Minimizing the Age of Information of Multiple Sources

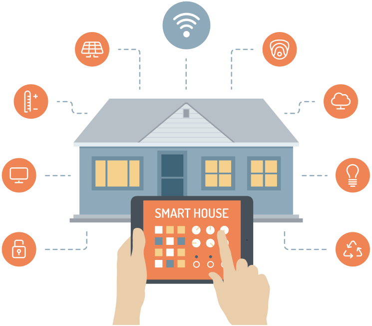
Vishrant Tripathi
Advisor - Prof. Sharayu Moharir

Electrical Engineering Department, IIT-Bombay

October 16, 2017

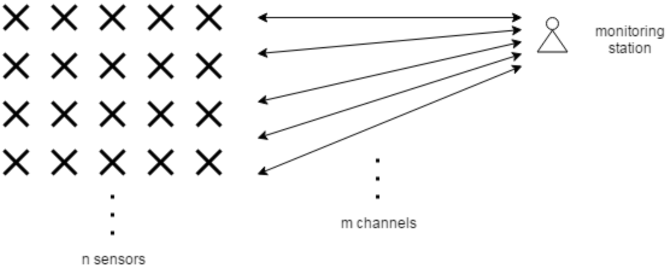
Motivation

- This problem has direct applications to IoT (Internet of Things).
- For example, an IoT connected home or vehicle could have a large number of sensors monitoring different pieces of information, all of which needs to be sent to a central controller



- Having the freshest available data would be essential to making better decisions.

Setting



- n sensors communicating over m channels
- Time-slotted system
- One channel per sensor per time-slot

Channel Model

Assumption

(ON-OFF Channel Model)

$$X_{i,j}(t) = \begin{cases} 1, & \textit{if sensor } i \textit{ can communicate over channel } j \\ 0, & \textit{otherwise} \end{cases}$$

We have that $\forall i, j$,

$$\mathbb{P}(X_{i,j}(t) = 1 | X_{i,j}(\tau) : \forall \tau < t, i, j) \geq p_{min} > 0.$$

The processes $X_{i,j}(t)$ *evolve independently across all sensor-channel pairs.*

Channel Model

Examples of such channel models include:

- $X_{i,j}(t)$ is an independent Bernoulli random variable with parameter $p_{i,j}(t) \geq p_{\min}$.
- $X_{i,j}$ is a Markov chain, independent across all users and channels with

$$\mathbb{P}(X_{i,j}(t) = 1 | X_{i,j}(t-1)) \geq p_{\min}.$$

Goal

- $l_i(t)$ is the age of the latest measurement from sensor i
- Our goal is to minimize the time-average cost of the age of information

$$C(t) = f(l_i(t); 1 \leq i \leq n),$$

where f is a non-decreasing function of the l_is

Examples:

- $f(l_i(t); 1 \leq i \leq n) = \sum_{i=1}^{n} g(l_i(t))$
- $f(l_i(t); 1 \leq i \leq n) = \max_i l_i(t)$

A Converse Result

$n_l(t)$: the number of sensors with age $\geq l$ at time t, then

$$n_l(t) \geq (n - lm)^+.$$

This can be proved using a simple counting argument.

Our Algorithm: Max-Age Matching

Key Ideas -

- Use a locally greedy strategy to minimize the age of information increment in each time-slot
- Convert this problem of greedy minimization to a minimum weight perfect matching problem in bipartite graphs

Max-Age Matching

Algorithm 1 Max-Age Matching

Input: Connectivity and age information for the current time-slot
Output: A valid allocation of sensors to channels

1: **procedure** $\textsc{Max-Age-Matching}(X_{i,j})$
2: Construct a bipartite graph $G(X, Y, E)$ using connectivity and age information.
3: $\text{M} = \textsc{FindMaxWeightMatching}(G)$
4: Use M to allocate sensors to channels
5: **end procedure**

