Frugal Sensor Assignment

M.P. Johnson[1] H. Rowaihy[2] D. Pizzocaro[3]
A. Bar-Noy[1] S. Chalmers[4] T. La Porta[2] A. Preece[3]

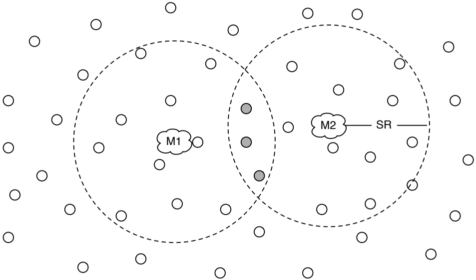
[1]The City University of New York [2]Pennsylvania State University [3]Cardiff University [4]University of Aberdeen

Supported by the International Technology Alliance (ITA), US ARL, UK MoD.

September 15, 2008

Mission-oriented sensor networks

- Typically, many sensor networks are application-specific
- New model: shared sensor networks
  - To each application, sensor network is mission-specific



- With sharing, comes arbitration
  - New algorithms and policies are required for sharing
  - Decisions based on resources, priorities, cost, ownership...

Outline

Outline

1. Motivation
2. Static problem
   - Problem and algorithms
   - Performance evaluation
3. Dynamic problem
   - Problem and algorithms
   - Performance evaluation
4. Conclusion

The problem

- A sensor network may be tasked with multiple simultaneous missions, e.g.
  - detect events in *this area*
  - perform localization (or stereo vision, or...) in an area
  - forward *this packet* to next hop

→ Such missions must compete for available sensors

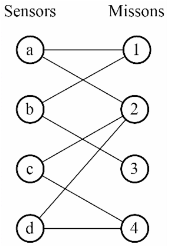
- Possible settings: disaster area, military, surveillance

- Some assignments might be better than others, due to
  - distance & geography
  - information modalities
  - remaining battery life

- *Which sensors should go to which missions?*

The problem: matching sensors to missions

- Context: sensor network w/ multiple sensors and missions
  $\longrightarrow$ assignment problem: which sensors should watch which targets?
  - Sensor-mission edges have utilities



- Not just weighted bipartite matching:
  - A mission might require multiple sensors
- Missions have utility demands:
  - to be met with sensor utility
  - If demand is met, receive missions profit
- Or semi-matching with demands (SMD)
  - Assignment problem: each sensor goes to ≤ 1 mission

Related work

- Large literature on sensor coverage
  - Our focus: *assignment*

- Generalized Assignment Problem (GAP)
  - Generalization of Multiple Knapsack
  - No demand lower bounds, but we use as subroutine...

- Combinatorial Auctions: "Winner Determination Problem"
  - Generalization of our (static) problem
  - Very general, very hard

- Our (ITA project's) previous work
  - SMD: Max-profit assignments, no costs
  - SUM: GAP-like assignment problem (see poster tonight)
  - NUM: maximizing network utility, for predefined sensor-mission assignments

Related work

- Large literature on sensor coverage
  - Our focus: *assignment*
- Generalized Assignment Problem (GAP)
  - Generalization of Multiple Knapsack
  - No demand lower bounds, but we use as subroutine...
- Combinatorial Auctions: "Winner Determination Problem"
  - Generalization of our (static) problem
  - Very general, very hard
- Our (ITA project's) previous work
  - SMD: Max-profit assignments, no costs
  - SUM: GAP-like assignment problem (see poster tonight)
  - NUM: maximizing network utility, for predefined sensor-mission assignments

Related work

- Large literature on sensor coverage
  - Our focus: *assignment*
- Generalized Assignment Problem (GAP)
  - Generalization of Multiple Knapsack
  - No demand lower bounds, but we use as subroutine...
- Combinatorial Auctions: "Winner Determination Problem"
  - Generalization of our (static) problem
  - Very general, very hard
- Our (ITA project's) previous work
  - SMD: Max-profit assignments, no costs
  - SUM: GAP-like assignment problem (see poster tonight)
  - NUM: maximizing network utility, for predefined sensor-mission assignments

Related work

- Large literature on sensor coverage
  - Our focus: *assignment*
- Generalized Assignment Problem (GAP)
  - Generalization of Multiple Knapsack
  - No demand lower bounds, but we use as subroutine...
- Combinatorial Auctions: "Winner Determination Problem"
  - Generalization of our (static) problem
  - Very general, very hard
- Our (ITA project's) previous work
  - SMD: Max-profit assignments, no costs
  - SUM: GAP-like assignment problem (see poster tonight)
  - NUM: maximizing network utility, for predefined sensor-mission assignments

