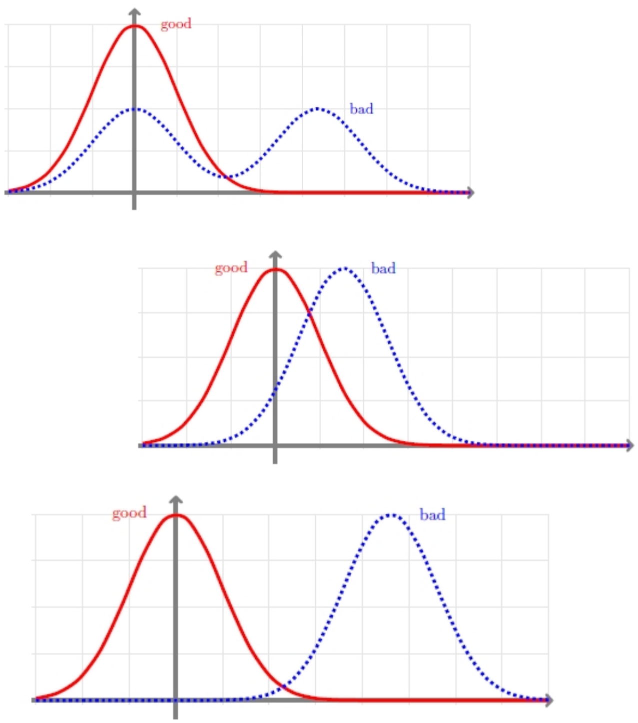
Optimal Sensor Sequencing for Container Inspection

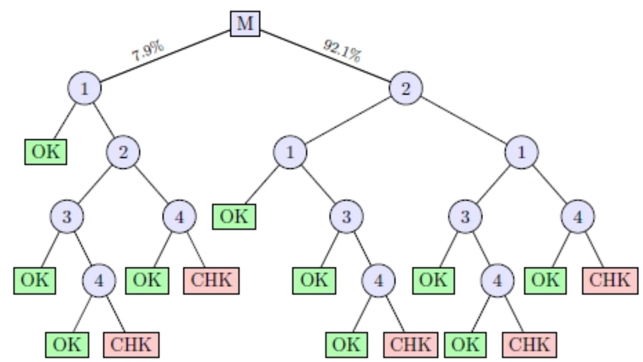
Endre Boros, Paul Kantor, Noam Goldberg, Jonathan Word
Rutgers University, SCILS and RUTCOR
Paul.kantor@rutgers.edu Endre.Boros@rutcor.rutgers.edu

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

$$\text{Min}_P \; C(P) + \pi K(1 - \Delta(P))$$

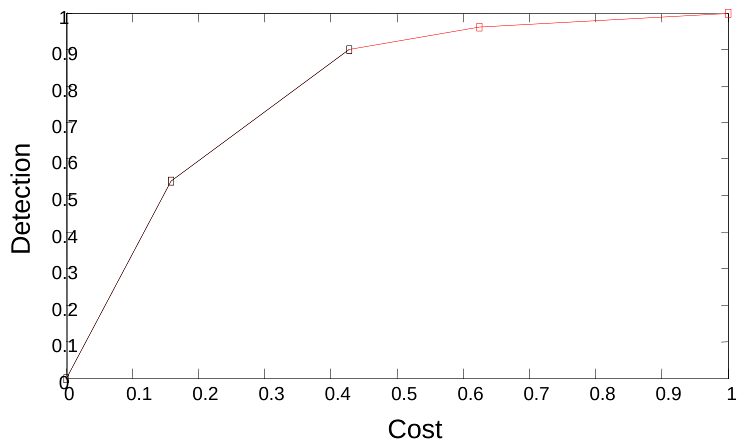
$\Delta(P)$, $C(P)$ - detection rate, and operating cost of policy P

π (~ 0), K (~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:

