Issues of Sensor Network Design

Pei-Ling Chiu
PhD. Student
Department of Information Management
National Taiwan University

Outline

- Background of Sensor Placement
- Research Framework
- Sensor Placement For Target Location
- Energy Efficient Sensor Networks Design
- Future Works
Background of Sensor Placement

Sensor nodes

- Small
- Low cost
- Low power
- sensing, processing, communication
- Memory, power, and computation capacities are limited.

2004/11/19
Sensor Networks

Applications

- Military applications
  - Surveillance
  - Target Location
  - Tracking

- Civil applications
  - Habitat monitoring applications
  - Environment observation
  - Forecasting system
  - Traffic monitoring
  - Health applications
  - Smart space (home/office)
Sensor Placement Problems

- **Goals**
  - reducing total *cost*
  - extending *coverage*
  - increasing network *lifetime*

Sensor Detection Models

- **0/1 detection model**
- **Probabilistic detection model**

  We consider the detection model of a sensor to be a *0/1 coverage model.*
Sensor Placement

- Random Placement
  - When the environment is unknown
  - Sensors may be thrown to any place by aircrafts randomly.
- Controlled Placement/Grid-based placement
  - If the terrain properties were predetermined.
  - A certain quality of service level can be guaranteed.
Grid-based placement

The positioning resolution requirement can be determined by the granularity of the grid points.

Sensor and its coverage

Grid points

Research Framework
Research Framework

<table>
<thead>
<tr>
<th>issues levels</th>
<th>Placement</th>
<th>Energy efficiency</th>
<th>Fault Tolerance</th>
<th>Mobile sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1 sensors (detection)</td>
<td>Sensor network for Target location</td>
<td>K-cover sensor network deployment</td>
<td>Robust sensor networks design</td>
<td>Consider: coverage, power</td>
</tr>
<tr>
<td>Level 2 cluster heads (transmission)</td>
<td>Cluster head deployment</td>
<td></td>
<td>Consider: Network reliability</td>
<td></td>
</tr>
</tbody>
</table>

Sensor Density vs. Service Level

<table>
<thead>
<tr>
<th>Sensor Density</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td>No Complete Coverage</td>
</tr>
<tr>
<td></td>
<td>Complete Coverage, No Complete Discrimination</td>
</tr>
<tr>
<td>high</td>
<td>Complete Coverage, Complete Discrimination</td>
</tr>
<tr>
<td></td>
<td>Complete Coverage, Complete Discrimination, Survivability; Robustness</td>
</tr>
</tbody>
</table>
Algorithms

- Lagrangean relaxation based heuristic
- Simulated Annealing based algorithm

Sensor Placement For Target Location

Reference:
Complete Coverage

- If any grid point in a sensor field can be detected by at least one sensor, we call the field a complete covered sensor field.

Power Vector

- A power vector, \(<v_1, v_2, \ldots, v_m>\), is used to indicate whether sensors \(i, 1 \leq i \leq m\), can cover a grid point in a field.
- The power vector of grid point 8 is \(<0, 0, 1, 1, 0, 0>\) corresponding to sensor 4, 6, 7, 9, 10, 12.
Complete Discrimination

- In a complete covered sensor field, when each grid point is identified by a unique power vector, we call the sensor field a complete discriminated sensor field.
- As soon as a target is discovered in the sensor field, the back-end is able to locate it according to the power vector of the grid.
- Complete Coverage/Discrimination

Distance Error

- Resource limitations
- Positioning accuracy becomes a major issue of the problem.
- Distance Error
  - The distance of two grid points is defined as the Euclidean distance between them.
  - The distance error of two indistinguishable grid points is defined as the Euclidean distance between them.
Objectives

- Sensor Placement For Target Location
  - Complete Discrimination
    - It is implied that the minimum Hamming distance of power vectors for any pair of grid points will be maximized.
  - High Discrimination
    - When complete discrimination is not possible.
    - It is implied that the maximum distance error will be minimized.

