Maximizing Network Lifetime on the Line with Adjustable Sensing Ranges

Amotz Bar-Noy  Ben Baumer

The Graduate Center of the City University of New York

ALGOSENSORS 2011
Max Planck Institut für Informatik
Saarbrücken, Germany
September 8th, 2011
Introduction

Problem (Adjustable Range Restricted Strip Cover)

*Input:* A set $S$ of $n$ adjustable-range sensors on the line $[0, 1]$

*Output:* A schedule $(\mathbf{R}, \mathbf{t})$ of maximum lifetime $T$

*Constraints:*

- *(Coverage)* Every point $(x, t) \in [0, 1] \times [0, T]$ is covered by some sensor $s \in S$
- *(Battery)* No sensor uses more than 1 unit of battery life

*Notation:*

- $\mathbf{R}$ is an $n \times k$ matrix of radial assignments ($k =$ # of time slots)
- $\mathbf{t}$ is a $k$-vector of time slots ($T = ||\mathbf{t}||_1$)

*Battery drainage rate:*

- In general, battery drains according to $r_i^\alpha$, for some $\alpha > 0$
- We consider only $\alpha = 1$
Motivation
Disaster Relief: Highway coverage
Example: Pre-emptive scheduling can help

- Consider \( S = \{ \frac{1}{8}, \frac{1}{2}, \frac{7}{8} \} \)

(a) \( T = 5 \frac{1}{3} \), preemptive

(b) \( T = 4 \frac{2}{3} \), non-preemptive
Related Work

- **Restricted Strip Cover (RSC)** [Buchsbaum, et al. SODA ’07]
  - Each sensor has a **fixed range**
  - Each sensor has a **fixed duration** (of lifetime)
  - Assumed **non-preemptive** scheduling
  - Proved NP-Hardness and gave $O(\log \log \log n)$ approximation algorithm
  - Constant factor algorithm [Gibson & Varadarajan ’09]
Related Work

- **Restricted Strip Cover (RSC)** [Buchsbaum, et al. SODA ’07]
  - Each sensor has a **fixed range**
  - Each sensor has a **fixed duration** (of lifetime)
  - Assumed **non-preemptive** scheduling
  - Proved NP-Hardness and gave $O(\log \log \log n)$ approximation algorithm
  - Constant factor algorithm [Gibson & Varadarajan ’09]

- **Duty-Cycling**: Maximize the number of covers $k$
  - Pach and Tóth: $k$-fold cover can be decomposed into $\Omega(\sqrt{k})$ covers
  - Improved to optimal $\Omega(k)$ [Aloupis, et al. ’10]
  - Restrictions lifted by Gibson & Varadarajan
Related Work

- **Restricted Strip Cover (RSC)** [Buchsbaum, et al. SODA ’07]
  - Each sensor has a **fixed range**
  - Each sensor has a **fixed duration** (of lifetime)
  - Assumed **non-preemptive** scheduling
  - Proved NP-Hardness and gave $O(\log \log \log n)$ approximation algorithm
  - Constant factor algorithm [Gibson & Varadarajan ’09]

- **Duty-Cycling**: Maximize the number of covers $k$
  - Pach and Tóth: $k$-fold cover can be decomposed into $\Omega(\sqrt{k})$ covers
  - Improved to optimal $\Omega(k)$ [Aloupis, et al. ’10]
  - Restrictions lifted by Gibson & Varadarajan

- **Target Coverage**: Exact **poly-time** algorithm [Peleg & Lev-Tov ’05]
  - No NP-Hardness result is known for area coverage
Our Contribution

- First to consider true lifetime for area coverage with adjustable sensing ranges
Our Contribution

- First to consider true lifetime for area coverage with adjustable sensing ranges
- Worst-case and average-case analysis for several natural algorithms
Our Contribution

- First to consider true `lifetime` for area coverage with adjustable sensing ranges
- Worst-case and average-case analysis for several natural algorithms
- Non-trivial linear time algorithm that performs very well on average
  - \( \sim 90\% \) of theoretical max
  - We are fighting for that last 10%!