Motivation beyond SMD: frugality

- Static setting: network may be shared between multiple users. Each user can control the network for some time.
  - Single mission may not overtax the system
  - $\longrightarrow$ missions are given explicit *budgets*

- Dynamic setting: one user controls the network for its entire duration
  - Assume sensors have finite batteries, missions use energy
  - Now no budgets necessary, *just be rational*
    - Maximize total profits, over entire lifetime
  - Two subcases: operational lifetime known / not

Motivation beyond SMD: frugality

- Static setting: network may be shared between multiple users. Each user can control the network for some time.
  - Single mission may not overtax the system

  $\longrightarrow$ missions are given explicit *budgets*
- Dynamic setting: one user controls the network for its entire duration
  - Assume sensors have finite batteries, missions use energy
  - Now no budgets necessary, *just be rational*
    - Maximize total profits, over entire lifetime
  - Two subcases: operational lifetime known / not

Outline

1. Motivation
2. Static problem
   - Problem and algorithms
   - Performance evaluation
3. Dynamic problem
   - Problem and algorithms
   - Performance evaluation
4. Conclusion

Offline problem definition

- Similar to previous problem (SMD) but with 2 added factors:
  - Each mission has demand, profit and budget (b_j)
  - Sensors have associated costs (c_{ij})
- The problem is modeled with the following program:

$$\begin{array}{ll} \textit{max:} & \sum_{j=1}^{m} p_j(y_j) \\ \textit{s.t.:} & \sum_{i=1}^{n} x_{ij} e_{ij} \geq d_j y_j, \text{ for each } M_j, \\ & \sum_{i=1}^{n} x_{ij} c_{ij} \leq b_j, \text{ for each } M_j, \\ & \sum_{j=1}^{m} x_{ij} \leq 1, \text{ for each } S_i, \\ & x_{ij} \in \{0,1\}\ \forall x_{ij} \text{ and} \\ & y_j \in [0,1]\ \forall y_j \end{array}$$

Figure: MP **P** for static setting

- Goal: maximize profit without exceeding budget
- Problem is NP-complete, hard to approximate in general

Offline problem definition

- Similar to previous problem (SMD) but with 2 added factors:
  - Each mission has demand, profit and budget (b_j)
  - Sensors have associated costs (c_{ij})
- The problem is modeled with the following program:

$$
\boxed{
\begin{array}{ll}
max: & \sum_{j=1}^{m} p_j(y_j) \\
s.t.: & \sum_{i=1}^{n} x_{ij} e_{ij} \geq d_j y_j, \text{ for each } M_j, \\
& \sum_{i=1}^{n} x_{ij} c_{ij} \leq b_j, \text{ for each } M_j, \\
& \sum_{j=1}^{m} x_{ij} \leq 1, \text{ for each } S_i, \\
& x_{ij} \in \{0,1\} \ \forall x_{ij} \text{ and} \\
& y_j \in [0,1] \ \forall y_j
\end{array}
}
$$

Figure: MP **P** for static setting

- Goal: maximize profit without exceeding budget
- Problem is NP-complete, hard to approximate in general

Offline problem definition

- Similar to previous problem (SMD) but with 2 added factors:
  - Each mission has demand, profit and budget (b_j)
  - Sensors have associated costs (c_{ij})
- The problem is modeled with the following program:

$$
\begin{array}{ll}
\mathit{max:} & \sum_{j=1}^{m} p_j(y_j) \\
\mathit{s.t.:} & \sum_{i=1}^{n} x_{ij} e_{ij} \geq d_j y_j, \text{ for each } M_j, \\
& \sum_{i=1}^{n} x_{ij} c_{ij} \leq b_j, \text{ for each } M_j, \\
& \sum_{j=1}^{m} x_{ij} \leq 1, \text{ for each } S_i, \\
& x_{ij} \in \{0,1\} \ \forall x_{ij} \text{ and} \\
& y_j \in [0,1] \ \forall y_j
\end{array}
$$

Figure: MP **P** for static setting

- Goal: maximize profit without exceeding budget
- Problem is NP-complete, hard to approximate in general

Offline problem definition

- Similar to previous problem (SMD) but with 2 added factors:
  - Each mission has demand, profit and budget (b_j)
  - Sensors have associated costs (c_{ij})
- The problem is modeled with the following program:

$$\boxed{\begin{array}{ll} max: & \sum_{j=1}^{m} p_j(y_j) \\ s.t.: & \sum_{i=1}^{n} x_{ij} e_{ij} \geq d_j y_j, \text{ for each } M_j, \\ & \sum_{i=1}^{n} x_{ij} c_{ij} \leq b_j, \text{ for each } M_j, \\ & \sum_{j=1}^{m} x_{ij} \leq 1, \text{ for each } S_i, \\ & x_{ij} \in \{0,1\}\ \forall x_{ij} \text{ and} \\ & y_j \in [0,1]\ \forall y_j \end{array}}$$

