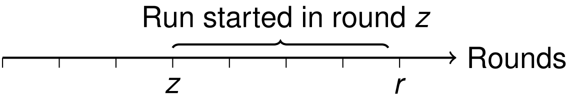
Lemma 6

There are at most $\frac{d_r+1}{2}$ *unfinished runs in round* r*.*

$\leftrightarrow$ The run started in round $r - \frac{d_r+1}{2}$ is finished at round r

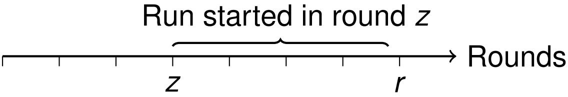
How Many Unfinished Hopper Runs Are There?

- $r :=$ current round
- $z :=$ earlier round



How Many Unfinished Hopper Runs Are There?

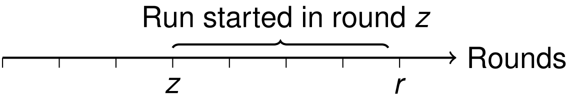
- $r :=$ current round
- $z :=$ earlier round



1. $z < r - \frac{d_r+1}{2}$

How Many Unfinished Hopper Runs Are There?

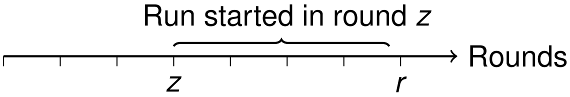
- $r :=$ current round
- $z :=$ earlier round



1. $z < r - \frac{d_r+1}{2}$
2. Run of round z needs $< 2d_z + 2$ timesteps to finish

How Many Unfinished Hopper Runs Are There?

- $r :=$ current round
- $z :=$ earlier round



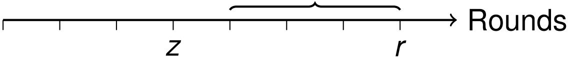
1. $z < r - \frac{d_r+1}{2}$
2. Run of round z needs $< 2d_z + 2$ timesteps to finish
3. Max. distance of explorer between z and $r = \frac{r-z}{2}$
   $\rightarrow d_z \leq d_r + \frac{r-z}{2}$

- Run of round z ends in which round?

Dynamic Scenario Performance

- $z < r - \frac{d_r+1}{2}$

Unfinished runs at r ? At most $r - z = \frac{d_r+1}{2}$ many

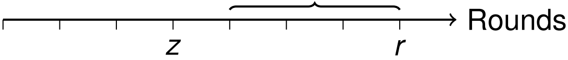


- Number of relays $\leq 2d_r + 2$ (number of unfinished *Hopper* runs)
- Number of relays $\leq 2d_r + 2\frac{d_r+1}{2} = 3d_r + 1$

Dynamic Scenario Performance

- $z < r - \frac{d_r+1}{2}$

Unfinished runs at r ? At most $r - z = \frac{d_r+1}{2}$ many



- Number of relays $\leq 2d_r + 2$ (number of unfinished *Hopper* runs)
- Number of relays $\leq 2d_r + 2\frac{d_r+1}{2} = 3d_r + 1$
- The strategy keeps chain length in $O(d_r)$

Outlook

- Can be generalized (drop grid requirement)
- Keeps optimal characteristics

Conclusion

- The oscillation of the strategy and its sequential nature improve the *Go-to-the-Middle* strategy
- It converts a chain into an optimal in $O(n)$ timesteps ($n =$ number of relays)
  - Which is optimal
- It allows the explorer to move with constant speed.
  - Which is optimal

Questions?