Max-Age Matching Example

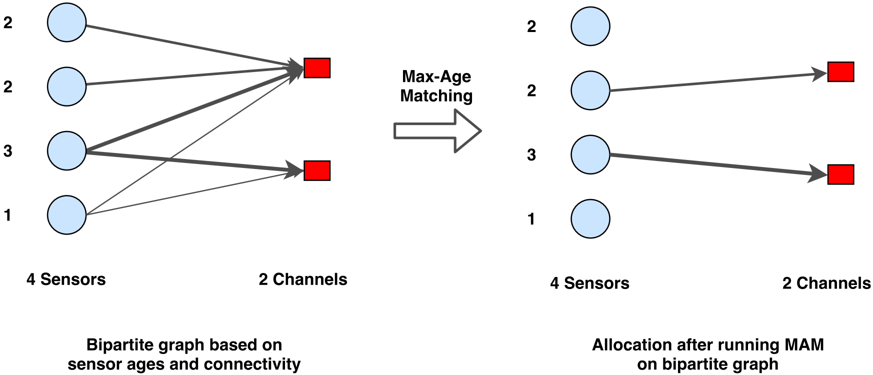


Figure: MAM Example with 4 sensors, 2 channels and sensor ages (2,2,1,3)

Optimality of Max-Age Matching

Theorem

$n_l^{MAM}(t)$: *the number of sensors with age* $\geq l$ *under MAM*

$$n_l^{MAM}(t) = (n - lm)^+ \ \forall l \geq 0, \text{ with probability } \geq 1 - 3m(1 - p_{min})^m \left\lceil \frac{n}{m} \right\rceil$$

Corollary

- $C^{MAM}(t)$: *the cost of the age of information under MAM*
- $C^{OPT}(t)$: *the cost under the optimal scheduling policy*

$$C^{MAM}(t) = C^{OPT}(t), \text{ with probability } \geq 1 - 3m(1 - p_{min})^m \left\lceil \frac{n}{m} \right\rceil$$

The probability $\rightarrow 1$ as $n, m \uparrow \infty$ if m grows at least as fast as $\Omega(\log(n))$.

Our 2^{nd} Algorithm: Iterative Max-Age Scheduling

Key Idea -

- Simply iterate through the entire set of channels, allocating sensors which can connect to a particular channel in descending order of age.
- The advantage here is that of reduced complexity. No other algorithm can be simpler, since this algorithm goes through all the inputs only once.

Iterative Max-Age Scheduling

Algorithm 2 Allocate Sensors to Channels in each Time-slot

Input: $X_{i,j}$ - the connectivity information for the current time-slot
Output: A valid allocation in each time-slot

```
procedure FindAllocation()
    Define a priority of sensors in decreasing order of costs g(l_i(t)).
    For sensors with equal costs, use lexicographic ordering.
    for every channel j do
        Find highest priority un-allocated sensor i s.t. X_{i,j} = 1
        Allocate sensor i to channel j
    end for
    Output the Allocation for the next time-slot.
end procedure
```

Iterative Max-Age Scheduling Example

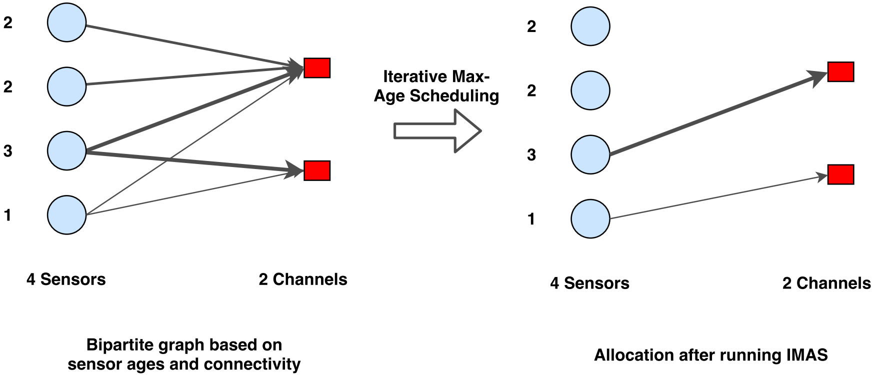


Figure: IMAS Example with 4 sensors, 2 channels and sensor ages (2,2,1,3)