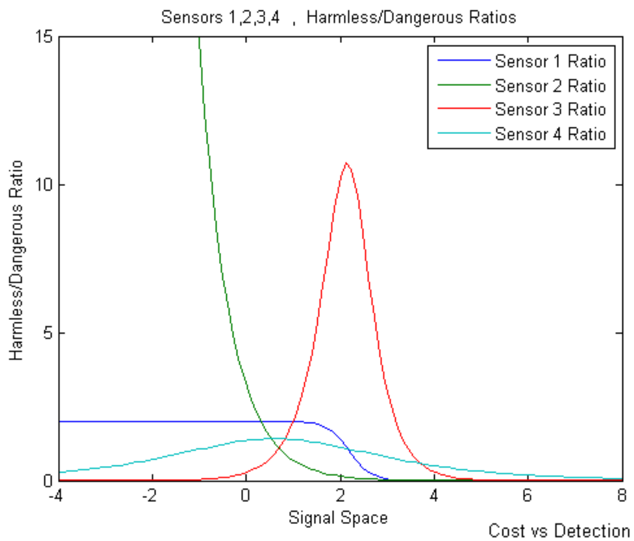
- **Move from signal space to ROC space**

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

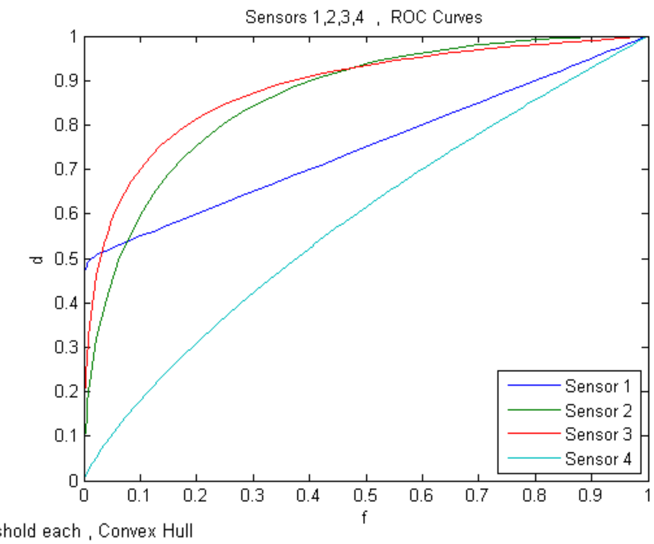
- **Effective approximation of concave envelop**

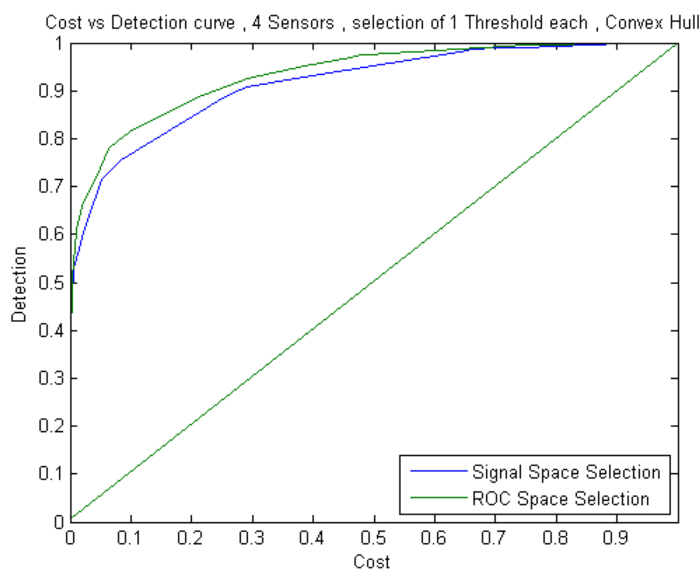
Move from signal space to ROC space:

