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

Lukas Humbel

2. Mai 2012
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
- 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
- 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
- Moves with constant speed
- Moves with constant speed
- Can be removed everywhere
- Inserted only at home
Chain Attributes

- *Valid* condition
- *Optimal* condition
What to measure
Explorer fixed

Quality measurement: Time to optimal chain
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
  - \textit{constant}
What can we expect?

- Dynamic Scenario
  - Explorer can move as fast as a relay
  - constant
  - Chain length? $O(\text{minimal length})$
What can we expect?

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

- **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

\[ d = \Delta_x + \Delta_y \]
Executed sequentially. $v_{i+1}$ moves after $v_i$

One sequence is called a *run*
Neighbors not in line → move

Neighbors in line → stay
If \( v_i \) moves to \( v_{i+2} \), \( v_{i+1} \) and \( v_{i+2} \) are removed.
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.
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.
Static Scenario Performance
Theorem 1

After $n$ runs, the chain has optimal length
\[ \vec{u}_i = position(v_{i+1}) - position(v_i) \]
$\vec{u}_i = \text{position}(v_{i+1}) - \text{position}(v_i)$

$C = (\Rightarrow, \uparrow, \uparrow, \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \uparrow, \uparrow) = (\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_k)$
\[ \vec{u}_i = \text{position}(v_{i+1}) - \text{position}(v_i) \]

\[ C = (\Rightarrow, \uparrow, \uparrow, \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \uparrow, \uparrow) \]
\[ = (\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_k) \]

\[ \vec{u}_i \text{ and } \vec{u}_j \text{ are oppositional } \iff \vec{u}_i = -\vec{u}_j \]
\[ \vec{u}_i = \text{position}(v_{i+1}) - \text{position}(v_i) \]

\[ C = (\Rightarrow, \uparrow, \uparrow, \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \uparrow, \uparrow) \]

\[ = (\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_k) \]

\[ \vec{u}_i \text{ and } \vec{u}_j \text{ are oppositional } \iff \vec{u}_i = -\vec{u}_j \]

Optimal (Manhattan) length 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

\[ C = (\Rightarrow, \uparrow, \uparrow, \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \uparrow, \uparrow) \]
$C = (\uparrow, \Rightarrow, \uparrow, \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \uparrow, \uparrow)$
\[ C = (\uparrow, \uparrow, \Rightarrow, \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \uparrow, \uparrow) \]
$C = (\uparrow, \uparrow, \Rightarrow, \Rightarrow, \Rightarrow, \ldots, \Rightarrow, \uparrow, \uparrow)$
If $\vec{u}_0$ is oppositional to any other $\vec{u}_i$, $\vec{u}_0$ will meet it at some point

$$C = (\ldots, \Rightarrow, \Leftarrow, \ldots)$$

triggers a removal
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.
$C = (\uparrow, \uparrow, \Rightarrow, \Rightarrow, \Rightarrow, \downarrow, \ldots, \uparrow, \uparrow, \Rightarrow)$
$C = (\uparrow, \uparrow, \Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow, \downarrow, \ldots, \uparrow, \uparrow, \Rightarrow)$
\[ C = (\uparrow, \Rightarrow, \uparrow, \Rightarrow, \Rightarrow, \downarrow, \ldots, \uparrow, \uparrow, \Rightarrow) \]
$C = (\uparrow, \Rightarrow, \Rightarrow, \uparrow, \Rightarrow, \downarrow, \ldots, \uparrow, \uparrow, \Rightarrow)$
Static Scenario - Strategy Effects On Configuration

\[ C = (\uparrow, \Rightarrow, \Rightarrow, \Rightarrow, \uparrow, \Rightarrow, \downarrow, \ldots, \uparrow, \uparrow, \Rightarrow) \]

\[ C' = (\uparrow, \Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow, \uparrow, \downarrow, \ldots, \uparrow, \uparrow, \Rightarrow) \]

\[ C'' = (\uparrow, \Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow, \ldots, \uparrow, \uparrow, \Rightarrow) \]
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 $u_{i+1}$ is the first vector oppositional to $\vec{u_0}$.

$C' = (\vec{u_1}, \vec{u_2}, \ldots, \vec{u_i}, u_{i+2}, \ldots \vec{u_k})$ is the configuration after the run.
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
Assume after \( n \) runs, there is an oppositional pair \( \vec{u}_p \) and \( \vec{u}_q \) with \( p < q \).

\[
C = (\ldots, u_p, \ldots, u_n)
\]

Distance: \( n - p \)

At most \( n - p + 1 \) runs earlier, \( \vec{u}_p \) was at position 0

and hence would have been removed.
Assume after $n$ runs, there is an oppositional pair $\vec{u}_p$ and $\vec{u}_q$ with $p < q$.

$$C = (\ldots, u_p, \ldots, u_n)$$

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.
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.
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

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

Base inserts new relay
Follow Run
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.
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.
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 $C$
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 $C'$
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
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
$d_r := \text{(Manhattan) distance between explorer and home at beginning of round } r.$
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 $= 2d_r$
Explorer speed?
Dynamic Scenario Performance

- Explorer speed?
- Must pipeline

\[
\begin{align*}
\text{Time} & \\
(t+6) & \text{Explorer moves} \\
(t+4) & \text{Hopper run started, } v_1 \text{ moves} \\
(t+1) & \text{Follow run started, } v_1 \text{ moves} \\
t & \text{Explorer moves}
\end{align*}
\]

One round
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 $3d_r + 2$
Each *Hopper* run operates on an optimal chain.
- Chain has $2d_r$ relays.
- Run takes at most $2d_r + 2$ time steps.
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

Run of round $r$

Run of round $r - 1$

Run of round $r - 2$
How Many Unfinished Hopper runs Are There?

Lemma 6

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

$\iff$ The run started in round $r - \frac{d_r + 1}{2}$ is finished at round $r$
$r := \text{current round}$

$z := \text{earlier round}$

Run started in round $z$

\[ r > z + 2 \text{ timesteps} \]

Max. distance of explorer between $z$ and $r$

\[ d_z \leq d_r + (r - z)^2 \]

Run of round $z$ ends in which round?
How Many Unfinished Hopper Runs Are There?

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

**Run started in round $z**

$z < r - \frac{d_r+1}{2}$
How Many Unfinished Hopper Runs Are There?

- \( r \) := current round
- \( z \) := earlier round

Run started in round \( z \)

- 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

### Run started in round $z$

![Diagram showing the relationship between $z$, $r$, and the rounds]

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 \( \leq 2d_r + 2 \frac{d_r + 1}{2} = 3d_r + 1 \)
• $z < r - \frac{d_r + 1}{2}$

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 $\leq 2d_r + 2 \cdot \frac{d_r + 1}{2} = 3d_r + 1$

The strategy keeps chain length in $O(d_r)$
Outlook

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