Optimized Information Gathering in Robotics and Sensor Networks

Andreas Krause

California Institute of Technology

rsrg@caltech

..where theory and practice collide

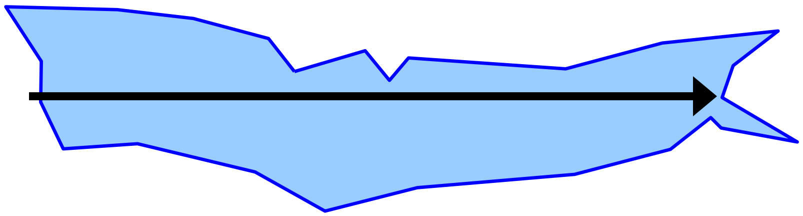
Monitoring algal blooms

- Algal blooms threaten freshwater
  - **4 million people** without water
  - **1300 factories** shut down
  - **$14.5 billion** to clean up
  - Other occurrences in Australia, Japan, Canada, Brazil, Mexico, Great Britain, Portugal, Germany …
- Growth processes still unclear [Carmichael]
  - **Need to characterize growth in the lakes, not in the lab!**

Tai Lake China
10/07 MSNBC

Monitoring rivers and lakes [IJCAI '07]

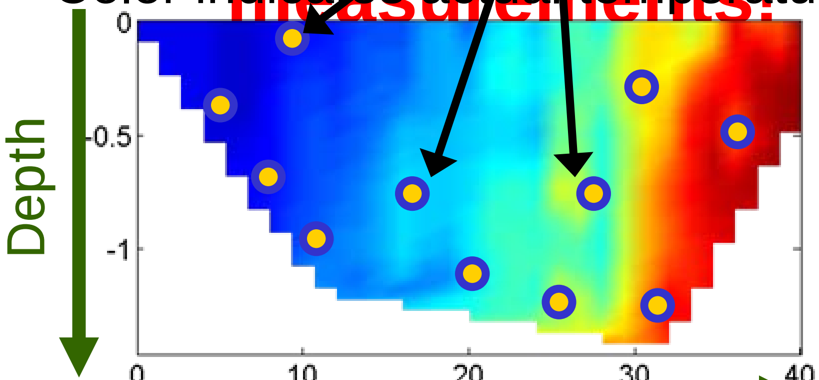
- Need to monitor large spatial phenomena
  - Temperature, nutrient distribution, fluorescence, …



Can only make a limited number of measurements!

NIMS
Kaiser et.al. (UCLA)

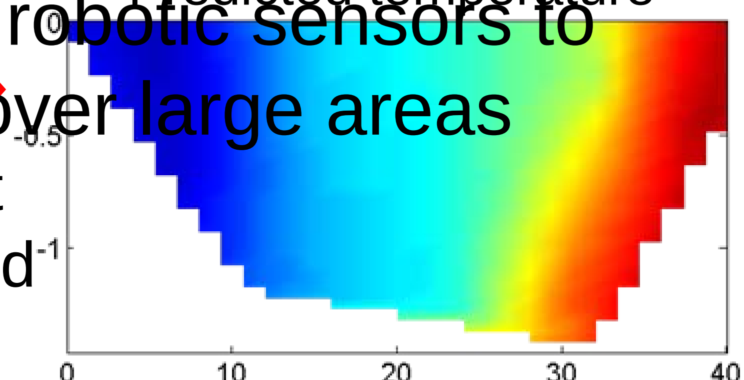
Color indicates actual temperature



Use robotic sensors to cover large areas

Predict at unobserved locations

Predicted temperature

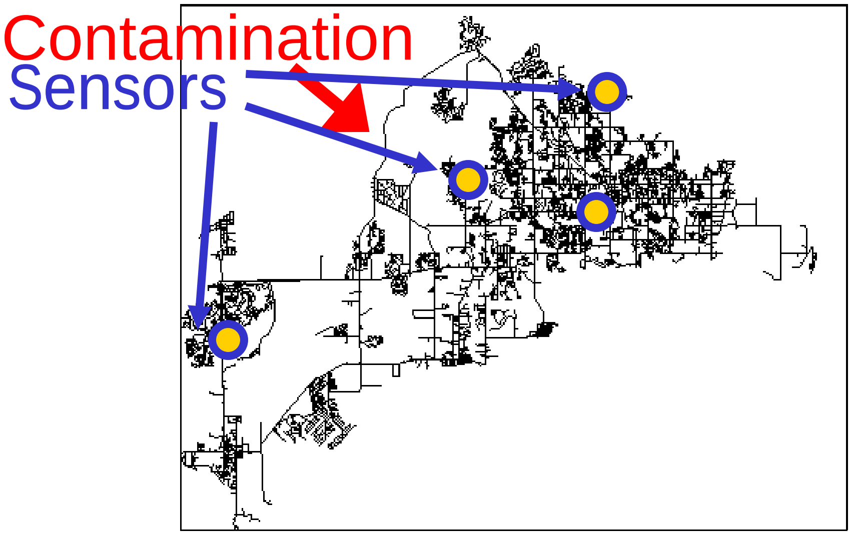


Where should we sense to get most accurate predictions?

Monitoring water networks

[K, Leskovec, Guestrin, VanBriesen, Faloutsos, J Wat Res Mgt 2008]

- Contamination of drinking water could affect millions of people



Simulator from EPA

Hach Sensor

~$14K

- Place sensors to detect contaminations
- "Battle of the Water Sensor Networks" competition

Where should we place sensors to quickly detect contaminati

Information gathering problems

- Want to learn something about the state of the world
  - Estimate water quality in a geographic region, detect outbreaks, …
- We can choose (partial) observations…
  - Make measurements, place sensors, choose experimental parameters …

- … but they are expensive / limited
  - hardware cost, power consumption, grad student time …

Want to cost-effectively get most useful information!

Related work

Sensing problems considered in

Experimental design (Lindley '56, Robbins '52…), Spatial statistics (Cressie '91, …), Machine Learning (MacKay '92, …), Robotics (Sim&Roy '05, …), Sensor Networks (Zhao et al '04, …), Operations Research (Nemhauser '78, …)

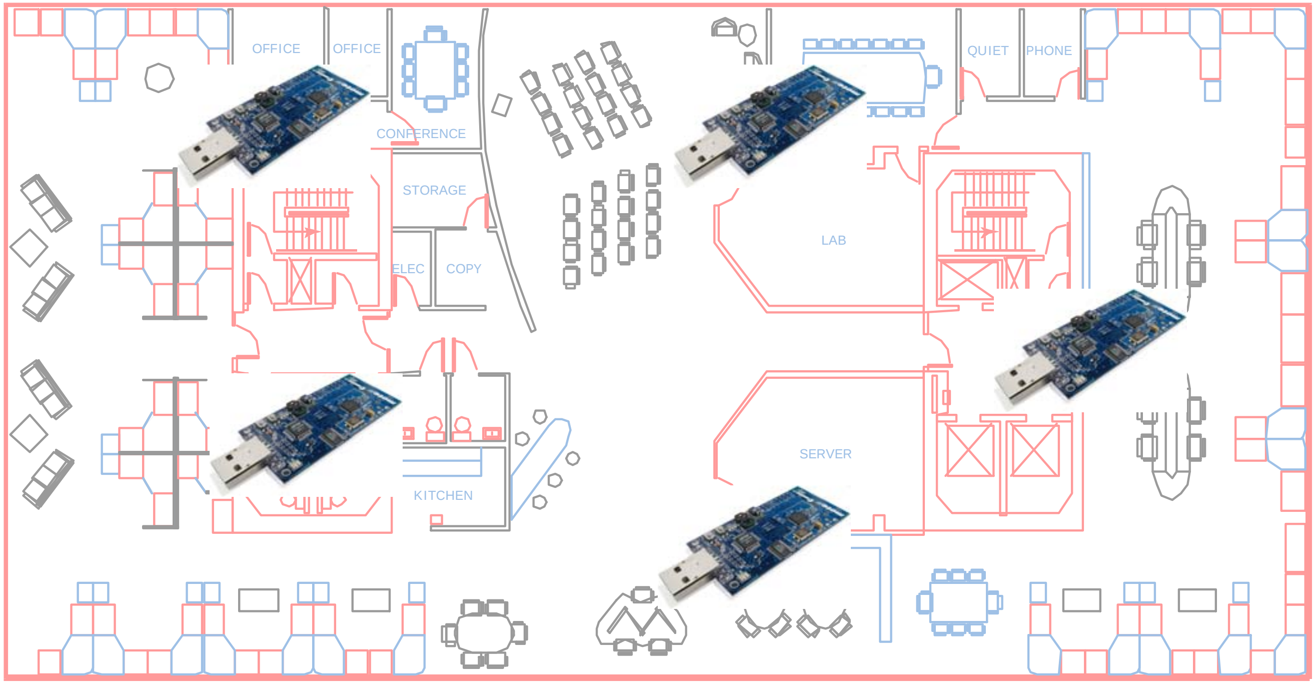
Existing algorithms typically

- **Heuristics**: No guarantees! Can do arbitrarily badly.
- **Find optimal solutions** (Mixed integer programming, POMDPs):

  Very difficult to scale to bigger problems.