Signal space



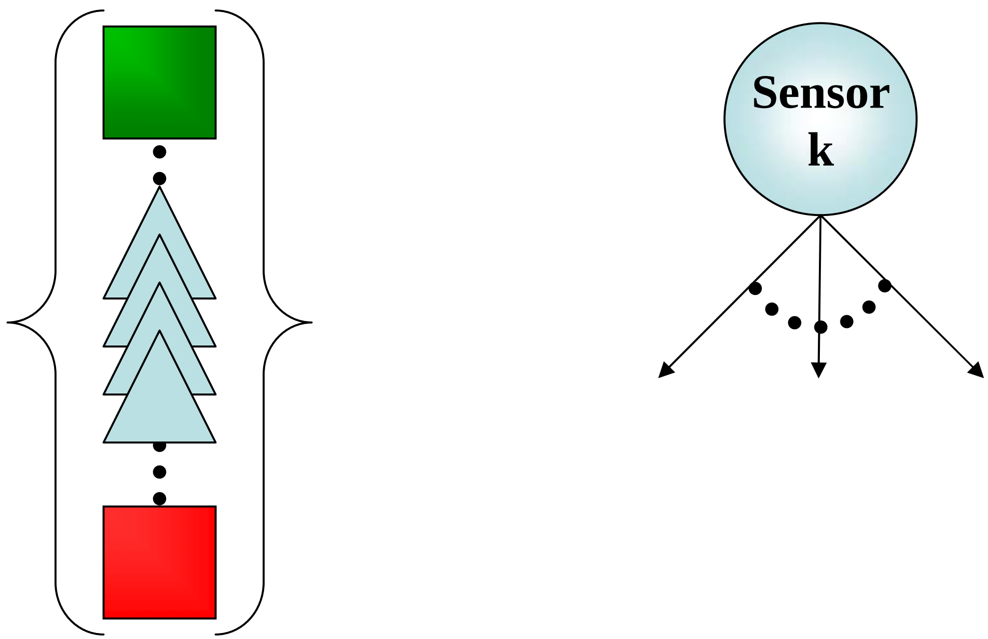
ROC space





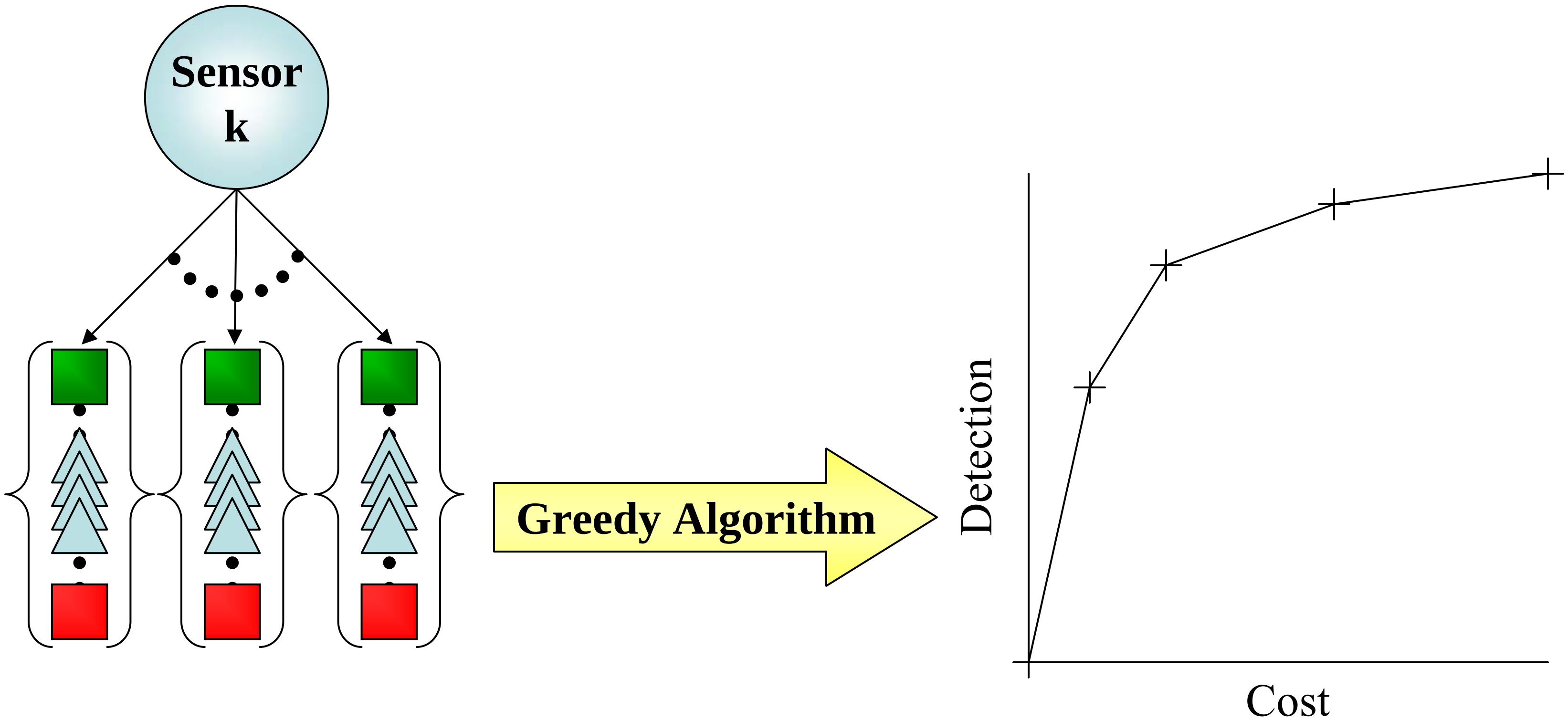
Dynamic Programming: Sensor Fusion

Given a set of policies and an additional sensor, what is the best set of policies that we can construct?



