Optimal Sensor Sequencing for Container Inspection

Endre Boros, Paul Kantor, Noam Goldberg, Jonathan Word
Rutgers University, SCILS and RUTCOR

Research Supported by: DHS DNDO#CBET-0735910
The Problem:
There are many tests that can be applied (document checks, passive sensors of several kinds, active sensors). Find the “optimal” detection policy based on these tests!
Assumptions:

Randomness arises from the enormous variability in contents, screening and background, not in the sensors themselves. Therefore, a repeat reading with a sensor gives the same value.

Sensors are stochastically independent. So the probability of any collection of readings or signals, given the TRUTH, is the product of the probabilities for the individual readings.
Background

- Complete enumeration (Stroud and Saeger, 2003)
- Linear programming model (Boros, Fedzhora, Kantor, Stroud, and Saeger, 2006)
- Threshold optimization (Zhang, Schroepfer, Elsayed, 2006)
- Heuristic search (Madigan, Mittal, Roberts, 2007)
- 3-sensor cost-time model (Young, Li, Zhu, Xie, Elsayed, and Asamov, 2008)
- Dynamic programming (Boros, Kantor, Goldberg and Word, 2008)
Move to a decision support model:

Minimize total damage over all available policies

\[
\text{Min}_p \ C(P) + \pi K (1 - \Delta(P))
\]

\(\Delta(P), C(P)\) - detection rate, and operating cost of policy P

\(\pi \ (\sim 0), K \ (\sim \text{very large})\) - a priori probability of a “bomb”, and expected cost of false negative

\[
\text{Max}_p \ \{ \Delta(P) \mid C(P) \leq B \}
\]

mixing and domination of policies

concave envelop of best policies
Improve computational efficiency:

- Move from **signal** space to **ROC** space

- Dynamic programming algorithm
  - **Sensor fusion** (multi-knapsack model)
  - **Bottom up enumeration**
  - **Large number of channels** (threshold optimization)

- **Effective approximation of concave envelop**
Move from **signal** space to **ROC** space:
Dynamic Programming: Sensor Fusion

Given a set of policies and an additional sensor, what is the best set of policies that we can construct?
Dynamic Programming: Sensor Fusion

Fusing sensor $k$ on top of the given policies optimally is a multi-knapsack problem that can be solved by a modified greedy algorithm:
Dynamic Programming: common concave envelope

We then merge the given policies with the best combination of them with sensor \( k \) on top – and generate the common concave envelope of all these policies.
Dynamic Programming: enumeration

- For each subset $S$ of sensors and element $k$ in $S$ we fuse sensor $k$ on top of the best policies constructible from $S\{k\}$
- Do it in order of increasing sizes of subsets $S$
Dynamic Programming: Summary

• We build the concave envelope of best possible policies constructible from the given set of $N$ tests.

• We solve $N^2$ sensor fusion problems (for up to $N \leq 20$)

• Each Sensor Fusion can be solved in $O(P \times B + P \log(P))$ time, where $B$ is the number of channels of the top sensor, and $P$ is the number of pure strategies being considered.
## Approximating the input

<table>
<thead>
<tr>
<th>Epsilon</th>
<th>Total Error</th>
<th>Points</th>
<th>Time (s)</th>
<th>Number of Channels of the Given Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00%</td>
<td>3.94%</td>
<td>1567</td>
<td>8.10</td>
<td>8</td>
</tr>
<tr>
<td>0.90%</td>
<td>3.55%</td>
<td>1589</td>
<td>8.26</td>
<td>8</td>
</tr>
<tr>
<td>0.80%</td>
<td>3.16%</td>
<td>1683</td>
<td>8.97</td>
<td>8</td>
</tr>
<tr>
<td>0.70%</td>
<td>2.77%</td>
<td>2004</td>
<td>11.11</td>
<td>9</td>
</tr>
<tr>
<td>0.60%</td>
<td>2.38%</td>
<td>2341</td>
<td>15.53</td>
<td>9</td>
</tr>
<tr>
<td>0.50%</td>
<td>1.99%</td>
<td>2811</td>
<td>22.05</td>
<td>10</td>
</tr>
<tr>
<td>0.40%</td>
<td>1.59%</td>
<td>5635</td>
<td>55.09</td>
<td>11</td>
</tr>
<tr>
<td>0.30%</td>
<td>1.19%</td>
<td>8710</td>
<td>118.10</td>
<td>13</td>
</tr>
<tr>
<td>0.20%</td>
<td>0.80%</td>
<td>13905</td>
<td>311.66</td>
<td>15</td>
</tr>
<tr>
<td>0.10%</td>
<td>0.40%</td>
<td>52477</td>
<td>3998.13</td>
<td>21</td>
</tr>
</tbody>
</table>
What about approximating the output in each step?

<table>
<thead>
<tr>
<th>Saeger and Stroud Sensors (4 sensors)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time (sec)</strong></td>
<td><strong>Number of Strategies</strong></td>
</tr>
<tr>
<td>1.5</td>
<td>695</td>
</tr>
<tr>
<td>3</td>
<td>1677</td>
</tr>
<tr>
<td>5</td>
<td>2283</td>
</tr>
<tr>
<td>114</td>
<td>13845</td>
</tr>
<tr>
<td>1440</td>
<td>52319</td>
</tr>
<tr>
<td>1441</td>
<td>68</td>
</tr>
</tbody>
</table>
Cost vs. Detection curve, 4 Sensors, 68 Strategies
Detection = 81.527%
Cost = 0.1977826 units = $11.867 (< $13+)