Optimality of IMAS

Theorem

$n_l^{IMAS}(t)$: *the number of sensors with age* $\geq l$*, under IMAS*

$$n_l^{IMAS}(t) = \begin{cases} n - lm + O(\log m), & \textit{for } 0 \leq l < \lceil \frac{n}{m} \rceil, \\ O(\log m), & \textit{for } l = \lceil \frac{n}{m} \rceil, \\ 0, & \textit{for } l > \lceil \frac{n}{m} \rceil. \end{cases}$$

with high probability, which goes to 1 as $n, m \uparrow \infty$ *if* m *grows at least as fast as* $\Omega(\log(n))$.

Optimality of IMAS

Corollary

$C^{IMAS}(t)$: *the cost of the age of information under IMAS*

1. *If f is defined as a sum of individual costs of sensors, we have*

$$\frac{C^{IMAS}(t)}{C^{OPT}(t)} = 1 + O\left(\frac{\log(m)}{m}\right)$$

2. *If f is defined as the maximum of individual costs, then*

$$C^{IMAS}(t) = C^{OPT}(t) + 1$$

with high probability, which goes to 1 as $n, m \uparrow \infty$ *if m grows at least as fast as* $\Omega(\log(n))$.

Complexity Comparison

Now that we have compared the performance of the two proposed algorithms, we can also compare their computational costs.

- The complexity of Max-Age Matching (MAM) is $O(n^3)$.
- The complexity of Iterative Max-Age Scheduling (IMAS) is $O(mn)$.

Energy Performance Trade-offs

- The above scheduling algorithms require all sensors to be ON in each time-slot
- Instead, we can use a batch based version of the above algorithms to save energy.
- Clearly, the performance of these batch based will be worse off as compared to the above algorithms.
- We use a batch size of $\frac{n}{k}$ where $k = \left\lceil \frac{n}{m} \right\rceil$ so that in each batch we have roughly the same number of active sensors as the number of channels.

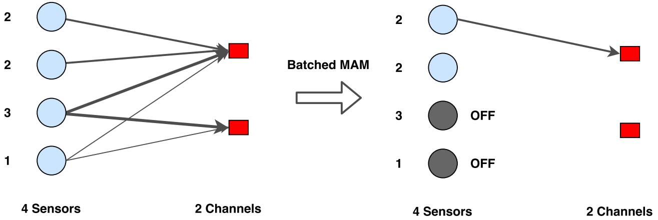
Batched Max-Age Matching

Algorithm 3 Batched Max-Age Matching

Input: Connectivity and age information for the batch being served in the current time-slot (serve batches in round robin fashion)
Output: A valid allocation of sensors to channels

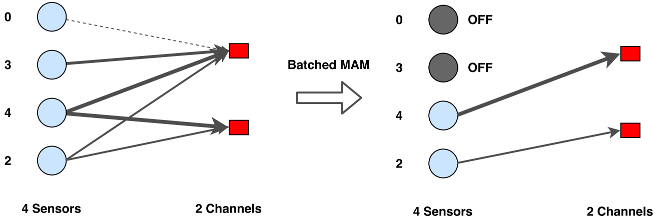
1: **procedure** MAX-AGE-MATCHING($X_{i,j}$)
2: Construct a bipartite graph $G(X, Y, E)$ as described earlier using connectivity and age information of the current batch.
3: M = FINDMAXWEIGHTMATCHING(G)
4: Use M to allocate sensors to channels
5: **end procedure**

B-MAM Example



Sensor ages and connectivity in first time-slot

Allocation after running B-MAM for first batch



Sensor ages and connectivity in second time-slot

Allocation after running B-MAM for second batch

Optimality of B-MAM

Theorem

$n_l^{B\text{-}MAM}(t)$: *the number of sensors with age* $\geq l$*, under B-MAM*

$$n_l^{MAM}(t) = (n - lm)^+$$

$\forall l \geq 0$, *with probability* $\geq 1 - 3n(1 - p_{min})^m$.