Given Parameters

- $A : \{1, 2, \ldots, m\}$: Index set of the location in the sensor field.
- $B : \{i \mid i \in A\}$: Index set of the sensor’s candidate locations.
- $r_k$: Detection radius of sensor located at $k$, $k \in B$
- $d_{ij}$: Euclidean distance between location $i$ and $j$, $i, j \in A$
- $c_k$: The cost of sensor allocated at location $k$, $k \in B$
- $G$: Total cost limitation
Decision Variables

\( y_k \) : 1, if a sensor is allocated at location \( k \), \( k \in B \).

\( v_i = (v_{i1}, v_{i2}, \ldots, v_{ik}) \) : A power vector of location \( i \), where \( v_{ik} \) is 1 if the target at location \( i \) can be detected by the sensor at location \( k \) and 0 otherwise.

Objective Function

\[
Z_{IP1} = \min_{v} \max_{(i,j)} \frac{d_{ij}}{1 + K \sum_{\forall k \in B} (v_{ik} - v_{jk})^2}
\]

(IP1)

(K is a big number.)
Constraints

\[ v_{ik}d_{ik} \leq y_k r_k \quad \forall i \in A, k \in B, i \neq k \] (1)

\[ \frac{d_{ik}}{r_k} > y_k - v_{ik} \quad \forall i \in A, k \in B, i \neq k \] (2)

\[ v_{ik} = y_k \quad \forall i \in A, k \in B, i = k \] (3)

\[ \sum_{k \in B} c_k y_k \leq G \] (4)

\[ \sum_{k \in B} v_{ik} \geq 1 \quad \forall i \in A \] (5)

Constraints (1), (2) and (3) require the relationship between sensor detection radius \( r_k \) and detection distance \( d_{ik} \).

Constraint (4) is the cost limitation.

Constraint (5) is the complete coverage limitation.
### Simulated Annealing

- Simulated Annealing (SA) is a method for obtaining good solutions to difficult optimization problems.

- SA can be considered as one type of randomized heuristic for combinatorial optimization problems.

<table>
<thead>
<tr>
<th>( r_i \cdot )</th>
<th>( r_k \cdot )</th>
<th>Conditions</th>
<th>Need</th>
<th>Constraint: (1)</th>
<th>Constraint: (2)</th>
<th>Notes</th>
<th>Physical meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Don’t care</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>-</td>
<td>No sensor, no power vector.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Don’t care</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>( d_k \leq 0 \cdot \cdot \cdot (1) )</td>
<td>No sensor, but a corresponding power vector presents.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>( d_k &gt; r_k \cdot )</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>( r_k &gt; 0 \cdot \cdot \cdot (2) )</td>
<td>Out of the detection range.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>( d_k \leq r_k \cdot )</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>( \frac{d_k}{r_k} &gt; 1 \cdot \cdot \cdot (2) )</td>
<td>Within the detection range.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>( d_k &gt; r_k \cdot )</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>( d_k \leq r_k \cdot \cdot \cdot (1) )</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>( d_k \leq r_k \cdot )</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>( \frac{d_k}{r_k} &gt; 0 \cdot \cdot \cdot (2) )</td>
<td></td>
</tr>
</tbody>
</table>

Notes: \( r_k \cdot d_k \leq r_k \cdot \cdot \cdot (1) \). \( \frac{d_k}{r_k} > r_k \cdot \cdot \cdot (2) \).
SA Algorithm in Pseudo-Code

Select an initial state \( i \in S; \)
Select an initial temperature \( T > 0; \)
Set temperature change counter \( t = 0; \)
Repeat
Set repetition counter \( n = 0; \)
    Repeat
Generate state \( j \), a neighbor of \( i; \)
Calculate \( \delta = f(j) - f(i); \)
If \( \delta < 0 \) then \( i := j \)
    else if random(0,1)< exp( \(-\delta / T \) ) then \( i := j \);
    \( n := n + 1; \)
    until \( n = N(t); \)
\( t := t + 1; \)
\( T := T(t); \)
Until stopping criterion true.