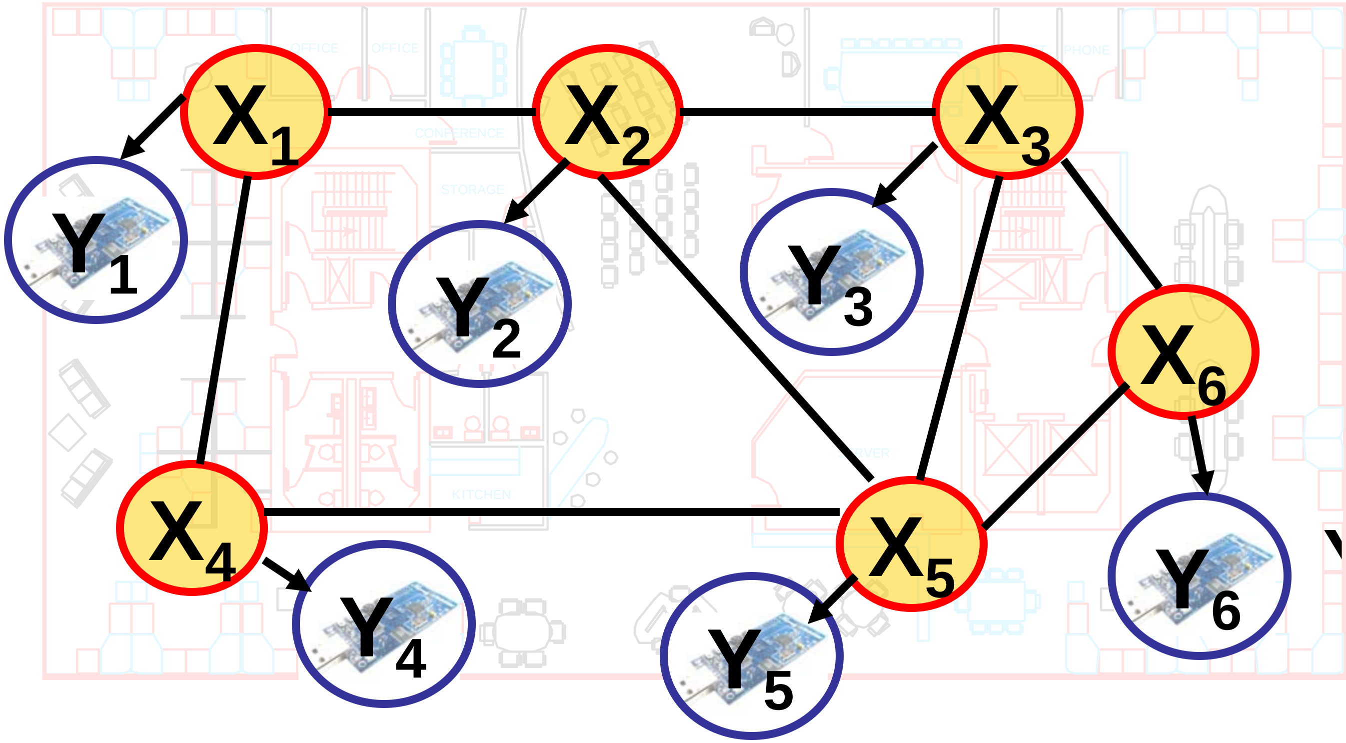
Want algorithms that have theoretical guarantees and scale to large problems!