Corollary

$C^{B\text{-}MAM}(t)$: *the cost of the age of information under B-MAM*

$$C^{B\text{-}MAM}(t) = C^{OPT}(t), \text{ with probability } \geq 1 - 3n(1 - p_{min})^m.$$

Order wise identical with MAM

The probability $\to 1$ as $n, m \uparrow \infty$ if m grows at least as fast as $\Omega(\log(n))$.

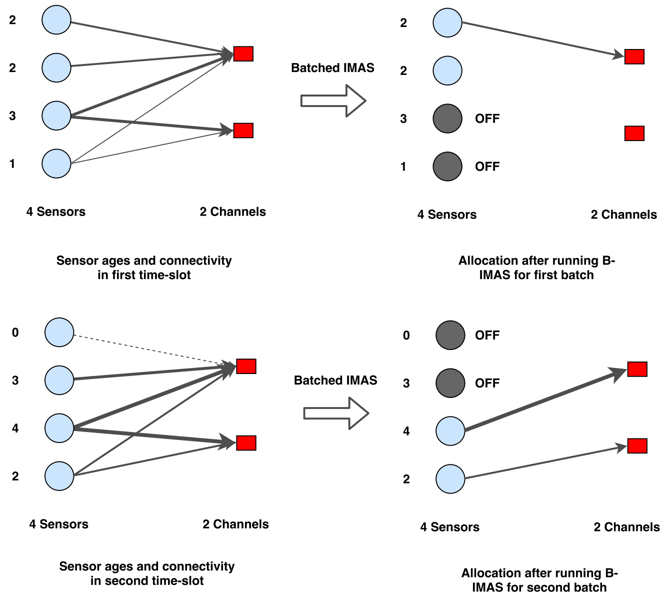
B-IMAS

Algorithm 4 Allocate Sensors to Channels in each Time-slot

Input: Connectivity and age information for the batch being served in the current time-slot (serve batches in round robin fashion)
Output: A valid allocation in each time-slot

```
procedure FindAllocation()
    Define a priority of sensors in decreasing order of costs g(l_i(t)).
    For sensors with equal costs, use lexicographic ordering.
    for every channel j in current batch do
        Find highest priority un-allocated sensor i s.t. X_{i,j} = 1
        Allocate sensor i to channel j
    end for
    Output the Allocation for the next time-slot.
end procedure
```

B-IMAS Example



Optimality of B-IMAS

Theorem

$n_l^{\text{B-IMAS}}(t)$: *the number of sensors with age* $\geq l$ *under B-IMAS*

$$n_l^{\text{B-IMAS}}(t) = \begin{cases} n - l(m - m^{\alpha}), \ \textit{for } 0 \leq l < \lceil \frac{n}{m} \rceil, \\ (2\lceil \frac{n}{m} \rceil - l)m^{\alpha}, \ \textit{for } \lceil \frac{n}{m} \rceil \leq l \leq 2\lceil \frac{n}{m} \rceil, \\ 0, \ \textit{for } l > 2\lceil \frac{n}{m} \rceil. \end{cases}$$

with probability
$\geq 1 - \lceil \frac{n}{m} \rceil (m^{\alpha}(1 - p_{min})^{m - m^{\alpha} + 1} + (m - m^{\alpha})(1 - p_{min})^{m^{\alpha} + 1})$ *where*
$0 < \alpha < 1$.

This probability $\rightarrow 1$ as $n, m \uparrow \infty$ if m grows at least as fast as $\Omega((\log n)^{1+a})$, where $\frac{1}{1+a} < \alpha$.