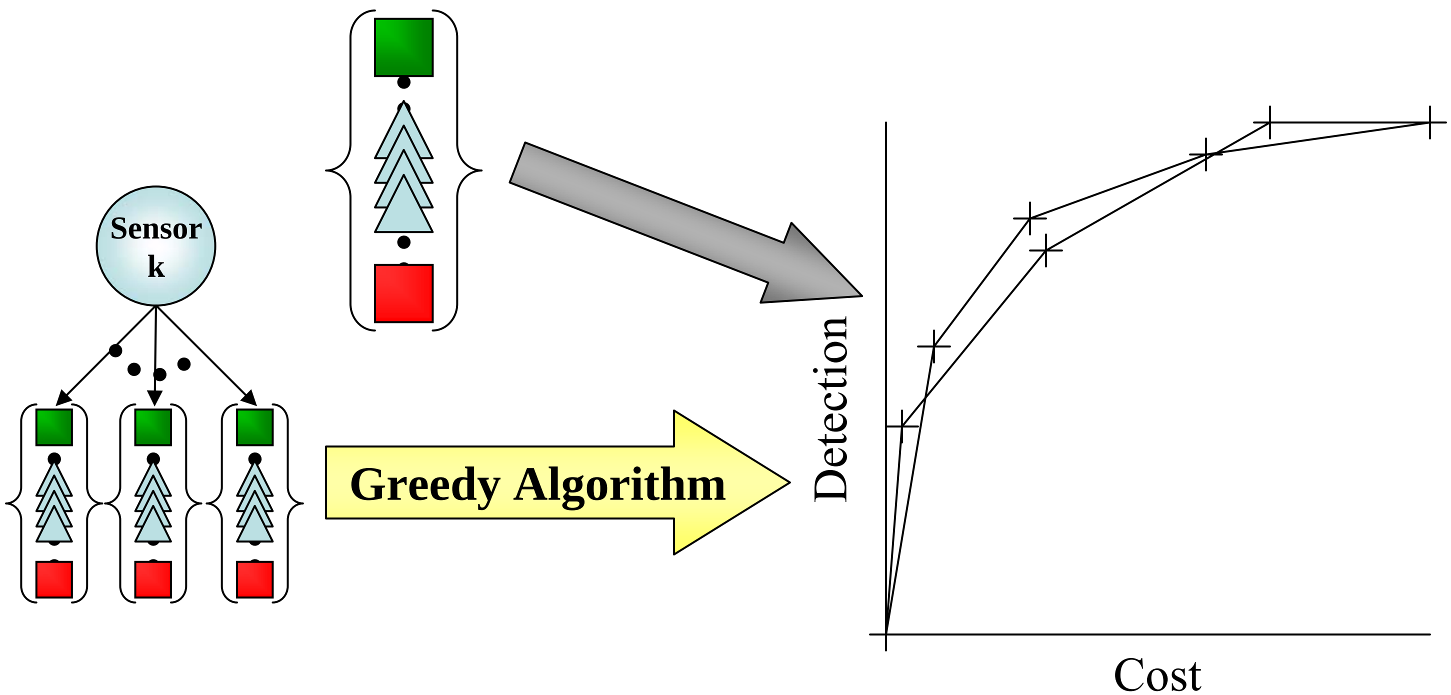
Dynamic Programming: Sensor Fusion

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