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RoundRobin</th>
<th>( T )</th>
<th>( \text{Var} )</th>
<th>( \text{AC} )</th>
<th>( \text{WC} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 2/3 )-RoundRobin</td>
<td>1.386</td>
<td>0.078</td>
<td>0.693</td>
<td>2/3</td>
<td></td>
</tr>
<tr>
<td>( 2/3 )-RoundRobin</td>
<td>1.738</td>
<td>0.022</td>
<td>0.869</td>
<td>2/3</td>
<td></td>
</tr>
<tr>
<td>Optimized ( 2/3 )-RoundRobin</td>
<td>1.791</td>
<td>0.896</td>
<td>2/3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Our Contribution

- First to consider true lifetime for area coverage with adjustable sensing ranges
- Worst-case and average-case analysis for several natural algorithms
- Non-trivial linear time algorithm that performs very well on average
  - $\sim 90\%$ of theoretical max
  - We are fighting for that last 10%!

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$E[T]$</th>
<th>$Var[T]$</th>
<th>$AC$</th>
<th>WC</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoundRobin</td>
<td>1.386</td>
<td>0.078</td>
<td>0.693</td>
<td>2/3</td>
</tr>
<tr>
<td>$k$-RoundRobin</td>
<td>1.386</td>
<td>0.078</td>
<td>0.693</td>
<td>2/3</td>
</tr>
<tr>
<td>$\log_2$-RoundRobin</td>
<td>1.738</td>
<td>0.022</td>
<td>0.869</td>
<td>2/3</td>
</tr>
<tr>
<td>Optimized $\log_2$-RoundRobin</td>
<td>1.791</td>
<td></td>
<td>0.896</td>
<td>2/3</td>
</tr>
</tbody>
</table>
Approximability

Lemma

If $|S| = n$, then $n \leq T_{OPT} \leq 2n$
Approximability

Lemma

If $|S| = n$, then $n \leq T_{OPT} \leq 2n$

Proof.

- Lower bound: any reasonable algorithm achieves $T \geq n$
- Upper bound: any sensor consumes at most 2 units of space-time
**Approximability**

**Lemma**

If $|S| = n$, then $n \leq T_{OPT} \leq 2n$

**Proof.**

- Lower bound: any reasonable algorithm achieves $T \geq n$
- Upper bound: any sensor consumes at most 2 units of space-time

- A 2-approximation is trivial
Approximability

Lemma

If $|S| = n$, then $n \leq T_{OPT} \leq 2n$

Proof.

- Lower bound: any reasonable algorithm achieves $T \geq n$
- Upper bound: any sensor consumes at most 2 units of space-time

- A 2-approximation is trivial
- Can we do better?
RoundRobin

- RoundRobin is perhaps the simplest algorithm
  - Have the sensors take turns covering the whole line
  - Approximation between 0.548 and 2/3
- Consider $S = \left\{ \frac{1}{4}, \frac{3}{4} \right\}$
- $\text{RoundRobin} \leq \frac{2}{3} \text{OPT}$

(c) $T_{OPT} = 4$

(d) $T_{RR} = \frac{2}{3}$
Average Case Analysis for RoundRobin

Distribution of network lifetime $T$ for a sensor:

$$F_T(t) = 2 \left(1 - \frac{1}{t}\right), \quad t \in [1, 2]$$
Average Case Analysis for RoundRobin

- Distribution of network lifetime $T$ for a sensor:

$$F_T(t) = 2 \left(1 - \frac{1}{t}\right), \quad t \in [1, 2]$$

- Moments of lifetime:
  - $\mathbb{E}[T] = \mu_T = 2 \ln 2 \approx 1.386$
  - $\text{Var}[T] = \sigma_T^2 = 2 - 4 \ln^2(2) \approx 0.078$
Average Case Analysis for RoundRobin

- Distribution of network lifetime $T$ for a sensor:
  \[ F_T(t) = 2 \left( 1 - \frac{1}{t} \right), \quad t \in [1, 2] \]

- Moments of lifetime:
  1. $\mathbb{E}[T] = \mu_T = 2 \ln 2 \approx 1.386$
  2. $\text{Var}[T] = \sigma_T^2 = 2 - 4 \ln^2(2) \approx 0.078$

- Average-case performance on uniformly distributed sensors is:
  \[ 1.386 \cdot n \]
Example: RoundRobin

Space–Time Diagram for Network Lifetime

Coverage (x)