Optimality of B-IMAS

Corollary

$C^{B\text{-}IMAS}(t)$: *the cost of the age of information under B-IMAS*
$\delta = \left\lceil \frac{n}{m} \right\rceil (m^\alpha (1 - p_{min})^{m - m^\alpha + 1} + (m - m^\alpha)(1 - p_{min})^{m^\alpha + 1})$

(a) *If f is defined as sum of costs of individual sensors,*

$$\frac{C^{B\text{-}IMAS}(t)}{C^{OPT}(t)} = 1 + O(m^{\alpha - 1}).$$

(b) *If f is max of sensor ages,*

$$C^{IMAS}(t) = C^{OPT}(t) + \left\lceil \frac{n}{m} \right\rceil.$$

with probability $\geq 1 - \delta$

Empirical CCDF plots

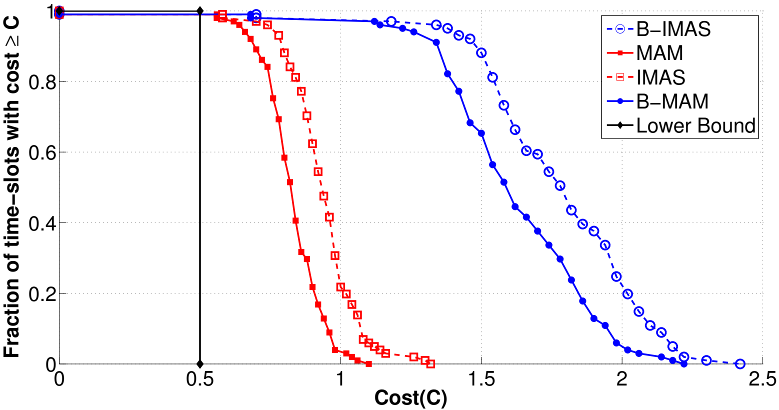


Figure: $n = 50$, $m = 25$, $p = 0.05$, cost type = average age

Empirical CCDF plots

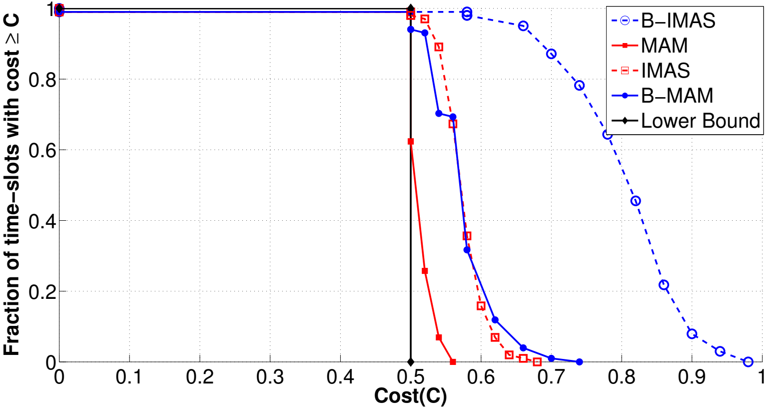


Figure: $n = 50,\ m = 25,\ p = 0.15$, cost type $=$ average age

Empirical CCDF plots

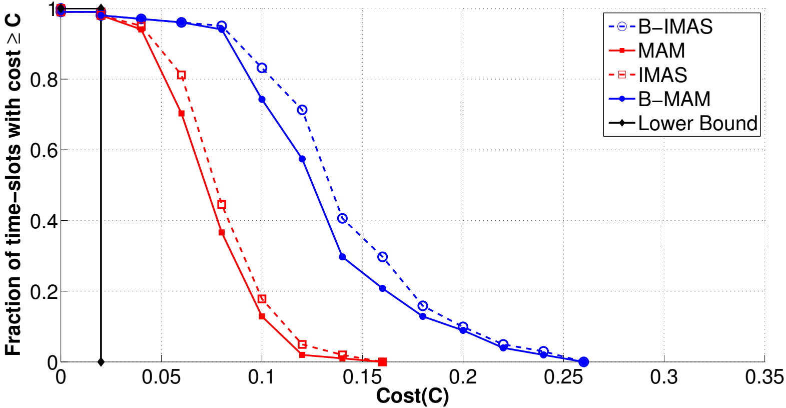


Figure: $n = 50$, $m = 25$, $p = 0.05$, cost type = max age