Running example: Detecting fires



Want to place sensors to detect fires in buildings

Monitoring as Bayesian regression



X_s: temperature at location s

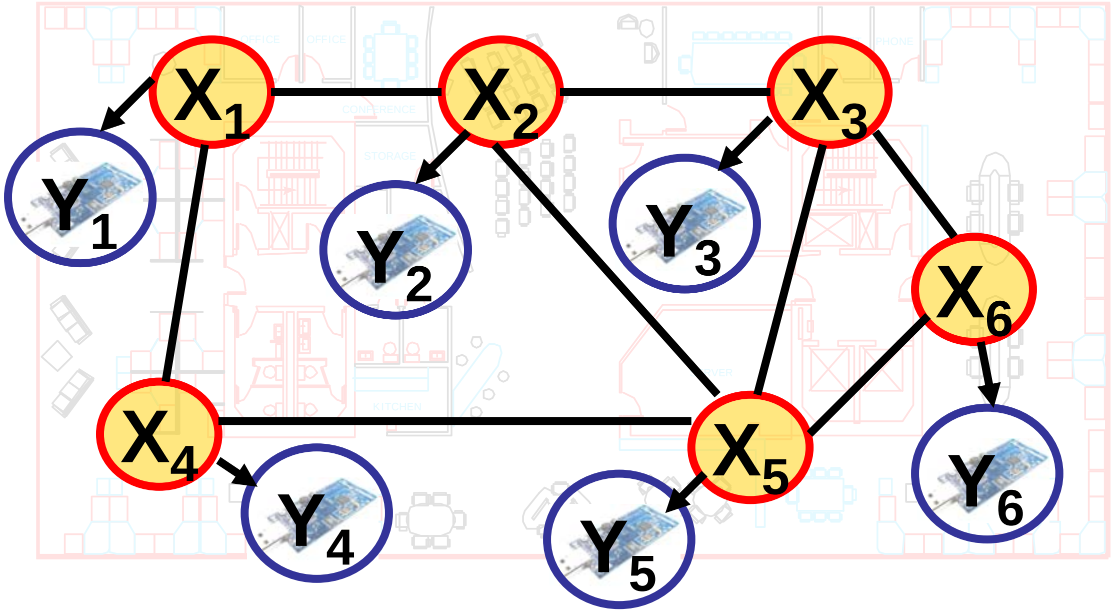
Y_s: sensor value at location s

$$\mathbf{Y_s = X_s + noise}$$

Joint probability distribution

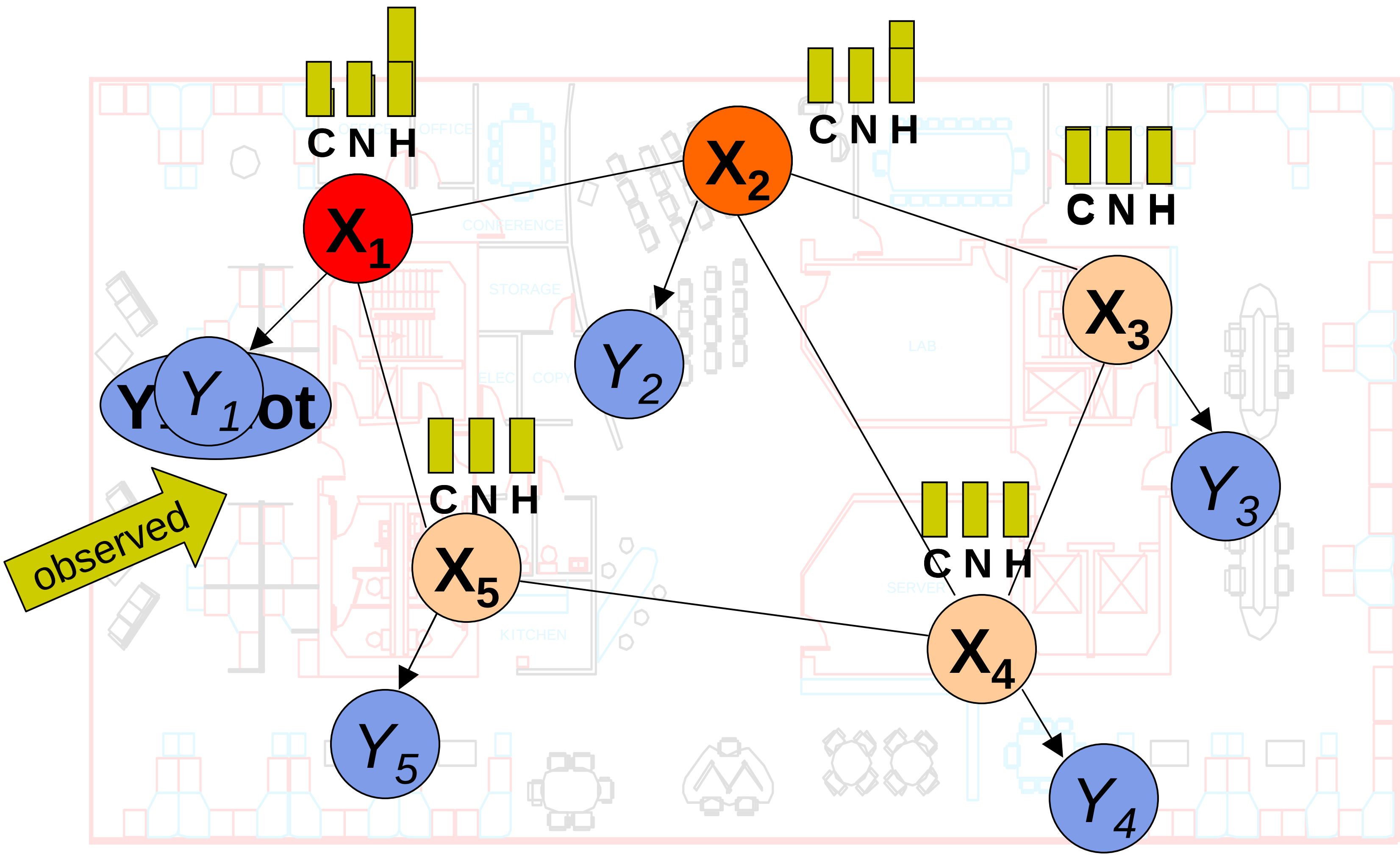
$$P(X_1,\ldots,X_n,Y_1,\ldots,Y_n) = \underbrace{P(X_1,\ldots,X_n)}_{\textbf{Prior}}\ \underbrace{P(Y_1,\ldots,Y_n \mid X_1,\ldots,X_n}_{\textbf{Likelihood}}$$

Why is this useful?



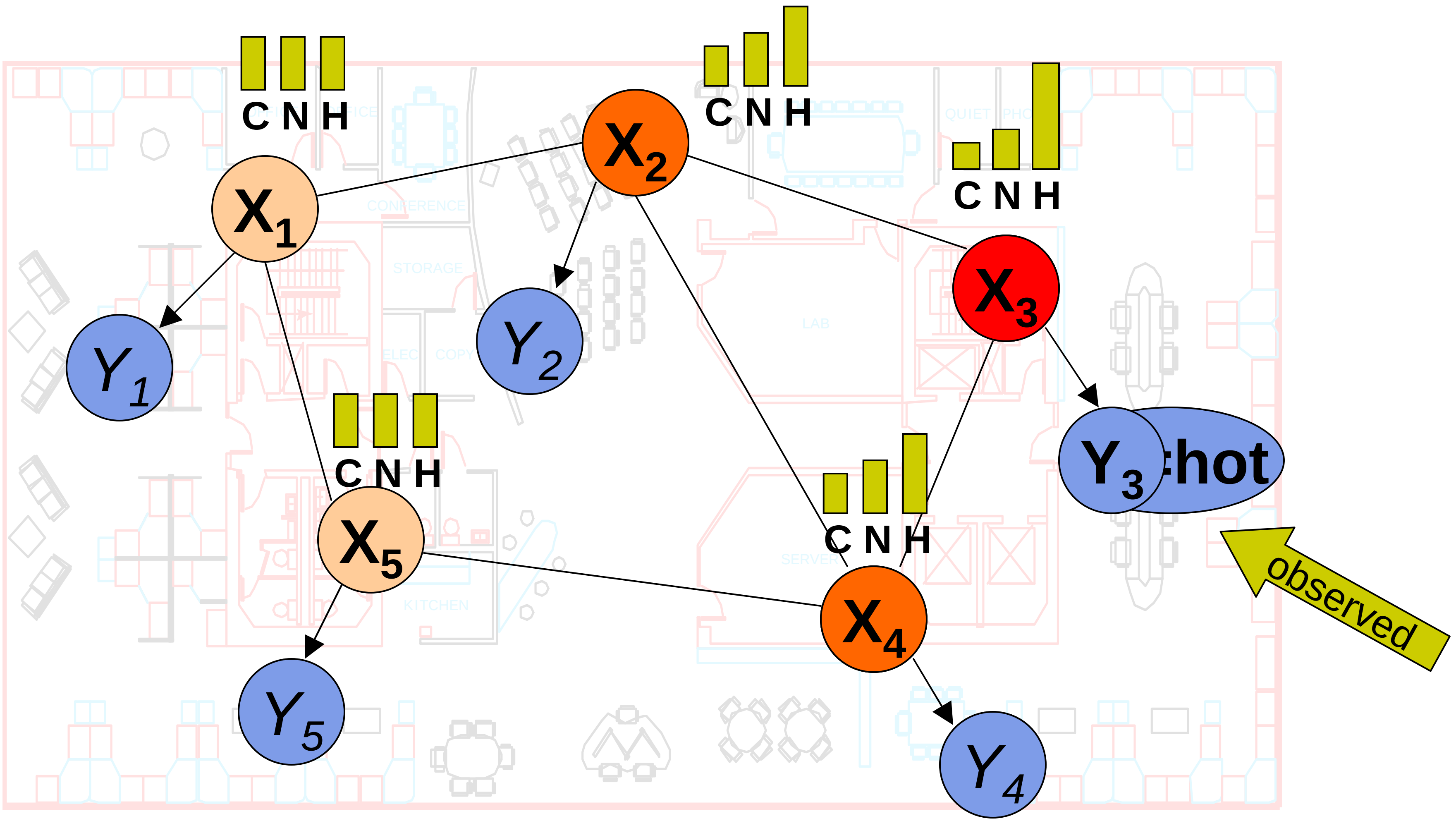
- **Robust reasoning**: Integrate measurements from multiple sensors. E.g.: $P(X_2 \mid y_1, y_2, y_3)$ likely more accurate than $P(X_2 \mid y_2)$
- **Exploiting correlation**: Can predict $P(X_1, X_3 \mid y_2)$ ➔ Can turn some sensors off to save battery life