Algorithm

1. Deploy sensors on all grid points // Initial guess
2. repeat until \( t < t_f \)
3. repeat \( r \) times
4. If the cost is still higher than the cost limitation then
5. Discard a sensor randomly
6. Decide the new configuration can be accepted or not
7. If accepted then goto step 10
8. Move a sensor to a new grid point randomly
9. Decide the new configuration can be accepted or not
10. If the best deployment is the desired solution then stop
11. \( t := t \times 0.75, \ r := r \times 1.3 \) // Cooling function
Experiments

- Two experiments
  - In the case of smaller sensor fields.
    - whether the algorithm can find the optimal solution under the minimum cost limitation
  - In the case of larger sensor fields under various cost limitations.

Parameters
- All sensors have the same cost.
- Sensor radius is 1
- K=10,000

Performance metric

\[
Sensor\ density(\%) = \frac{\sum_{k \in B} y_k \times 100\%}{n}
\]
Experiment I

- Scenario
  - Sensor field: rectangular and no more than 30 grid points.
  - Benchmark: the exhaustive search.
- First, we find a minimum sensor density by the exhaustive search.
- Then, we find the same result by the SA algorithm under the constraint of the sensor density.

2004/11/19 33

Experiment I

Results

Table 1: Comparison between the exhaustive search and the SA algorithm.

<table>
<thead>
<tr>
<th>Area</th>
<th># of sensors</th>
<th>Sensor density</th>
<th>Area</th>
<th># of sensors</th>
<th>Sensor density</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Opt.</td>
<td>SA</td>
<td></td>
<td>Opt.</td>
<td>SA</td>
</tr>
<tr>
<td>3x3</td>
<td>4</td>
<td>4</td>
<td>44.44%</td>
<td>6x4</td>
<td>10</td>
</tr>
<tr>
<td>4x3</td>
<td>6</td>
<td>6</td>
<td>50%</td>
<td>6x5</td>
<td>12</td>
</tr>
<tr>
<td>4x4</td>
<td>7</td>
<td>7</td>
<td>43.75%</td>
<td>7x3</td>
<td>9</td>
</tr>
<tr>
<td>5x3</td>
<td>6</td>
<td>6</td>
<td>40%</td>
<td>7x4</td>
<td>12</td>
</tr>
<tr>
<td>5x4</td>
<td>8</td>
<td>8</td>
<td>40%</td>
<td>8x3</td>
<td>10</td>
</tr>
<tr>
<td>5x5</td>
<td>10</td>
<td>10</td>
<td>40%</td>
<td>9x3</td>
<td>11</td>
</tr>
<tr>
<td>6x3</td>
<td>8</td>
<td>8</td>
<td>44.44%</td>
<td>10x3</td>
<td>12</td>
</tr>
</tbody>
</table>

Note: Opt.: by the exhaustive search.
Experiment I

- Remarks
  - Optimality – yes, at least in the smaller cases.

  - Solution time (for 10 x 3 grid points)
    - The exhaustive search: more than 65 minutes.
    - The SA algorithm: in 0.1 second.

  - The required sensor density ranges between 40% and 45%.

Experiment II

- Scenario
  - Sensor field:
    - rectangular
    - 10 x 10 and 30 x 30 sensor fields

  - Comparison: the best solution of a random placement approach in 1000 arbitrary generated solutions.
Experiment II

The sensor placement in a 10 by 10 grids
(The sensor radius is 1)

Minimum required density for complete coverage

25%

Minimum required density for complete coverage and discrimination

42%

No feasible solution

Experiment II

The sensor placement in a 10 by 10 grids
(The sensor radius is 2)

11%

27%
Experiment II

The sensor placement in a 30 by 30 grids.
(The sensor radius is 1)

Experiment II

The sensor placement in a 30 by 30 grids.
(The sensor radius is 2)
Experiment II

Remarks
- Very effective and scalable.
  - Optimality – No, but very close.
  - Solution time (for 30 x 30 grid points)
    - Only a couple of minutes.
- Robustness
  - ZIP is stable.
- For a complete covered deployment
  - The error distance is minimized ($ZIP=1$).
  - The required density is very close to the theoretical lower bound.

Energy Efficient Sensor Networks Design

Reference:
Problem

- Constructing a sensor network for the target positioning service
- The design goals
  - to achieve target positioning
  - to increase sensor lifetime
- The target positioning application
  - the complete coverage and discrimination issues
  - Such sensor networks require higher sensor density than the sensor network only providing surveillance application.