Empirical CCDF plots

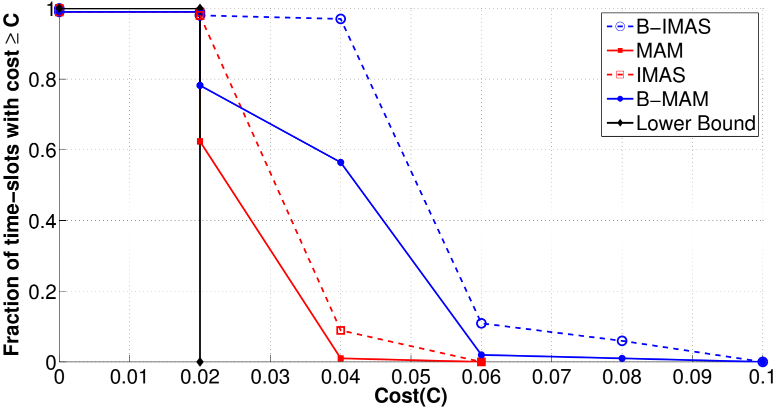


Figure: $n = 50$, $m = 25$, $p = 0.15$, cost type = max age

Time average cost v/s no. of sensors

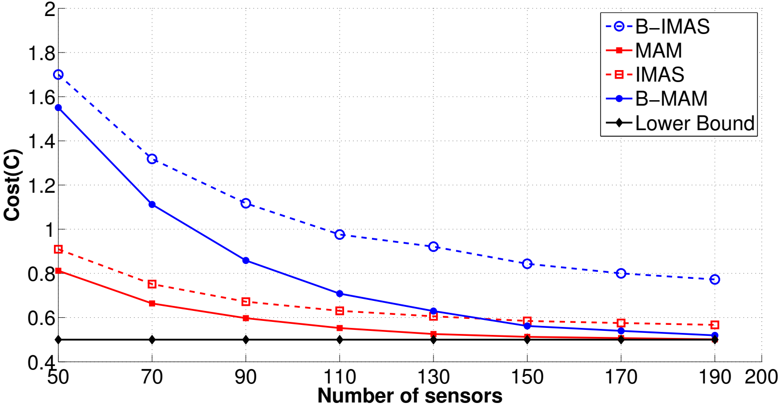


Figure: $n = 50$ to 190, $m = n/2$, $p = 0.05$, cost type = average age

Time average cost v/s no. of sensors

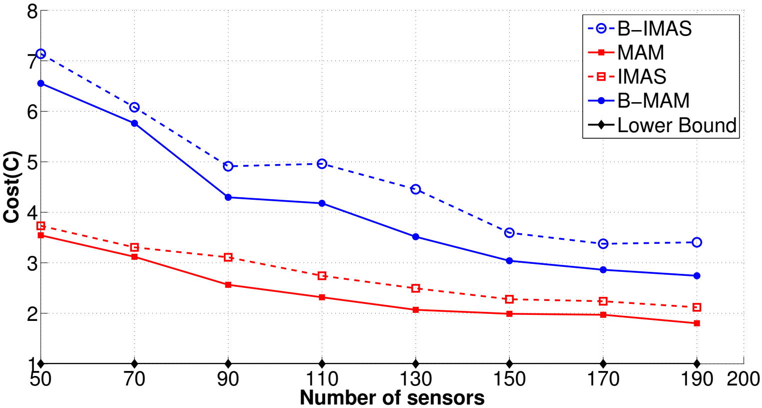


Figure: $n = 50$ to 190, $m = n/2$, $p = 0.05$, cost type = max age

Time average cost v/s no. of sensors

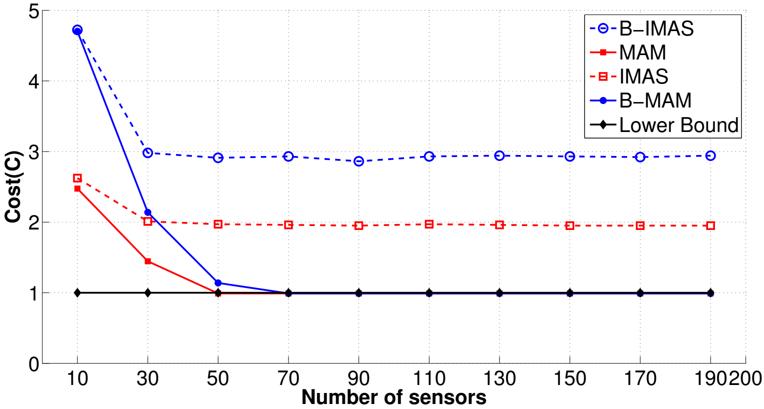


Figure: $n = 50$ to 190, $m = n/2$, $p = 0.25$, cost type = max age

Time average cost v/s connection probability

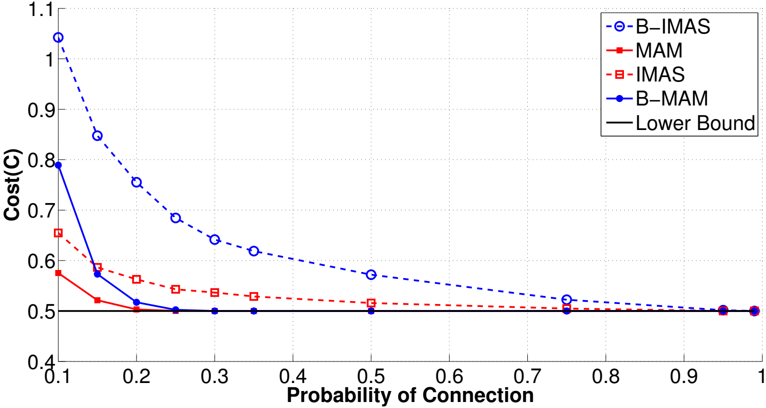


Figure: $n = 50$, $p = 0$ to 1, $m = 25$, cost type $=$ average age

Time average cost v/s connection probability

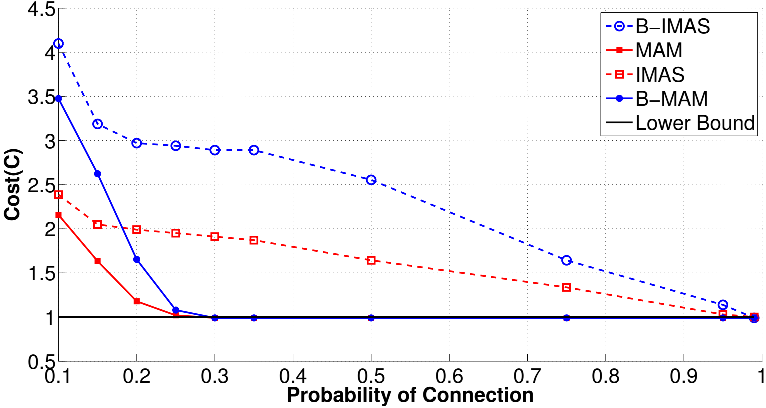


Figure: $n = 50$, $p = 0$ to 1, $m = 25$, cost type $=$ max age

Conclusions

We propose and analyze the optimality of two algorithms for solving the problem of allocating sensors to channels in a stochastic setting -

1. A perfect matching based computationally intensive approach and
2. An iterative approach that is cheaper to compute.
3. We also suggest two batched versions of the same algorithms, in cases when energy efficiency is an important parameter.

We then provide optimality results and compare the performances of all four algorithms through simulation examples.

Conclusions

Theoretical bounds and simulation examples suggest the following order of performance

- For small systems and/or bad channels -

$$C^{MAM} \leq C^{IMAS} \leq C^{B-MAM} \leq C^{B-IMAS}$$

- For large systems and/or good channels -

$$C^{MAM} \leq C^{B-MAM} \leq C^{IMAS} \leq C^{B-IMAS}$$

The End