Making observations



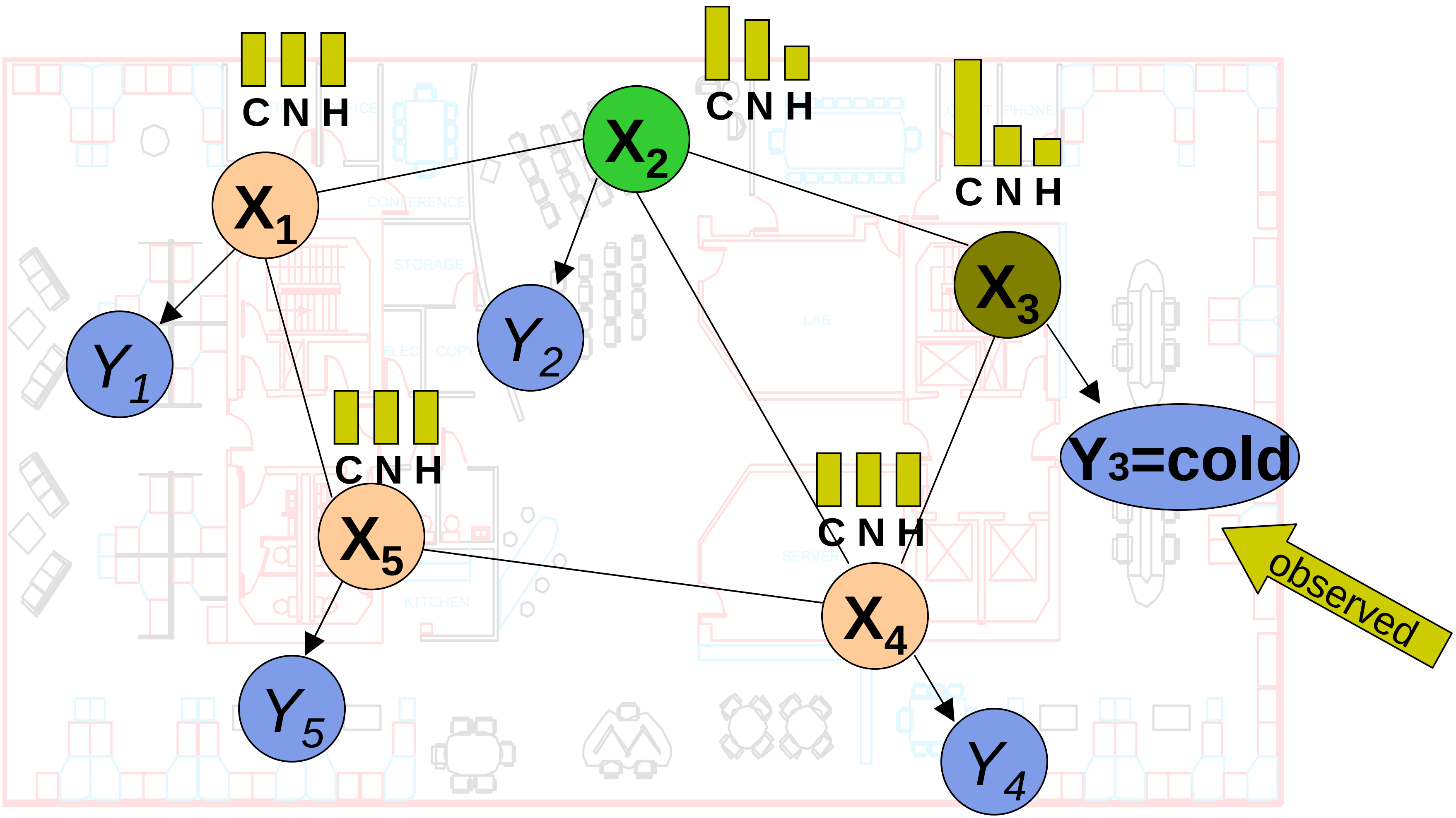
Less uncertain → Reward[P(**X**|Y_1=hot)] = 0.2

Making observations



Reward[P(**X**|Y_3=hot)] = 0.4

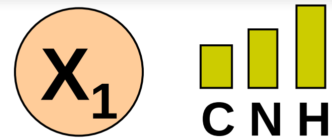
A different outcome…



Reward[P(**X**|Y_3=cold)] = 0.1

Value of information

Should we raise a fire alert?



Actions \ Temp. X	*Fiery hot*	*normal/cold*
No alarm	-$$$	0
Raise alarm	$	-$

Only have belief about temperature P(X = hot | obs)

➔ choose $a^* = \operatorname{argmax}_a \sum_x P(\mathbf{x}|\mathbf{obs})\, U(\mathbf{x},a)$

<u>Decision theoretic value of information</u>

$\text{Reward}[\, P(X \mid \text{obs})\,] = \max_{\mathbf{a}} \sum_x P(\mathbf{x}|\mathbf{obs})\, U(\mathbf{x},a)$

Other example reward functions

Entropy

Reward[P($\mathbf{X}$)] = -H($\mathbf{X}$) = $\sum_x$ P($\mathbf{x}$) $\log_2$ P($\mathbf{x}$)

Expected mean squared prediction error (EMSE)

Reward[P($\mathbf{X}$)] = -1/n $\sum_s$ Var(X_s),

Many other objectives possible and useful…

Value of information [Lindley ’56, Howard ’64]

For any set A of sensors, its value of information is

$$F(A) = \sum_{\mathbf{y_A}} \underbrace{P(\mathbf{y_A})}_{\text{Observations made by sensors } \mathbf{A}} \underbrace{\mathrm{Reward}[P(\mathbf{X} \mid \mathbf{y_A})]}_{\text{Reward when observing } Y_A = y_A}$$

Want to find a set A* µ V, |A*| · k   s.t.

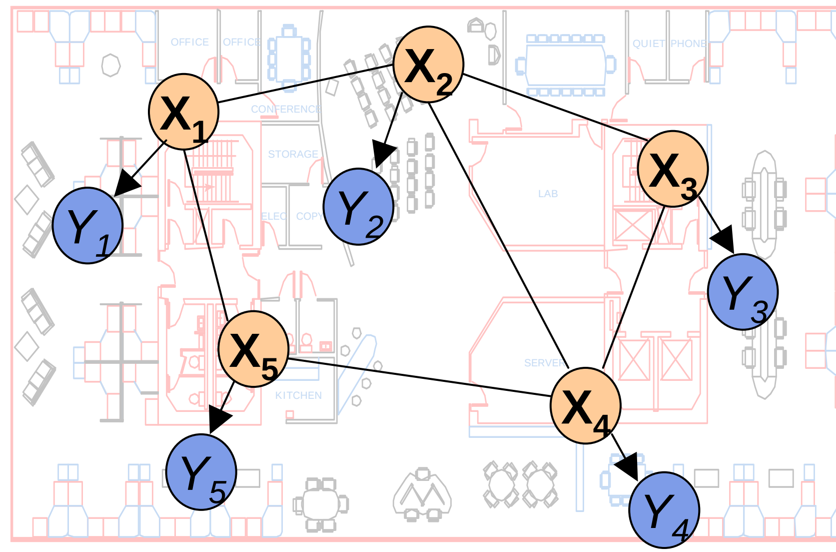
$$A^* = \mathrm{argmax}_{|A| \cdot k}\, F(A)$$

Optimizing Value of Information

- Given: finite set V of locations
- Want: A*µ V such that

$$\mathcal{A}^* = \underset{|\mathcal{A}|\leq k}{\operatorname{argmax}}\, F(\mathcal{A})$$

Typically NP-hard!



<u>Greedy algorithm:</u>

Start with A = ;