Figure: MP **P** for static setting

- Goal: maximize profit without exceeding budget
- Problem is NP-complete, hard to approximate in general

Offline problem definition

- Similar to previous problem (SMD) but with 2 added factors:
  - Each mission has demand, profit and budget (b_j)
  - Sensors have associated costs (c_{ij})
- The problem is modeled with the following program:

$$
\begin{array}{ll}
\textit{max:} & \sum_{j=1}^{m} p_j(y_j) \\
\textit{s.t.:} & \sum_{i=1}^{n} x_{ij} e_{ij} \geq d_j y_j, \text{ for each } M_j, \\
 & \sum_{i=1}^{n} x_{ij} c_{ij} \leq b_j, \text{ for each } M_j, \\
 & \sum_{j=1}^{m} x_{ij} \leq 1, \text{ for each } S_i, \\
 & x_{ij} \in \{0,1\} \; \forall x_{ij} \text{ and} \\
 & y_j \in [0,1] \; \forall y_j
\end{array}
$$

Figure: MP **P** for static setting

- Goal: maximize profit without exceeding budget
- Problem is NP-complete, hard to approximate in general

Offline problem definition

- Similar to previous problem (SMD) but with 2 added factors:
  - Each mission has demand, profit and budget (b_j)
  - Sensors have associated costs (c_{ij})
- The problem is modeled with the following program:

$$\boxed{\begin{array}{ll} max: & \sum_{j=1}^{m} p_j(y_j) \\ s.t.: & \color{red}{\sum_{i=1}^{n} x_{ij} e_{ij} \geq d_j y_j, \text{ for each } M_j,} \\ & \sum_{i=1}^{n} x_{ij} c_{ij} \leq b_j, \text{ for each } M_j, \\ & \sum_{j=1}^{m} x_{ij} \leq 1, \text{ for each } S_i, \\ & x_{ij} \in \{0,1\}\ \forall x_{ij} \text{ and} \\ & \color{red}{y_j \in [0,1]\ \forall y_j} \end{array}}$$

Figure: MP **P** for static setting

- Goal: maximize profit without exceeding budget
- Problem is NP-complete, hard to approximate in general

Offline problem definition

- Similar to previous problem (SMD) but with 2 added factors:
  - Each mission has demand, profit and budget (b_j)
  - Sensors have associated costs (c_{ij})
- The problem is modeled with the following program:

$$
\boxed{
\begin{array}{ll}
\textit{max:} & \color{red}{\sum_{j=1}^{m} p_j(y_j)} \\
\textit{s.t.:} & \sum_{i=1}^{n} x_{ij} e_{ij} \geq d_j y_j, \text{ for each } M_j, \\
 & \sum_{i=1}^{n} x_{ij} c_{ij} \leq b_j, \text{ for each } M_j, \\
 & \sum_{j=1}^{m} x_{ij} \leq 1, \text{ for each } S_i, \\
 & x_{ij} \in \{0,1\}\ \forall x_{ij} \text{ and} \\
 & y_j \in [0,1]\ \forall y_j
\end{array}
}
$$

Figure: MP **P** for static setting

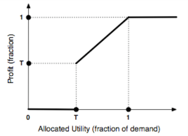
- Goal: maximize profit without exceeding budget
- Problem is NP-complete, hard to approximate in general

Profit function

- A mission's profit is based on amount of utility received
- Simple version: all-or-nothing profit, based on demand (SMD)
- Generalization: no profit until reach threshold, then fractional profit

Profit function

$$p_j(u_j) = \begin{cases} p_j, & \text{if } u_j \geq d_j \\ p_j \cdot u_j/d_j, & \text{if } T \leq u_j/d_j \\ 0, & \text{otherwise} \end{cases}$$

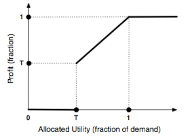


Profit function

- A mission's profit is based on amount of utility received
- Simple version: all-or-nothing profit, based on demand (SMD)
- Generalization: no profit until reach threshold, then fractional profit

Profit function

$$p_j(u_j) = \begin{cases} p_j, & \text{if } u_j \geq d_j \\ p_j \cdot u_j/d_j, & \text{if } T \leq u_j/d_j \\ 0, & \text{otherwise} \end{cases}$$



Special cases

Various special cases or relaxations may be easier to solve...

