A Knapsack Approach to Sensor-Mission Assignment with Uncertain Demands

Diego Pizzocaro\textsuperscript{a}  Matthew P. Johnson\textsuperscript{b}  Hosam Rowaihy\textsuperscript{c}
Stuart Chalmers\textsuperscript{d}  Alun Preece\textsuperscript{a}  Amotz Bar-Noy\textsuperscript{b}  Thomas La Porta\textsuperscript{c}

\textsuperscript{a} Cardiff University \hspace{1cm} \textsuperscript{b} The City University of New York \hspace{1cm} \textsuperscript{c} Pennsylvania State University
\textsuperscript{d} University of Aberdeen
- An **homogeneous sensor network** which is already deployed.

- It is usually required to support **multiple missions** to be accomplished simultaneously.

- Missions compete for the **exclusive usage of a sensor**: we need a scheme to assign individual sensors to missions.

- Missions have an **uncertain demand** for sensing resource capabilities (due to weather, unexpected events, ...)

Formal Model

- We model this assignment problem by introducing the **Sensor Utility Maximization** (SUM) model.

- SUM can be represented as a **bipartite graph**:
  
  - **Vertex sets:**
    - Sensors \(\{S_i\}\), Missions \(\{M_j\}\)
  
  - For each **mission** \((M_j)\):
    - \(d_j\) = utility demand
    - \(p_j\) = priority
  
  - For each **sensor-mission pair**:
    - \(e_{ij}\) = utility that \(S_i\) could contribute to \(M_j\)
    - \(p_{ij} = e_{ij} / d_j \times p_j\) profit of \(S_i\) for mission \(M_j\)
  
- **Goal:** a sensor assignment that **maximizes the total profit**.
Formal Model (contd)

• **Integer Linear Programming** (ILP) formulation:

Maximize:  \[
\sum_{j=1}^{m} \sum_{i=1}^{n} p_{ij} x_{ij}, \text{ where } p_{ij} = \frac{e_{ij}}{d_j} \times p_j.
\]

Such that:  \[
\sum_{i=1}^{n} x_{ij} e_{ij} \leq d_j, \text{ for each mission } M_j \in M,
\]
\[
\sum_{j=1}^{m} x_{ij} \leq 1, \text{ for each sensor } S_i, \text{ and}
\]
\[
x_{ij} \in \{0, 1\}, \text{ for each variable } x_{ij}
\]

• Most important constraint (knapsack-style):

**Total utility cumulated** by each mission \( M_j \leq \) its demand \( d_j \)

• This assumption is based:
  - on the **uncertainty** of the demands,
  - on the principle that **sensing resources are in high demand** and **should not be wasted**.

(from “JP 2-01 joint and national intelligence support to military operations”, see ref, in the paper)
Pre-existent Algorithms

- **Problem hardness:** it is **NP-Complete**
  - Proof by showing that SUM is a **generalization** of the **knapsack problem**

- We adapted **three pre-existent algorithms** to solve SUM:

  1. **Mission-side greedy**
     - Sort missions by decreasing priority ($p_j$)
     - **For each mission:** Assign sensors with highest utilities ($e_{ij}$) to the mission, until the cumulated utility does not exceed its demand ($d_{ij}$).

  2. **Sensor-side greedy**
     - Unsorted sensors
     - **For each sensor:** Assign the sensor to the mission where that sensor is of most use (the one which maximizes the profit of each sensor $p_{ij}$)

  3. **State-of-the-art approximation algorithm for GAP**
     - Proved that SUM is a **special case** of **Generalized Assignment Problem** (GAP)
     - Provide a very good guaranteed approximation of the optimal solution
Novel algorithm

- It is an **improved version of Sensor-side greedy:**

  - **Ordered Sensor-side greedy**
    
    - **IDEA:** Sorts sensors by decreasing max profit offer between all the missions to which the sensor can contribute ($\max\{p_{ij}\}$).
    
    - **Like basic Sensor-Side Greedy:** Assign sensor to the mission where that sensor is of most use (the one which maximizes the profit $p_{ij}$).
    
    - **Further improvement:** If a sensor cannot be assigned to the mission which maximize its profit, we consider the second most profitable mission, and so on ...
Novel algorithm

- It is an **improved version of Sensor-side greedy:**
  
  - **Ordered Sensor-side greedy**
    
    - **IDEA:** Sorts sensors by decreasing max profit offer between all the missions to which the sensor can contribute (\( \max\{p_{ij}\} \)).

  - Like **basic Sensor-Side Greedy**: Assign sensor to the mission where that sensor is of most use (the one which maximizes the profit \( p_{ij} \)).