Motivation

- It is not necessary to keep target positioning service always if intrusion events occur infrequently in the sensor field.

- In fact, the surveillance service is adequate when there is not any intruder in the sensor field.
Consideration

- Multiple independent sets of sensors are allocated during the deployment phase.
- These sets are activated in turn to keep complete coverage of the monitoring field when no any intruder has existed in the field.
- Once the intrusion event occurs, all sensors on the field are activated and work corporately to locate intruder.

We formulate the problem as a 0/1 integer programming problem

- The objective function is the minimization of the total deployment cost
- Subject to complete coverage and discrimination constraints under a given amount of cover \( K \)
- The problem is a variant of the set \( K \)-cover problem and thus is NP-complete.
### Assumptions

- All radiuses of the sensors are constant.
- Each candidate location can only allocate one sensor.
- The power consumption for inactive sensors can be ignored.
- Intrusion events occur infrequently.

### Objective Function and Constraints

<table>
<thead>
<tr>
<th>$Z_{IP}$</th>
<th>$= \min \sum_{j=1}^{m} \sum_{k=1}^{K} c_{j,k} x_{j,k}$</th>
<th>(IP)</th>
</tr>
</thead>
</table>

Subject to:

| $\sum_{j=1}^{m} a_{i,j} x_{j,k}$ | $\geq 1$ | $\forall i \in B, 1 \leq k \leq K$ | (1) |
| $\sum_{k=1}^{K} x_{j,k}$ | $\leq 1$ | $\forall j \in A$ | (2) |
| $y_{j}$ | $= \sum_{k=1}^{K} x_{j,k}$ | $\forall j \in A$ | (3) |
| $\sum_{j=1}^{m} (a_{i,j} - a_{i,l}) y_{j}$ | $\geq 1$ | $\forall i,l \in B, i \neq l$ | (4) |
| $x_{j,k}$ | $= 0$ or $1$ | $\forall j \in A, 1 \leq k \leq K$ | (5) |
| $y_{j}$ | $= 0$ or $1$ | $\forall j \in A$ | (6) |
Lagrangean Relaxation Problem

- dualize Constraints (1), (3) and (4)

\[
Z_0(u^1, u^2, u^3) = \min \left\{ \sum_{j=1}^{m} \sum_{k=1}^{K} c_{jk} x_{jk} + \sum_{j=1}^{m} \sum_{k=1}^{K} u^1_{jk} (1 - \sum_{j=1}^{m} a_{jk} x_{jk}) + \sum_{j=1}^{m} u^2_j (y_j - \sum_{j=1}^{K} x_{jk}) + \sum_{i=1}^{m} \sum_{j=1}^{m} u^3_{ij} (1 - \sum_{j=1}^{m} (a_{ij} - a_{ii}) y_{ij}) \right\}
\]

(LLR)

- S.t. (2), (5), (6)
- The multipliers \( u^1 \), \( u^2 \), and \( u^3 \) are the vectors of \( \{ u^1_{jk} \} \), \( \{ u^2_j \} \), and \( \{ u^3_{ij} \} \), respectively.

Subproblem 1

- For \( x_{jk} \)

\[
Z_{\text{sub1}}(u^1, u^2) = \min \left\{ \sum_{j=1}^{m} \sum_{k=1}^{K} \left( c_{j} - u^2_j \right) - \sum_{j=1}^{m} u^1_{ji} a_{ij} \right\} x_{jk} \right\}
\]

(sub1)

\[
\sum_{k=1}^{K} x_{jk} \leq 1 \quad \forall j \in A \quad (2)
\]
\[
x_{jk} = 0 \text{ or } 1 \quad \forall j \in A, 1 \leq k \leq K \quad (5)
\]
Subproblem 2

- For $y_j$

$$Z_{\text{sub2}}(u^+, u^-) = \min \sum_{j=1}^{m} (u^+_j - \sum_{i=1}^{n} \sum_{l \neq i}^{n} u^+_{il}(a_{ij} - a_{ij})^2)y_j$$ \hspace{1cm} \text{(sub2)}$$

s.t.

$$y_j = 0 \text{ or } 1 \hspace{1cm} \forall j \in A \hspace{1cm} \text{(6)}$$

Getting Primal Feasible Solution

- Step 1: Check constraint (3) for each sensor.