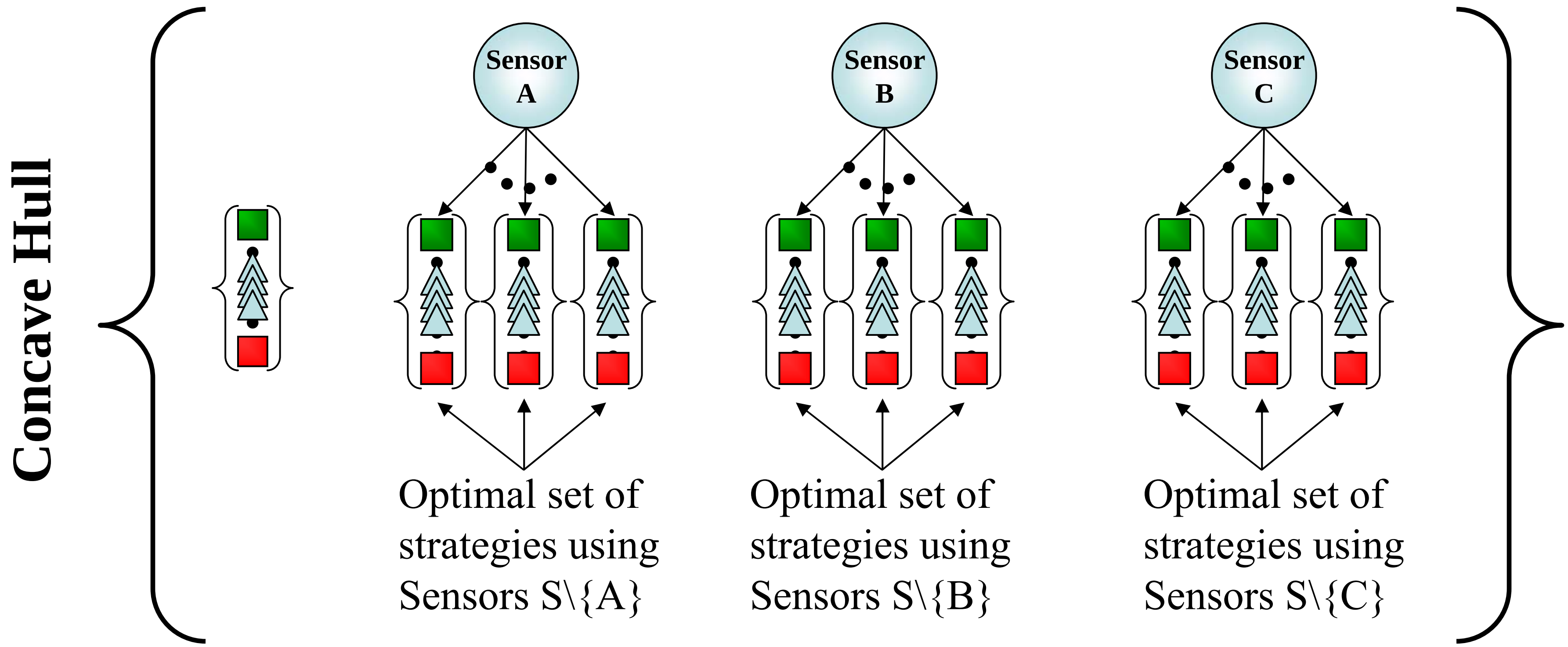
Dynamic Programming: common concave envelope