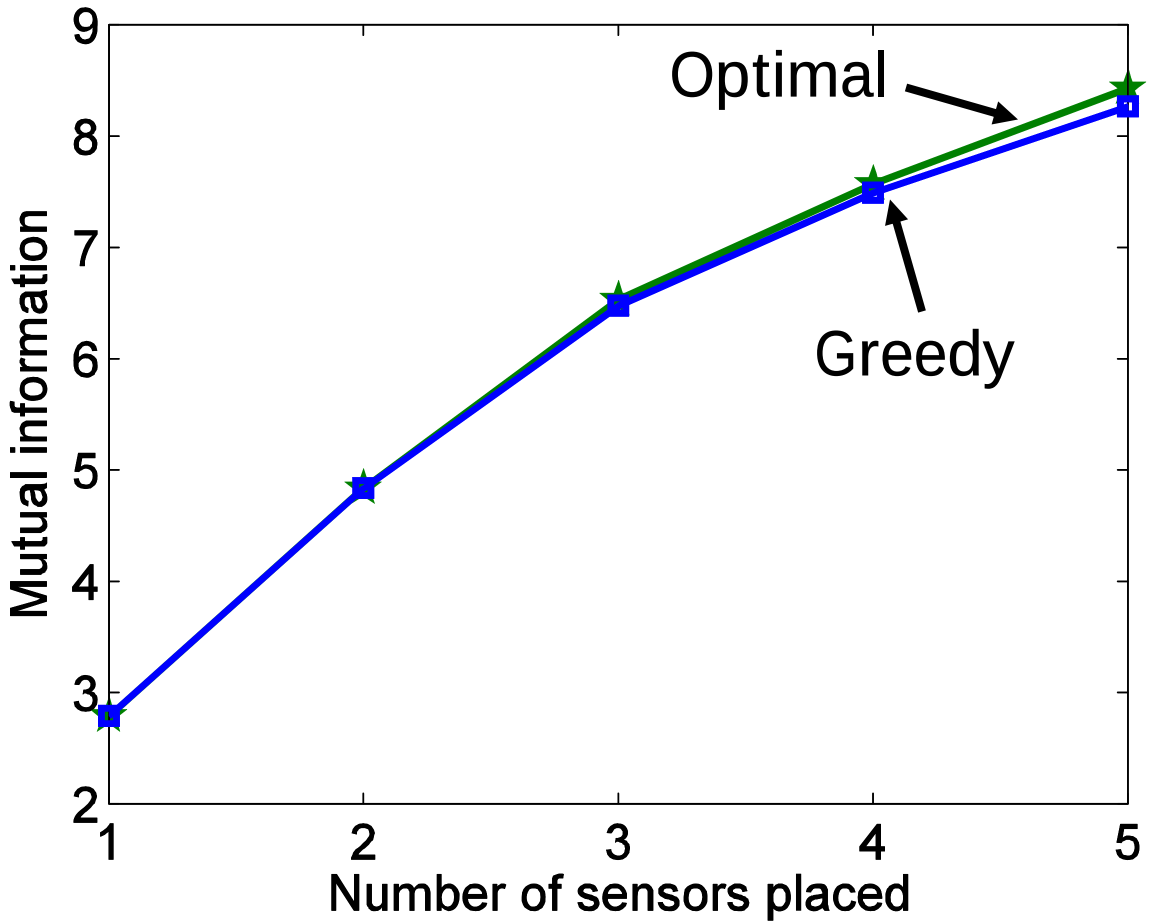
For i = 1 to k

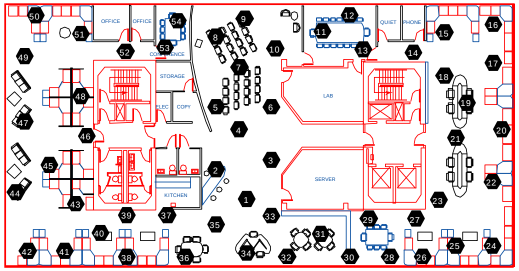
$\quad$ s* := argmax$_s$ F(A [{s})

$\quad$ A := A [{s*}

How well can this simple heuristic do?

Performance of greedy



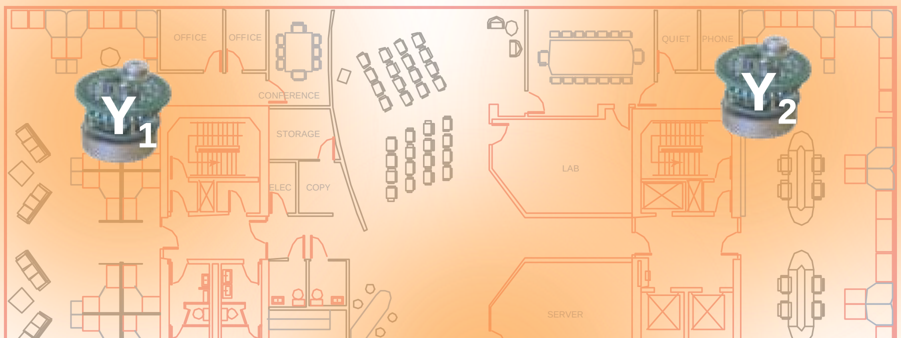


Temperature data from sensor network

- Greedy empirically close to optimal. Why?

Key observation: Diminishing returns

Placement A = $\{Y_1, Y_2\}$



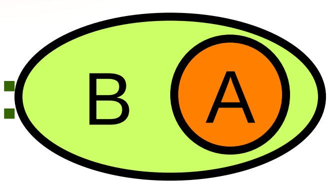
Placement B = $\{Y_1, \ldots, Y_5\}$

Theorem [Krause and Guestrin, UAI '05]**:**

Information gain $F(A) = H(X) – H(X \mid Y_A)$ is submodular!

New sensor Y'

Submodularity:



+ •Y' Large improvement

+ •Y' Small improvement

For A µ B, F(A [{Y'}) – F(A) ¸ F(B [{Y'}) – F(B)

One reason submodularity is useful

Theorem [Nemhauser et al ‘78]

Greedy algorithm gives constant factor approximation

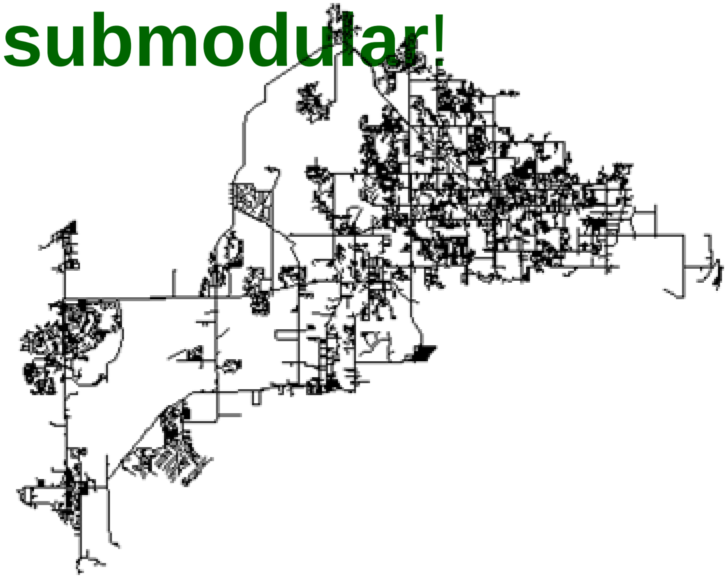
$$F(A_{greedy}) \geq \underbrace{(1-1/e)}_{\sim 63\%} F(A_{opt})$$

- Greedy algorithm gives near-optimal solution!
- For information gain: Guarantees best possible unless P = NP!
  [Krause & Guestrin ’05]

Competition [K, Leskovec, Guestrin, VanBriesen, Faloutsos, J Wat Res Mgt 2008]

- Real metropolitan area network (12,527 nodes)
- Water flow simulator provided by EPA
- 3.6 million contamination events
- Multiple objectives: Detection time, affected population, …
- Place 20 sensors that detect well “on average”

Theorem: All these objectives **submodular**!

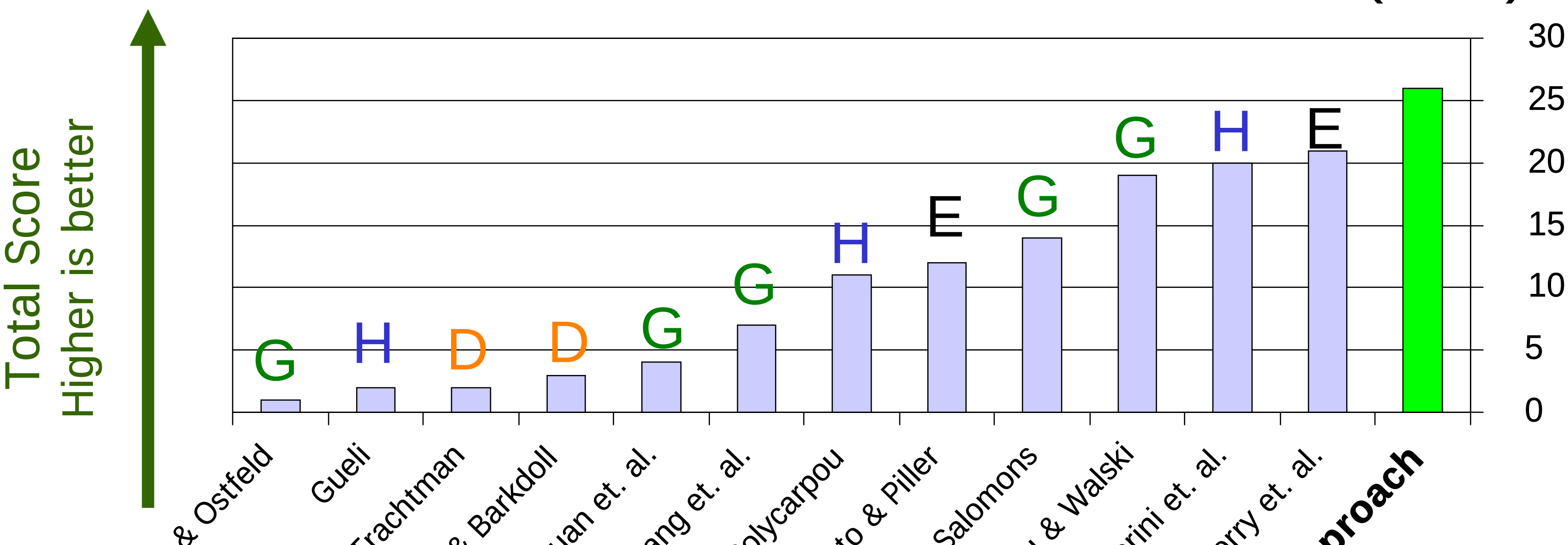


BWSN Competition results

- 13 participants
- Performance measured in 30 different criteria

G: Genetic algorithm D: Domain knowledge

H: Other heuristic E: “Exact” method (MIP)



24% better performance than runner-up! ☺

What was the trick?