  - Further improvement: If a sensor cannot be assigned to the mission which maximize its profit, we consider the second most profitable mission, and so on ...
Novel algorithm

- It is an improved version of Sensor-side greedy:
  - **Ordered Sensor-side greedy**
    - **IDEA:** Sorts sensors by decreasing max profit offer between all the missions to which the sensor can contribute (\( \max\{p_{ij}\} \)).

- **Like basic Sensor-Side Greedy:** Assign sensor to the mission where that sensor is of most use (the one which maximizes the profit \( p_{ij} \)).

- **Further improvement:** If a sensor cannot be assigned to the mission which maximize its profit, we consider the second most profitable mission, and so on ...
Novel algorithm

- It is an **improved version of Sensor-side greedy:**

  - **Ordered Sensor-side greedy**
    - **IDEA:** Sorts sensors by decreasing max profit offer between all the missions to which the sensor can contribute (\( \max\{p_{ij}\} \)).

- **Like basic Sensor-Side Greedy:** Assign sensor to the mission where that sensor is of most use (the one which maximizes the profit \( p_{ij} \)).

- **Further improvement:** If a sensor cannot be assigned to the mission which maximize its profit, we consider the second most profitable mission, and so on ...
Novel algorithm

- It is an **improved version of Sensor-side greedy:**

  - **Ordered Sensor-side greedy**
    - **IDEA:** Sorts sensors by decreasing max profit offer between all the missions to which the sensor can contribute (\(\max\{p_{ij}\}\)).

- **Like basic Sensor-Side Greedy:** Assign sensor to the mission where that sensor is of most use (the one which maximizes the profit \(p_{ij}\)).

- **Further improvement:** If a sensor cannot be assigned to the mission which maximize its profit, we consider the second most profitable mission, and so on ...
Novel algorithm

- It is an **improved version of Sensor-side greedy:**
  - **Ordered Sensor-side greedy**
    - **IDEA:** Sorts sensors by decreasing max profit offer between all the missions to which the sensor can contribute (max\{p_{ij}\}).

- **Like basic Sensor-Side Greedy:** Assign sensor to the mission where that sensor is of most use (the one which maximizes the profit p_{ij}).

- **Further improvement:** If a sensor cannot be assigned to the mission which maximize its profit, we consider the second most profitable mission, and so on ...
Novel algorithm

- It is an **improved version of Sensor-side greedy:**

  - **Ordered Sensor-side greedy**

    - **IDEA:** Sorts sensors by decreasing max profit offer between all the missions to which the sensor can contribute (\( \max\{p_{ij}\} \)).

    - **Like basic Sensor-Side Greedy:** Assign sensor to the mission where that sensor is of most use (the one which maximizes the profit \( p_{ij} \)).

    - **Further improvement:** If a sensor cannot be assigned to the mission which maximize its profit, we consider the second most profitable mission, and so on ...
Performance evaluation

- **Simulation environment** implemented in **Java**.

- **Missions** and **Sensors** are deployed in **random positions** in the field.

- **Utility** of sensor $S_i$ to mission $M_j$ is a function of the **distance** $D_{ij}$ between sensor and mission.

\[
e_{ij} = \begin{cases} 
\frac{1}{1+D_{ij}^2/c} & \text{if } D_{ij} \leq SR \\
0 & \text{otherwise}
\end{cases}
\]

- For each experiment:
  - **Constant number of sensors** (200, 500, 1000)
  - **Number of simultaneous missions** was varied **from 10 to 150**.

- Experiments divided into **two main groups** to test algs’ efficiency:
Performance evaluation (Optimality)

- The graphs show the sum of the profits $p_{ij}$ of each sensor-mission assignments (i.e. the value of the objective function in the ILP formulation).

- We show only the best two algs: **Sensor-Side Greedy** and **GAP approx algorithm**
  - Optimal results **differ only by 1%** in average.
  - The best solution quality: **GAP approx algorithm**
Performance evaluation (Time)

- Best effective **running time**: Ordered Sensor-side greedy

- Best **trade off optimality/running time**: Ordered Sensor-side greedy
Conclusion

• We considered an **homogeneous sensor network** which was **already deployed**, to support **competing simultaneous missions** with **uncertain demands**.

• We developed a **formal model** and a **new greedy algorithm**.

• **Simulation results** show that: our novel **Ordered Sensor-side greedy** algorithm **offers the best trade-off** between **optimality** of the solution and effective running **time**.

• **Future work:**
  - Heterogeneous sensor types
  - Sharing of a sensing resource
Thanks for listening