Time (t)

n = 10; T = 14.285; \bar{T} = 1.428
Performance of RoundRobin

Average Network Lifetime for RoundRobin

Sensor Location

Normalized Lifetime

$E[T] = 1.386$, $Var[T] = 0.078$
**log₂-RoundRobin**

- **Idea:** Generalize the subdivision of areas of responsibility

---

**Analysis:**
- Worst-case lifetime of \( \frac{4}{3} \) (96% of average lifetime for RoundRobin)
- Expected lifetime is:
  
  \[
  E[ T ] = 2 \ln \left( 3 \cdot 5 \cdot 9 \cdots 2^k + 1 \right) \approx 1.738
  \]
**log₂-RoundRobin**

- **Idea:** Generalize the subdivision of areas of responsibility
- **log₂-RoundRobin:**
  - Make each sensor responsible only for nearby areas
  - Size of area determined by location
  - Fix a depth $k$, sensors cover subintervals around $i/2^k$
  - Sensors closer to $\frac{1}{2}, \frac{1}{4}, \frac{3}{4}$, etc. cover larger subintervals

Analysis:
- Worst-case lifetime of $\frac{4}{3}$ (96% of average lifetime for RoundRobin)
- Expected lifetime is:
  $$E[T] = 2 \ln (3 \cdot 5 \cdot 9 \cdot \cdots \cdot 2^{k+1}) \cdot 2 \cdot 4 \cdot 8 \cdot \cdots \cdot 2^k) \approx 1.738$$
**log\_2-RoundRobin**

- **Idea:** Generalize the subdivision of areas of responsibility
- **log\_2-RoundRobin:**
  - Make each sensor responsible only for nearby areas
  - Size of area determined by location
  - Fix a depth $k$, sensors cover subintervals around $i/2^k$
  - Sensors closer to $\frac{1}{2}$, $\frac{1}{4}$, $\frac{3}{4}$, etc. cover larger subintervals
- **Analysis:**
  - Worst-case lifetime of $4/3$ (96% of average lifetime for RoundRobin)
  - Expected lifetime is:
    \[
    \mathbb{E}[T] = 2 \ln \left( \frac{3 \cdot 5 \cdot 9 \cdots 2^k + 1}{2 \cdot 4 \cdot 8 \cdots 2^k} \right) \approx 1.738
    \]
Load-Balancing in $\log_2$-RoundRobin

- Now we have $2^k$ subintervals, with an average of $n/2^k$ sensors in each.
- Chernoff: with high probability, deviations are $O\left(\sqrt{\frac{n \ln n}{2^k}}\right)$.
- Require $k = O(\ln n)$.
- Expected Average Lifetime:
  
  \[
  \frac{n_1}{n} 1.738 + \frac{n_2}{n} 1.386 \rightarrow 1.738
  \]

  since $\frac{n_2}{n} \rightarrow 0$ as $n \rightarrow \infty$. 
Example: $\log_2$-RoundRobin

Space–Time Diagram for Network Lifetime

Coverage (x)

Time (t)

n = 200; T = 299.465; $\overline{bar(T)} = 1.497$
Average Network Lifetime for $\log_2$-RoundRobin

Sensor Location

Normalized Lifetime

$\log_2$-RoundRobin

B. Baumer (CUNY)

Network Lifetime

ALGO 2011 15 / 20
**log**₂-RoundRobin

The average network lifetime for log₂-RoundRobin with depth = 2 is shown in the graph. The normalized lifetime is depicted along the y-axis, ranging from 1.0 to 2.0, and the sensor location is along the x-axis, ranging from 0.0 to 1.0. The expected network lifetime, denoted as **E[T]**, is calculated as 1.493.
Average Network Lifetime for $\log_2$-RoundRobin