- Fully-fractional relaxation
  - Fractional profits *and* fractional profits
  - Solvable by LP (also provides upperbound on OPT)
- 1-d special case
  - Sensors & missions lie on a 1-d line
    - E.g., national border, coastline
  - Other simplifying assumptions:
    - Fractional profits, profit $=$ demand
    - Contributions are $0/1$ based on distance
    - Budgets *are* supported
  - In this case, can solve optimally by DP
    - Extension of knapsack DP, but (non-pseudo) poly-time

Special cases

Various special cases or relaxations may be easier to solve...

- Fully-fractional relaxation
  - Fractional profits *and* fractional profits
  - Solvable by LP (also provides upperbound on OPT)

- 1-d special case
  - Sensors & missions lie on a 1-d line
    - E.g., national border, coastline
  - Other simplifying assumptions:
    - Fractional profits, profit = demand
    - Contributions are 0/1 based on distance
    - Budgets *are* supported
  - In this case, can solve optimally by DP
    - Extension of knapsack DP, but (non-pseudo) poly-time

Special cases

Various special cases or relaxations may be easier to solve...

- Fully-fractional relaxation
  - Fractional profits *and* fractional profits
  - Solvable by LP (also provides upperbound on OPT)

- 1-d special case
  - Sensors & missions lie on a 1-d line
    - E.g., national border, coastline
  - Other simplifying assumptions:
    - Fractional profits, profit = demand
    - Contributions are 0/1 based on distance
    - Budgets *are* supported
  - In this case, can solve optimally by DP
    - Extension of knapsack DP, but (non-pseudo) poly-time

Special cases

Various special cases or relaxations may be easier to solve...

- Fully-fractional relaxation
  - Fractional profits *and* fractional profits
  - Solvable by LP (also provides upperbound on OPT)

- 1-d special case
  - Sensors & missions lie on a 1-d line
    - E.g., national border, coastline
  - Other simplifying assumptions:
    - Fractional profits, profit = demand
    - Contributions are 0/1 based on distance
    - Budgets *are* supported
  - In this case, can solve optimally by DP
    - Extension of knapsack DP, but (non-pseudo) poly-time

Special cases

Various special cases or relaxations may be easier to solve...

- Fully-fractional relaxation
  - Fractional profits *and* fractional profits
  - Solvable by LP (also provides upperbound on OPT)
- 1-d special case
  - Sensors & missions lie on a 1-d line
    - E.g., national border, coastline
  - Other simplifying assumptions:
    - Fractional profits, profit = demand
    - Contributions are 0/1 based on distance
    - Budgets *are* supported
  - In this case, can solve optimally by DP
    - Extension of knapsack DP, but (non-pseudo) poly-time

General setting: greedy algorithm

- Take missions in order of profit/demand
  1. Assign sensors ordered by utility/cost
  2. Stop if mission is satisfied or budget is spent
  3. If mission does not reach threshold release all sensors

- each mission $\approx$ knapsack
  - Alg is doubly-greedy...

Multi-Round GAP (MRGAP)

- Missions as knapsacks, sensors with costs/contributions varying by mission → Generalized Assignment Problem (GAP)
- GAP has approx algs, but trouble:
  - we require minimum threshold for profit

→ Iterative alg, with GAP alg as subroutine [Cohen et al. 2006]

- Raise threshold from 0 to real threshold
- Each round, eliminate missions that fail (current) threshold

- This scheme can be implemented in a (semi-)distributed fashion

Multi-Round GAP (MRGAP)

- Missions as knapsacks, sensors with costs/contributions varying by mission → Generalized Assignment Problem (GAP)
- GAP has approx algs, but trouble:
  - we require minimum threshold for profit

→ Iterative alg, with GAP alg as subroutine [Cohen et al. 2006]

- Raise threshold from 0 to real threshold
- Each round, eliminate missions that fail (current) threshold

- This scheme can be implemented in a (semi-)distributed fashion

Multi-Round GAP (MRGAP)

- Missions as knapsacks, sensors with costs/contributions varying by mission → Generalized Assignment Problem (GAP)
- GAP has approx algs, but trouble:
  - we require minimum threshold for profit

→ Iterative alg, with GAP alg as subroutine [Cohen et al. 2006]

- Raise threshold from 0 to real threshold
- Each round, eliminate missions that fail (current) threshold

- This scheme can be implemented in a (semi-)distributed fashion

Experimental setup

- Implemented a simulator in Java for testing
- Success threshold is set to 50%
- Field size = 400m x 400m