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

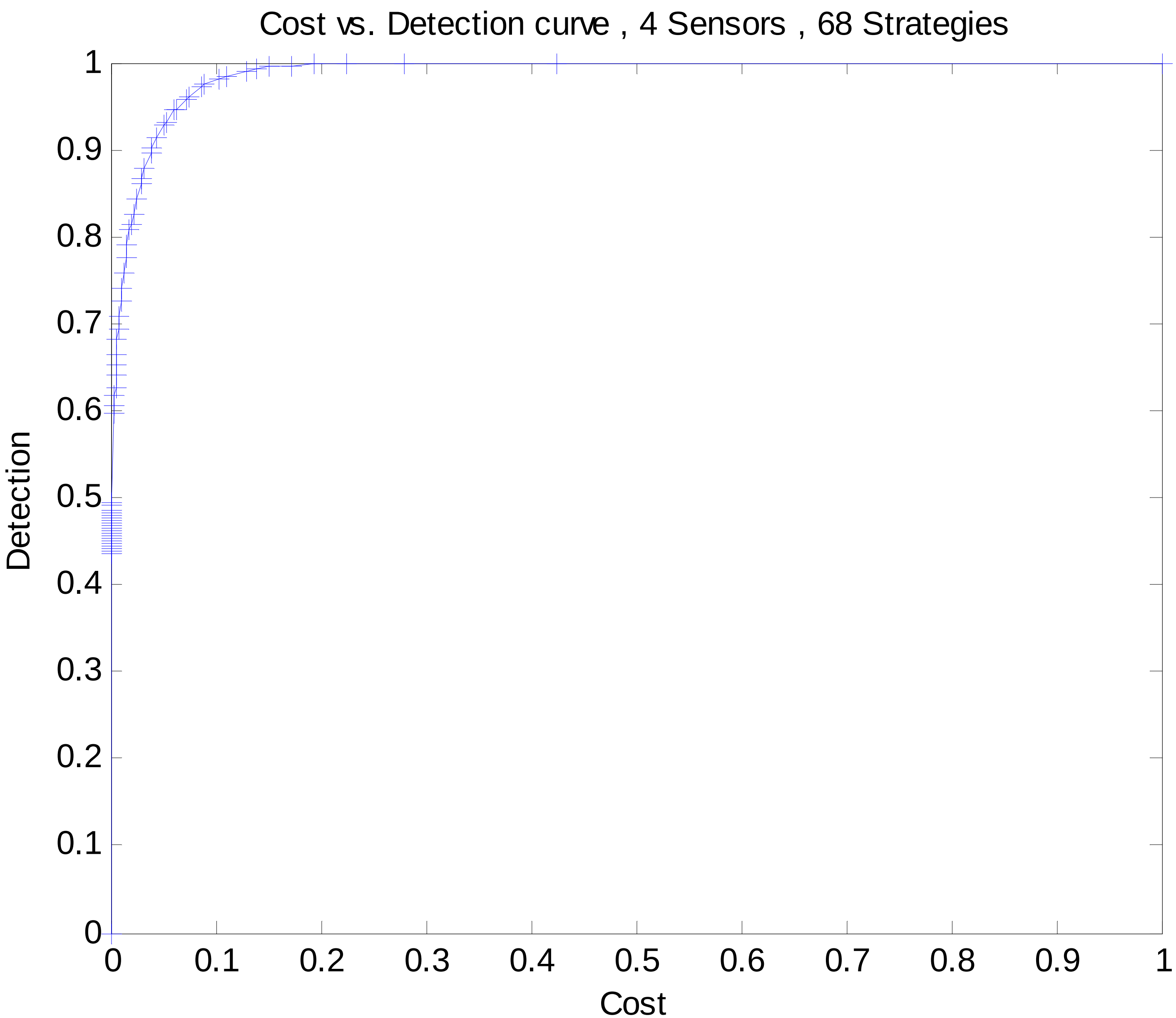
- We build the concave envelope of best possible policies constructible from the given set of N tests.
- We solve $N2^N$ sensor fusion problems (for up to $N \leq 20$)
- Each Sensor Fusion can be solved in $O(P*B+P\log(P))$ time, where B is the number of channels of the top sensor, and P is the number of pure strategies being considered

Approximating the input

Epsilon	Total Error	Points	Time (s)	Number of Channels of the Given Sensors				
1.00%	3.94%	1567	8.10		8	14	6	3
0.90%	3.55%	1589	8.26		8	14	6	3
0.80%	3.16%	1683	8.97		8	15	6	3
0.70%	2.77%	2004	11.11		9	16	6	3
0.60%	2.38%	2341	15.53		9	17	7	3
0.50%	1.99%	2811	22.05		10	19	7	3
0.40%	1.59%	5635	55.09		11	21	8	4
0.30%	1.19%	8710	118.10		13	24	9	4
0.20%	0.80%	13905	311.66		15	29	11	4
0.10%	0.40%	52477	3998.13		21	40	15	6

What about approximating the output in each step?

Saeger and Stroud Sensors (4 sensors)		
Time (sec)	Number of Strategies	Maximum Relative Error %
1.5	695	7.76%
3	1677	5.1%
5	2283	3.16%
114	13845	2.38%
1440	52319	0.8%
1441	68	0.81%


Cost vs. Detection curve , 4 Sensors , 68 Strategies
Detection
Cost
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Detection = 81.527%

Cost = 0.1977826 units = $11.867 (< $13+)