Sensor Location

Depth = 3, $E[T] = 1.614$
**log\_2-RoundRobin**

Average Network Lifetime for log-RoundRobin

Sensor Location

Normalized Lifetime

Depth = 4, E[T] = 1.676
**log₂-RoundRobin**

Average Network Lifetime for log₂-RoundRobin

Sensor Location

Normalized Lifetime

depth = 5, E[T] = 1.707
log₂-RoundRobin

Average Network Lifetime for log₂-RoundRobin

Sensor Location

Normalized Lifetime

depth = 6, E[T] = 1.722
**log₂-RoundRobin**

### Average Network Lifetime for log₂-RoundRobin

**Depth:** 7, **$E[T] = 1.73$**

**Sensor Location**

<table>
<thead>
<tr>
<th>Normalized Lifetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
</tr>
<tr>
<td>0.2</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.6</td>
</tr>
<tr>
<td>0.8</td>
</tr>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>1.2</td>
</tr>
<tr>
<td>1.4</td>
</tr>
<tr>
<td>1.6</td>
</tr>
<tr>
<td>1.8</td>
</tr>
<tr>
<td>2.0</td>
</tr>
</tbody>
</table>

**Network Lifetime**

- **B. Baumer (CUNY)**
**log$_2$-RoundRobin**

Average Network Lifetime for log$_2$-RoundRobin

Sensor Location

Normalized Lifetime

depth = 8, $E[T] = 1.734$
**log₂-RoundRobin**

### Average Network Lifetime for log₂-RoundRobin

**Sensor Location**

- **Normalized Lifetime**
  - 0.0, 0.2, 0.4, 0.6, 0.8, 1.0
  - 1.0, 1.2, 1.4, 1.6, 1.8, 2.0

**Average Network Lifetime for log₂-RoundRobin**

- depth = 9, E[T] = 1.736
**log<sub>2</sub>-RoundRobin**

Average Network Lifetime for log–RoundRobin

Sensor Location

Normalized Lifetime

- Depth = 10, $E[T] = 1.737$
Optimization of $\log_2$-RoundRobin

- Observation: The worst performing level of $\log_2$-RoundRobin covers half of the interval!
  - Every interval borders a Worst Performing Interval (WPI)
  - Loads will remain balanced if we narrow the WPI’s uniformly
Optimization of $\log_2$-RoundRobin

- Observation: The worst performing level of $\log_2$-RoundRobin covers half of the interval!
  - Every interval borders a Worst Performing Interval (WPI)
  - Loads will remain balanced if we narrow the WPI’s uniformly

- Optimized $\log_2$-RoundRobin
  - Shrink the WPI’s by $\epsilon/2^{k+1}$
  - Find $\epsilon(k)$ that produces local maximum
  - Requires solution of $k - 1$ degree polynomial
  - Expected lifetime improves by 3.1% to 1.791
Intuition for Optimization

Average Network Lifetime for log-RoundRobin

Sensor Location

Normalized Lifetime

depth = 3, E[T] = 1.614
### Convergence for optimal $\epsilon$

| $k$ | $\epsilon$ | $T_k(0)$ | $T_k(\epsilon)$ | Gain % | $|U_k(k; \epsilon)|$ % |
|-----|------------|----------|-----------------|--------|-------------------|
| 2   | 0          | 1.492783 | 1.492783        | 0      | 50.00             |
| 3   | 0          | 1.614033 | 1.614033        | 0      | 50.00             |
| 4   | 0.211103   | 1.675576 | 1.696157        | 1.23   | 39.44             |
| 5   | 0.371297   | 1.706584 | 1.743439        | 2.16   | 31.44             |
| 6   | 0.448178   | 1.722149 | 1.767123        | 2.61   | 27.59             |
| 7   | 0.485871   | 1.729946 | 1.778990        | 2.84   | 25.71             |
| 8   | 0.504537   | 1.733848 | 1.784931        | 2.95   | 24.77             |
| 9   | 0.513826   | 1.735800 | 1.787904        | 3.00   | 24.31             |
| 10  | 0.518459   | 1.736777 | 1.789391        | 3.03   | 24.08             |
| 11  | 0.520773   | 1.737265 | 1.790134        | 3.04   | 23.96             |
| 12  | 0.521929   | 1.737509 | 1.790506        | 3.05   | 23.90             |
| 15  | 0.522941   | 1.737723 | 1.790831        | 3.06   | 23.85             |
| 20  | 0.523081   | **1.737752** | **1.790876** | 3.06   | 23.85             |

**Table:** Numerical Approximations for Optimal Choice of $\epsilon$
The Future

- Is this problem NP-hard?
- Can a richer characterization of $OPT$ improve the approximation bounds?
- What happens for $\alpha \neq 1$?
- What happens in 2D?
The End

Vielen Dank!