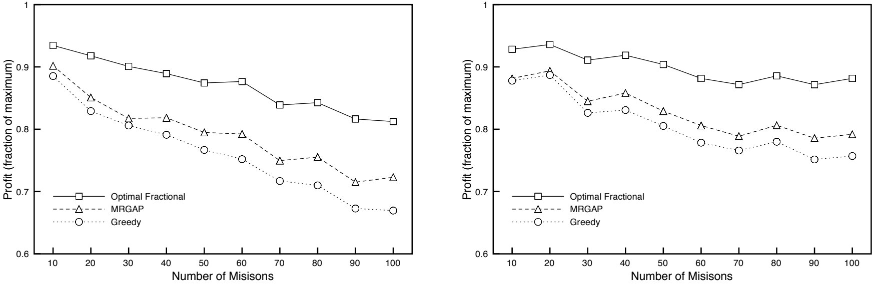
Utility of sensor to mission based on separating distance

$$e_{ij} = \begin{cases} \frac{1}{1+D_{ij}^2/c}, & \text{if } D_{ij} \leq r \\ 0, & \text{otherwise} \end{cases}$$

Experimental setup

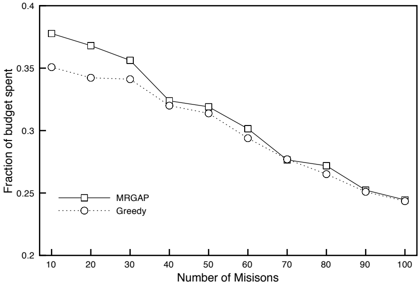
- Sensing Range (SR) = 30m, c = 60m
- Mission demands exply distributed, avg = 2, min = 0.5
  - drop unsatisfiable missions
- Mission profits exply distributed, avg = 10, max = 100
- Sensor costs uniformly distributed in the range [0,1]
- Mission budget uniform distributed in the range [0,6]

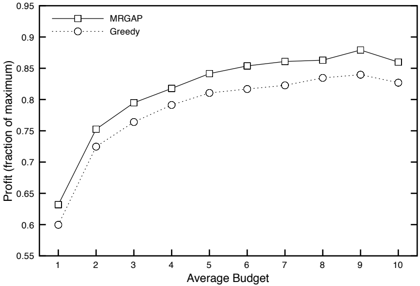
Static results: achieved profits



- Optimal Fractional: solve the LP that allows fractional sensor assignments and ignores threshold
- MRGAP achieves higher profits than Greedy
- Difference grows as #missions increases

Static results: budgets





- MRGAP: higher profits than Greedy, only slightly higher cost
- Fraction of spent budget *decreases* as #missions grows
  - Many missions → not enough sensors to exhaust budget

Outline

1. Motivation
2. Static problem
   - Problem and algorithms
   - Performance evaluation
3. Dynamic problem
   - Problem and algorithms
   - Performance evaluation
4. Conclusion

Dynamic/online setting

- Sensors have limited lifetime (battery)
- The dynamic problem can be modeled as follows:

$$\begin{array}{ll} \textit{max:} & \sum_t \sum_{j=1}^{m} p_j(y_{jt}) \\ \textit{s.t.:} & \sum_{i=1}^{n} x_{ijt} e_{ij} \geq d_j y_{jt}, \text{ for each } M_j \text{ and } t, \\ & \sum_{j=1}^{m} x_{ijt} \leq 1, \text{ for each } S_i \text{ and time } t, \\ & \sum_t \sum_{j=1}^{m} x_{ijt} \leq B, \text{ for each } S_i, \\ & x_{ijt} \in \{0,1\} \ \forall x_{ijt} \text{ and} \\ & y_{jt} \in [0,1] \ \forall y_{jt} \end{array}$$

Figure: MP **P'** for dynamic setting

- Goal: Maximize total lifetime profits
- Without knowing the future
- Also NP-complete, no competitive alg...

Dynamic/online setting

- Sensors have limited lifetime (battery)
- The dynamic problem can be modeled as follows:

$$\begin{array}{|ll|}\hline
\textit{max:} & \sum_t \sum_{j=1}^m p_j(y_{jt}) \\
\textit{s.t.:} & \sum_{i=1}^n x_{ijt} e_{ij} \geq d_j y_{jt}, \text{ for each } M_j \text{ and } t, \\
& \sum_{j=1}^m x_{ijt} \leq 1, \text{ for each } S_i \text{ and time } t, \\
& \sum_t \sum_{j=1}^m x_{ijt} \leq B, \text{ for each } S_i, \\
& x_{ijt} \in \{0,1\}\ \forall x_{ijt} \text{ and} \\
& y_{jt} \in [0,1]\ \forall y_{jt} \\ \hline
\end{array}$$

Figure: MP **P'** for dynamic setting

- Goal: Maximize total lifetime profits
- Without knowing the future
- Also NP-complete, no competitive alg...