Simulated all **3.6M contaminations** on 2 weeks / 40 processors, 16 GB in main memory (compressed)

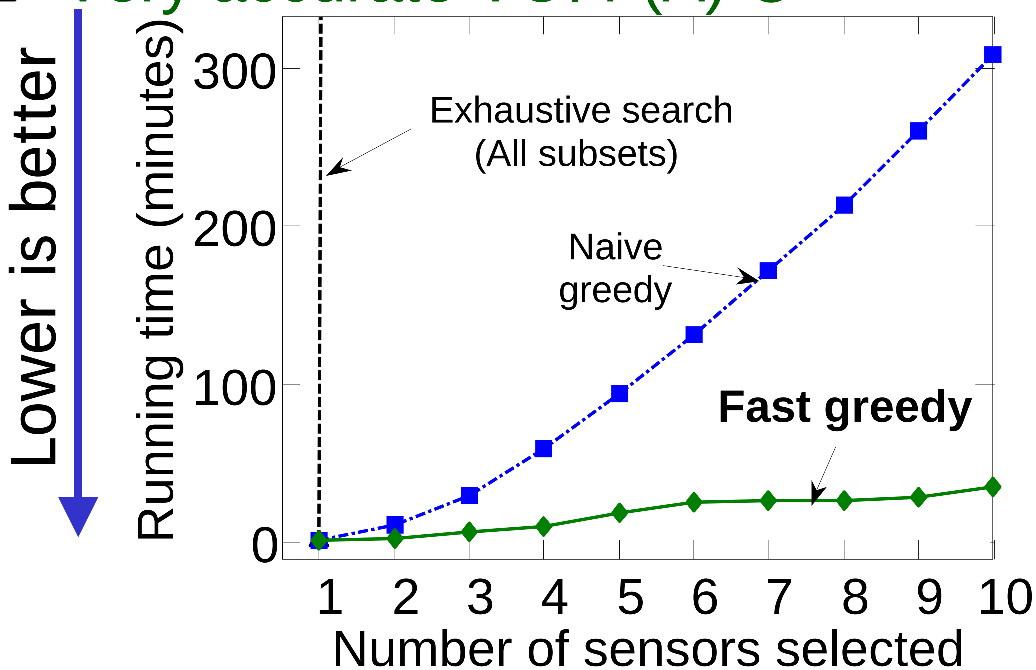
152 GB data on disk

➔ Very accurate VOI F(A) ☺

Very slow evaluation of F(A) ☹

30 hours/20 sensors

6 weeks for all 30 settings ☹



ubmodularity to the rescue:

Using "lazy evaluations":
1 hour/20 sensors

ys! ☺

Advantage through theory and engineering!

Challenges for environmental monitoring

Use robots to monitor environment

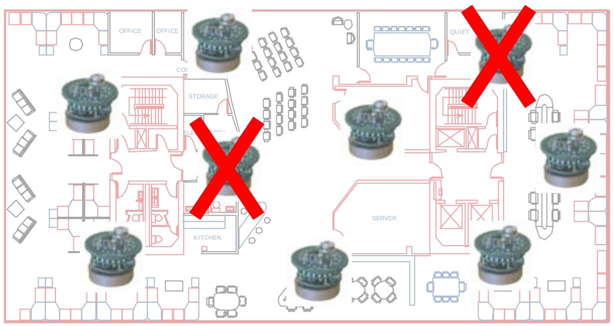
Not just select best k locations A for given F(A). Need to

… be robust against uncertainty in the function F

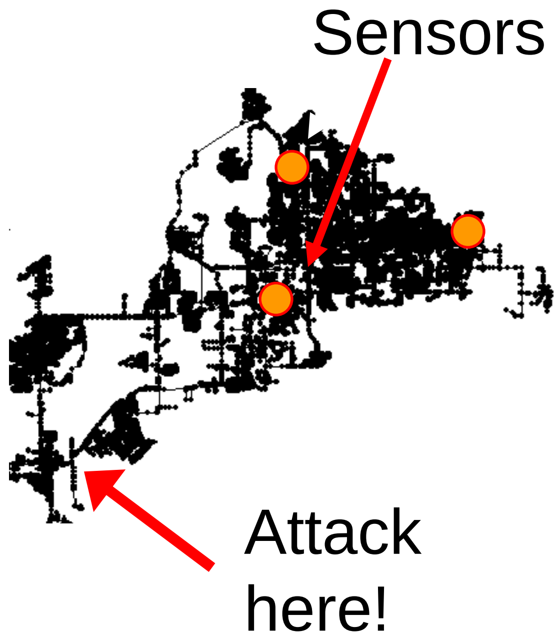
… take into account cost of traveling between locations

… cope with environments that change over time

Why do we need robustness?



→ Sensor failures

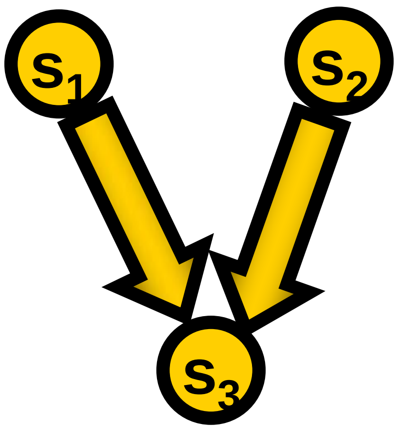


→ Adversarial environments

Unified view: $\mathcal{A}^* = \operatorname*{argmax}_{|\mathcal{A}| \leq k} \min_i F_i(\mathcal{A})$

How does the greedy algorithm do?

$V=\{s_1, s_2, s_3\}$ Buy k=2 sensors F_i = intrusion at s_i



Optimal solution

Greedy picks s_3 first

Then, can choose only s_1 or s_2

Optimal score: 1

Greedy score: ε

➔ Greedy does arbitrarily badly ☹

Can we do better?

The *SATURATE* algorithm

Extremely simple and intuitive algorithm:

Implement using binary search

1. Guess optimal value c
2. Find cheapest solution A s.t. $F_i(A)$¸ c for all i
   (i.e., all objectives F_i are "saturated")

Implement using a greedy algorithm!

Theoretical guarantees

[K, McMahan, Guestrin, Gupta JMLR '08]

Theorem: *SATURATE* finds a solution A_S such that

$$\min_i F_i(A_S) \geq OPT_k \text{ and } |A_S| \leq \alpha k$$

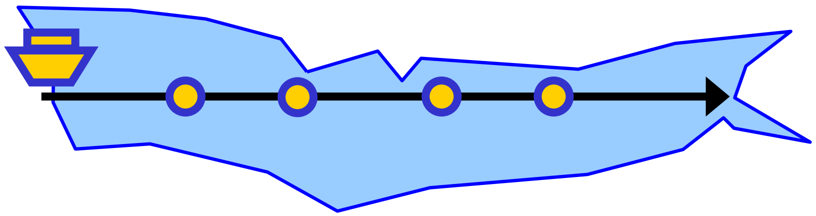
Optimal value achievable using k sensors

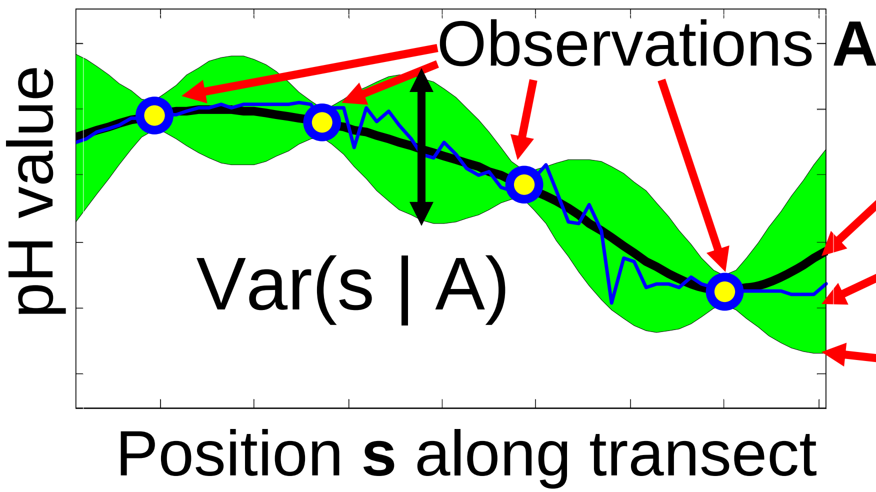
$\alpha > 1$, depends *logarithmically* on problem instance

No better guarantees possible under reasonable complexity theoretic assumptions!