- Step 2: For each cover $k$ of the sensor network, check coverage constraint (1).
  - Addition, Exchange, Drop

- Step 3: Check the discrimination constraint (4) for the whole sensor network.
Experiment

- Scenario
  - The algorithm was tested on a 10 x 10 service area.
  - The distance between two adjacent grid points defines the length unit.
  - Seven sets of experiments are conducted which consider sensor radius $r$ ranging from $1^+$ to $7^+$.
  - All sensors have the same deployment cost.

Comparison of $U_r$ between the theoretical and best found values

- From this perspective, the proposed algorithm is very effective for maximizing the network lifetime.

- $U_r$: Maximum $K$

<table>
<thead>
<tr>
<th>$r$</th>
<th>$U_r$ (theoretic)</th>
<th>$U_r$ (the best found)</th>
<th>degradation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1^+$</td>
<td>3</td>
<td>3</td>
<td>0%</td>
</tr>
<tr>
<td>$2^+$</td>
<td>6</td>
<td>6</td>
<td>0%</td>
</tr>
<tr>
<td>$3^+$</td>
<td>11</td>
<td>11</td>
<td>0%</td>
</tr>
<tr>
<td>$4^+$</td>
<td>17</td>
<td>17</td>
<td>0%</td>
</tr>
<tr>
<td>$5^+$</td>
<td>26</td>
<td>26</td>
<td>0%</td>
</tr>
<tr>
<td>$6^+$</td>
<td>35</td>
<td>34</td>
<td>2.9%</td>
</tr>
<tr>
<td>$7^+$</td>
<td>45</td>
<td>43</td>
<td>4.4%</td>
</tr>
</tbody>
</table>
Sensor Densities Obtained in Experiment

<table>
<thead>
<tr>
<th># of covers</th>
<th>Sensor radius</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1+</td>
</tr>
<tr>
<td>1</td>
<td>0.41</td>
</tr>
<tr>
<td>3</td>
<td>0.8</td>
</tr>
<tr>
<td>6</td>
<td>0.8</td>
</tr>
<tr>
<td>11</td>
<td>0.78</td>
</tr>
<tr>
<td>17</td>
<td>0.79</td>
</tr>
<tr>
<td>26</td>
<td>0.97</td>
</tr>
<tr>
<td>34</td>
<td>0.97</td>
</tr>
<tr>
<td>43</td>
<td>0.97</td>
</tr>
</tbody>
</table>

The Lifetime Extending Times vs. Average Sensor Density Per Cover

- The proposed sensor placement algorithm is extremely effective for minimizing the sensor density increase in extending lifetime.

[Graph showing lifetime extending times vs. sensor density per cover]
The Proposed Approach vs. The Duplicate Placement Approach

<table>
<thead>
<tr>
<th>Radius</th>
<th>The duplicate sensor placement</th>
<th>The proposed approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#Duplication</td>
<td>Increased cost</td>
</tr>
<tr>
<td>1°</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2°</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>3°</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>4°</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>5°</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>6°</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>7°</td>
<td>43</td>
<td>43</td>
</tr>
</tbody>
</table>

The Computation Time

- The efficiency of the algorithm thus can be confirmed.

Table 5: The maximum execution time of each set experiments.

<table>
<thead>
<tr>
<th>Sensor radius</th>
<th>1°</th>
<th>2°</th>
<th>3°</th>
<th>4°</th>
<th>5°</th>
<th>6°</th>
<th>7°</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution time (Second)</td>
<td>53</td>
<td>61</td>
<td>85</td>
<td>35</td>
<td>25</td>
<td>91</td>
<td>51</td>
</tr>
</tbody>
</table>
The Sensor Radius vs. The Density Requirement

Future Works
Research Issues

- Robust Sensor Networks Design
- Cluster Heads Placement
- Mobile Sensor Networks