Dynamic/online setting

- Sensors have limited lifetime (battery)
- The dynamic problem can be modeled as follows:

$$
\boxed{
\begin{array}{ll}
\textit{max:} & \sum_t \sum_{j=1}^{m} p_j(y_{jt}) \\
\textit{s.t.:} & \sum_{i=1}^{n} x_{ijt} e_{ij} \geq d_j y_{jt}, \text{ for each } M_j \text{ and } t, \\
 & \sum_{j=1}^{m} x_{ijt} \leq 1, \text{ for each } S_i \text{ and time } t, \\
 & \sum_t \sum_{j=1}^{m} x_{ijt} \leq B, \text{ for each } S_i, \\
 & x_{ijt} \in \{0,1\} \ \forall x_{ijt} \text{ and} \\
 & y_{jt} \in [0,1] \ \forall y_{jt}
\end{array}
}
$$

Figure: MP **P'** for dynamic setting

- Goal: Maximize total lifetime profits
- Without knowing the future
- Also NP-complete, no competitive alg...

Dynamic/online setting

- Sensors have limited lifetime (battery)
- The dynamic problem can be modeled as follows:

$$
\begin{array}{ll}
\textit{max:} & \sum_t \sum_{j=1}^{m} p_j(y_{jt}) \\
\textit{s.t.:} & \sum_{i=1}^{n} x_{ijt} e_{ij} \geq d_j y_{jt}, \text{ for each } M_j \text{ and } t, \\
 & \sum_{j=1}^{m} x_{ijt} \leq 1, \text{ for each } S_i \text{ and time } t, \\
 & \sum_t \sum_{j=1}^{m} x_{ijt} \leq B, \text{ for each } S_i, \\
 & x_{ijt} \in \{0,1\}\ \forall x_{ijt} \text{ and} \\
 & y_{jt} \in [0,1]\ \forall y_{jt}
\end{array}
$$

Figure: MP **P'** for dynamic setting

- Goal: Maximize total lifetime profits
- Without knowing the future
- Also NP-complete, no competitive alg...

Dynamic/online setting

- Sensors have limited lifetime (battery)
- The dynamic problem can be modeled as follows:

$$\begin{array}{ll} \textit{max:} & \sum_t \sum_{j=1}^{m} p_j(y_{jt}) \\ \textit{s.t.:} & \sum_{i=1}^{n} x_{ijt} e_{ij} \geq d_j y_{jt}, \text{ for each } M_j \text{ and } t, \\ & \sum_{j=1}^{m} x_{ijt} \leq 1, \text{ for each } S_i \text{ and time } t, \\ & \sum_t \sum_{j=1}^{m} x_{ijt} \leq B, \text{ for each } S_i, \\ & x_{ijt} \in \{0,1\}\ \forall x_{ijt} \text{ and} \\ & y_{jt} \in [0,1]\ \forall y_{jt} \end{array}$$

Figure: MP **P'** for dynamic setting

- Goal: Maximize total lifetime profits
- Without knowing the future
- Also NP-complete, no competitive alg...

Dynamic/online setting

- Sensors have limited lifetime (battery)
- The dynamic problem can be modeled as follows:

$$
\begin{array}{ll}
\textit{max:} & \sum_t \sum_{j=1}^{m} p_j(y_{jt}) \\
\textit{s.t.:} & \sum_{i=1}^{n} x_{ijt} e_{ij} \geq d_j y_{jt}, \text{ for each } M_j \text{ and } t, \\
 & \sum_{j=1}^{m} x_{ijt} \leq 1, \text{ for each } S_i \text{ and time } t, \\
 & \sum_t \sum_{j=1}^{m} x_{ijt} \leq B, \text{ for each } S_i, \\
 & x_{ijt} \in \{0,1\} \; \forall x_{ijt} \text{ and} \\
 & y_{jt} \in [0,1] \; \forall y_{jt}
\end{array}
$$

Figure: MP **P'** for dynamic setting

- Goal: Maximize total lifetime profits
- Without knowing the future
- Also NP-complete, no competitive alg...