Example: Lake monitoring

- Monitor pH values using robotic sensor





Prediction at unobserved locations

True (hidden) pH values

Use **probabilistic model** (Gaussian processes) to estimate prediction error

Where should we sense to **minimize our maximum error**?

$$A^* = \operatorname{argmin} \max \mathrm{Var}(s \mid A) = \operatorname{argmax} \min_{s} \underbrace{\mathrm{Var}(s) - \mathrm{Var}(s \mid \mathcal{A})}_{\text{(often) submodular}}$$

[Das & Kempe '08]

➔ **Robust sensing problem!**

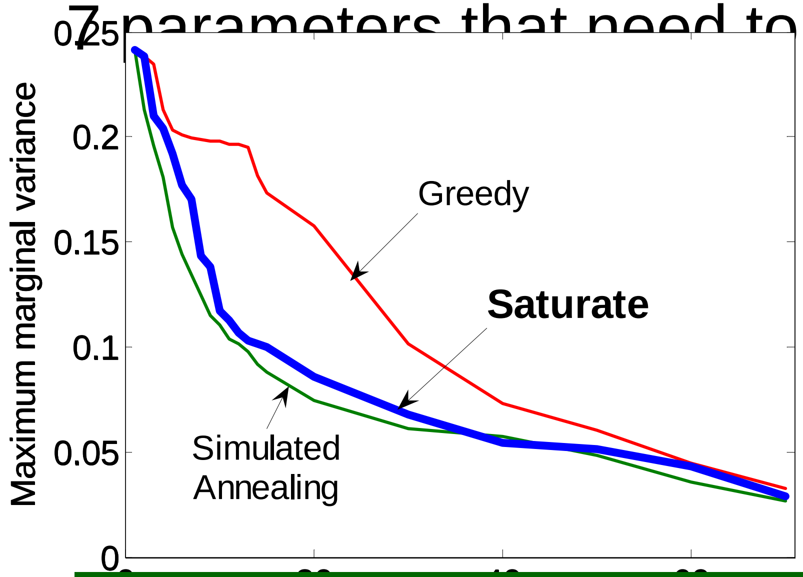
Comparison with state of the art

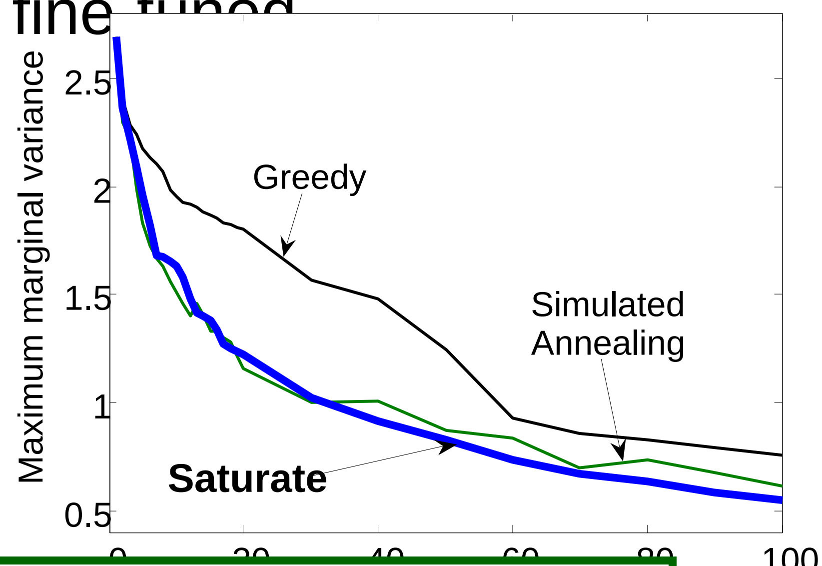
Algorithm used in geostatistics: *Simulated Annealing*

[Sacks & Schiller ’88, van Groeningen & Stein ’98, Wiens ’05,…]

7 parameters that need to be fine tuned

better



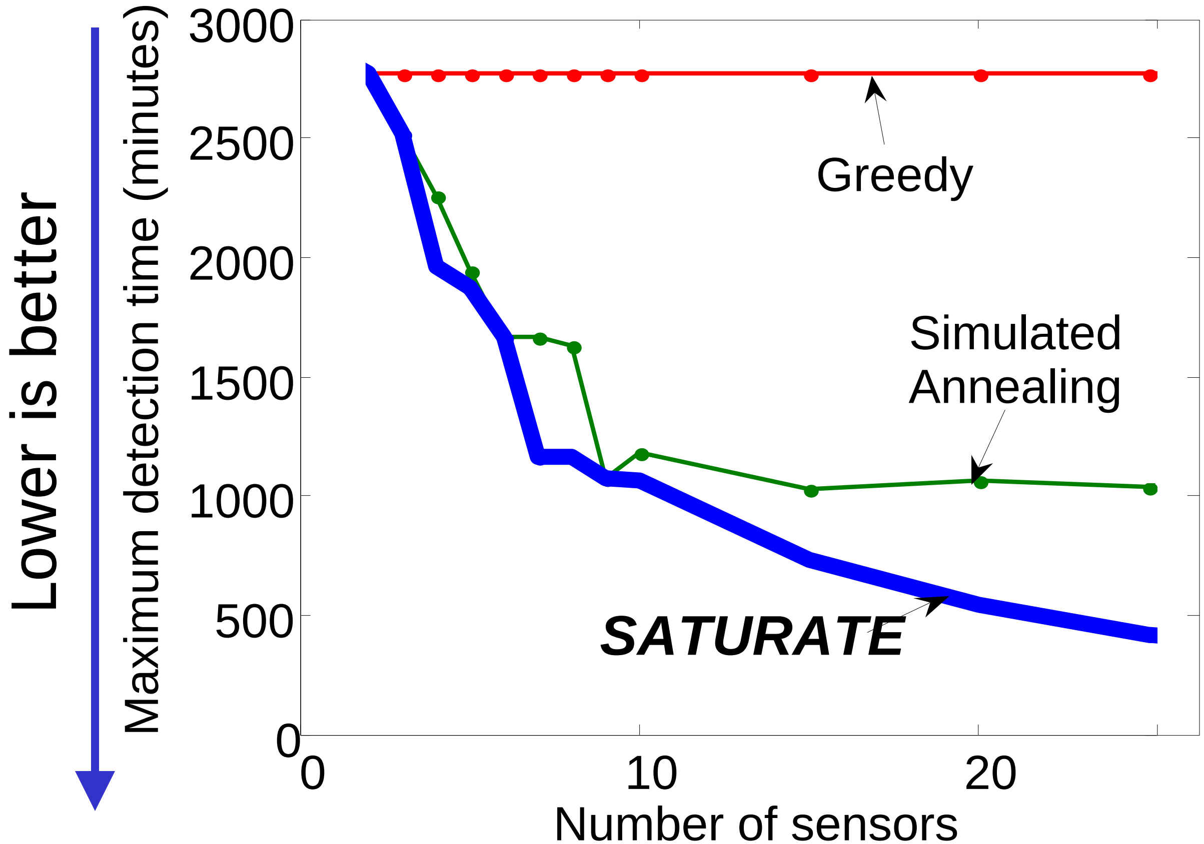


Enviro

***SATURATE* is competitive & 10x faster**

No parameters to tune!