Dynamic/online setting

- Sensors have limited lifetime (battery)
- The dynamic problem can be modeled as follows:

$$
\begin{array}{ll}
\textit{max:} & \sum_t \sum_{j=1}^{m} p_j(y_{jt}) \\
\textit{s.t.:} & \sum_{i=1}^{n} x_{ijt} e_{ij} \geq d_j y_{jt}, \text{ for each } M_j \text{ and } t, \\
 & \sum_{j=1}^{m} x_{ijt} \leq 1, \text{ for each } S_i \text{ and time } t, \\
 & \sum_t \sum_{j=1}^{m} x_{ijt} \leq B, \text{ for each } S_i, \\
 & x_{ijt} \in \{0,1\} \ \forall x_{ijt} \text{ and} \\
 & y_{jt} \in [0,1] \ \forall y_{jt}
\end{array}
$$

Figure: MP **P'** for dynamic setting

- Goal: Maximize total lifetime profits
- Without knowing the future
- Also NP-complete, no competitive alg...

0/1 special case

- Again assume restrictions:
  - Fractional profits
  - 0/1 contributions
  - Free preemption
- Then can get a .63-approximation (competitive)
- Reduction to recent multi-slot Adwords alg (Buchbinder et al.)
  - sensor ≈ advertiser
  - battery life ≈ advertiser budget
  - timestep ≈ adword

General setting: semi-online approach

- Maximize profit taking sensor energy into account
- Two cases:
  - Network lifetime is unknown (maximize profit and lifetime)
    - Energy-aware scheme
  - Network operational lifetime is known (e.g. 1 week)
    - Energy and lifetime-aware scheme
- If distributions of mission properties are known
  - Estimate expected effective profit a sensor can provide a *typical* mission:

$$\hat{P} = E\left[\frac{u}{d}\right] \times \frac{E[p]}{P} \quad (1)$$

General setting: semi-online approach

- Maximize profit taking sensor energy into account
- Two cases:
  - Network lifetime is unknown (maximize profit and lifetime)
    - Energy-aware scheme
  - Network operational lifetime is known (e.g. 1 week)
    - Energy and lifetime-aware scheme

- If distributions of mission properties are known
  - Estimate expected effective profit a sensor can provide a *typical* mission:

$$\hat{P} = E\left[\frac{u}{d}\right] \times \frac{E[p]}{P} \quad (1)$$

General setting: semi-online approach

- Maximize profit taking sensor energy into account
- Two cases:
  - Network lifetime is unknown (maximize profit and lifetime)
    - Energy-aware scheme
  - Network operational lifetime is known (e.g. 1 week)
    - Energy and lifetime-aware scheme
- If distributions of mission properties are known
  - Estimate expected effective profit a sensor can provide a *typical* mission:

$$\hat{P} = E\left[\frac{u}{d}\right] \times \frac{E[p]}{P} \quad (1)$$

General setting: semi-online approach

- Maximize profit taking sensor energy into account
- Two cases:
  - Network lifetime is unknown (maximize profit and lifetime)
    - Energy-aware scheme
  - Network operational lifetime is known (e.g. 1 week)
    - Energy and lifetime-aware scheme

- If distributions of mission properties are known
  - Estimate expected effective profit a sensor can provide a *typical* mission:

$$\hat{P} = E\left[\frac{u}{d}\right] \times \frac{E[p]}{P} \tag{1}$$

General setting: semi-online approach

- Maximize profit taking sensor energy into account
- Two cases:
  - Network lifetime is unknown (maximize profit and lifetime)
    - Energy-aware scheme
  - Network operational lifetime is known (e.g. 1 week)
    - Energy and lifetime-aware scheme

- If distributions of mission properties are known
  - Estimate expected effective profit a sensor can provide a *typical* mission:

$$\hat{P} = E\left[\frac{u}{d}\right] \times \frac{E[p]}{P} \tag{1}$$

Energy-aware scheme

- Initially, sensors accept any mission with pretty good profit
- Energy falls $\rightarrow$ grow more conservative
- A sensor computes "eagerness" to serve a *particular* mission:

$$P^* = \frac{u}{d} \times \frac{p}{P} \times f \tag{2}$$

where f is the fraction of remaining energy

- Propose if high enough ($P^* \geq \hat{P}$)

$\rightarrow$ Mission leaders selects sensors based on the offered utility

Energy-aware scheme

- Initially, sensors accept any mission with pretty good profit
- Energy falls $\rightarrow$ grow more conservative
- A sensor computes "eagerness" to serve a *particular* mission:

$$P^* = \frac{u}{d} \times \frac{p}{P} \times f \tag{2}$$

where f is the fraction of remaining energy

- Propose if high enough ($P^* \geq \hat{P}$)

$\rightarrow$ Mission leaders selects sensors based on the offered utility

Energy-aware scheme

- Initially, sensors accept any mission with pretty good profit
- Energy falls $\longrightarrow$ grow more conservative
- A sensor computes "eagerness" to serve a *particular* mission:

$$P^* = \frac{u}{d} \times \frac{p}{P} \times f \tag{2}$$

  where f is the fraction of remaining energy
- Propose if high enough ($P^* \geq \hat{P}$)

$\longrightarrow$ Mission leaders selects sensors based on the offered utility

Energy/lifetime-aware scheme

More complicated incentives now...

- Initially eager to accept missions
- Energy falls → more conservative
- Approach end of network lifetime → more eager again

- Sensors determine which missions to propose to based on:
  - Expected sensor occupancy time
  - Remaining sensor operational time (based on residual energy)
  - Actual and expected mission profit contributions

Energy/lifetime-aware scheme

More complicated incentives now...

- Initially eager to accept missions
- Energy falls → more conservative
- Approach end of network lifetime → more eager again

- Sensors determine which missions to propose to based on:
  - Expected sensor occupancy time
  - Remaining sensor operational time (based on residual energy)
  - Actual and expected mission profit contributions

Energy/lifetime-aware scheme

- Sensors determine their expected occupancy time (α)

$$\alpha = \tau \times \lambda \times g \times \gamma \times E[l]$$

Remaining network lifetime

Mission arrival rate

Probability mission within range

Probability proposal accepted

Expected mission lifetime

- Sensors calculate the following (t_b = sensor's remaining lifetime):

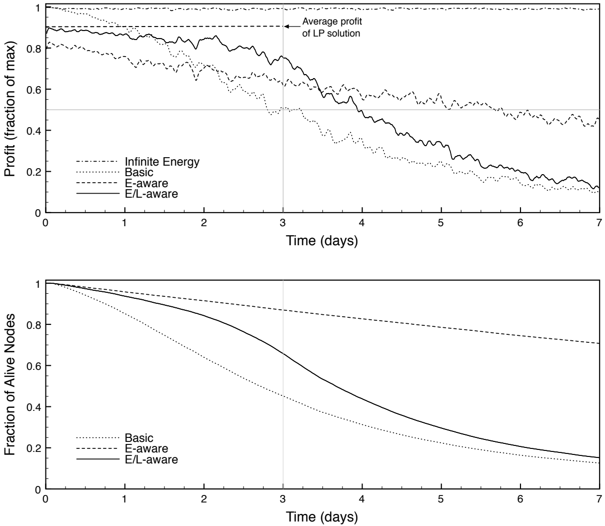
$$P^* = \frac{u}{d} \times \frac{p}{P} \times \frac{t_b}{\alpha} \qquad (3)$$

- Propose if high enough ($P^* \geq \hat{P}$)
- Mission leader selects sensors greedily based on utility

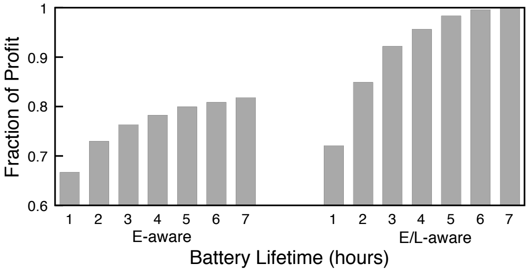
Experimental setup

- Network size = 500 nodes
- Sensors start with energy that can last for 2 hours of continuous sensing
    - Energy is only used for sensing
- Missions arrival times are Poisson distribution, with avg. = 4 or 8 missions/hour
- Mission lifetimes are exply distributed, with avg. = 1 hour, min = 5 minutes and max = 4 hours
- Network operational lifetime is 3 days

Dynamic results: $\lambda = 8$ missions/hr



Dynamic results: effects of battery sizes



- Profit is fraction of total in first 3 days
- Increasing battery lifetime has high effect in the beginning
- E/L-aware scheme uses energy more effectively because it takes both energy and lifetime into account

Outline

1. Motivation
2. Static problem
   - Problem and algorithms
   - Performance evaluation
3. Dynamic problem
   - Problem and algorithms
   - Performance evaluation
4. Conclusion

Conclusion

- We gave approximation (heuristic) algorithms for the static and dynamic problems.
- And stronger guarantees for some special cases.
- In our experiments, our algorithms appear to perform well.
- But the problem is still quite abstract...

Open problems and future work

- "Bundles": nonadditive utility models
- More realistic special cases / problems
- Algorithms based on geometric utilities?

Open problems and future work

- "Bundles": nonadditive utility models
- More realistic special cases / problems
- Algorithms based on geometric utilities?

Open problems and future work

- “Bundles”: nonadditive utility models
- More realistic special cases / problems
- Algorithms based on geometric utilities?

Thanks!

mpjohnson@gmail.com