Results on water networks

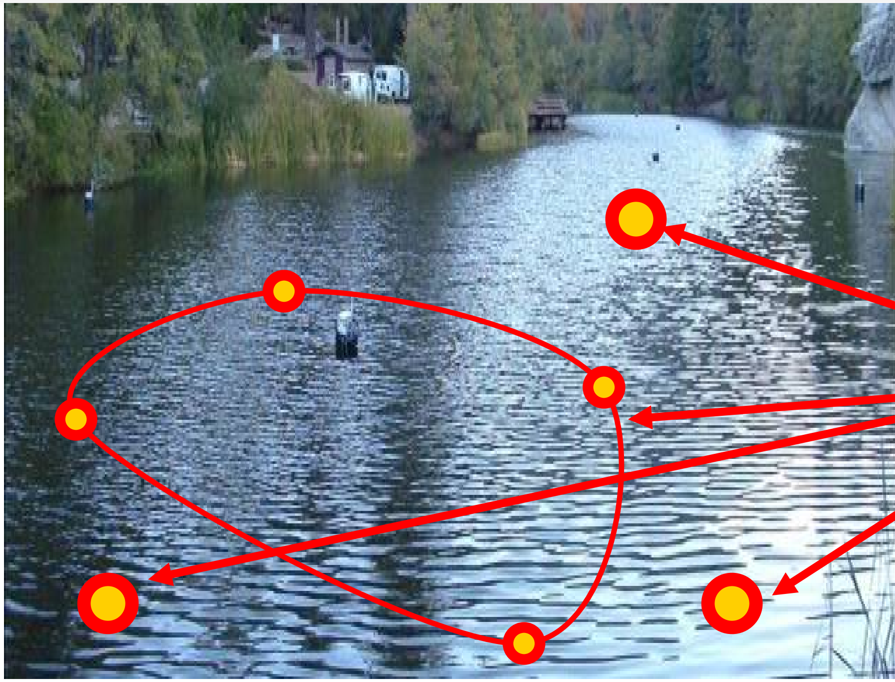


No decrease until **all** contaminations detected!

Water networks

60% lower worst-case detection time!

Informative path planning

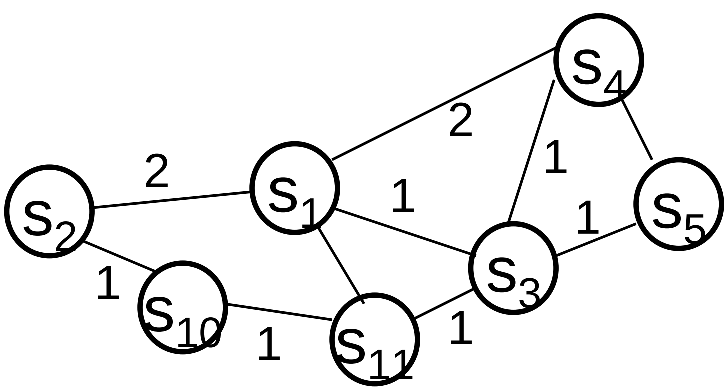


So far:

max F(A) s.t. |A|· k

Most informative locations might be far apart!

Robot needs to travel between selected locations



Locations V nodes in a graph

C(A) = cost of cheapest path connecting nodes A

$$\boxed{\max F(A) \ \text{ s.t. } \ C(A) \cdot B}$$

The **pSPIEL** Algorithm [K, Guestrin, Gupta, Kleinberg IPSN '06]

- ***pSPIEL***: Efficient **nonmyopic** algorithm

  (**p**added **S**ensor **P**lacements at **I**nformative and cost-**E**ffective **L**ocations)

- Select **starting** and **ending** location $\mathbf{s_1}$ and $\mathbf{s}_B$
- **Decompose** sensing region into small, well-separated clusters
- Solve cardinality constrained problem **per cluster** (greedy)
- **Combine** solutions using orienteering algorithm

Guarantees for **pSPIEL** [Singh, K, Kaiser IJCAI '09]

Theorem:
pSPIEL finds a path A with

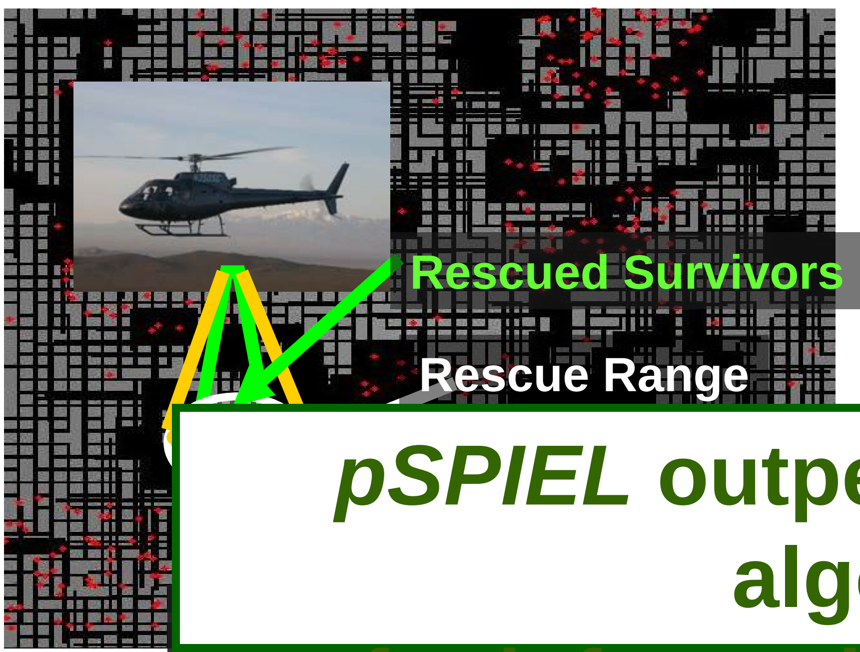
submodular utility $F(A) \geq \Omega(1)\ OPT_F$
path length $C(A) \leq O(1)\ OPT_C$

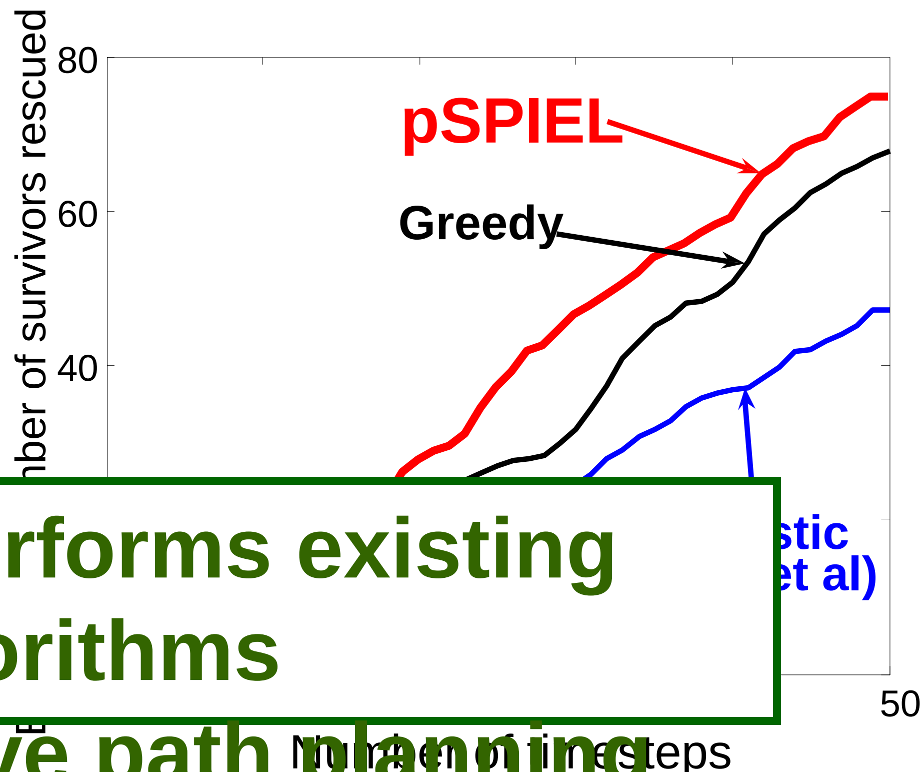
***See store for details**

pSPIEL Results: Search & Rescue

Sensor Planning Research Challenge

- Coordination of multiple mobile sensors to detect survivors of major urban disaster
- Buildings obstruct viewfield of camera
- F(A) = Expected # of people detected





***pSPIEL* outperforms existing algorithms for informative path planning**

Structure in ML / AI problems

ML last 10 years:

Convexity

Kernel machines
SVMs, GPs, MLE…

ML "next 10 years:"

Submodularity
☺

New structural properties

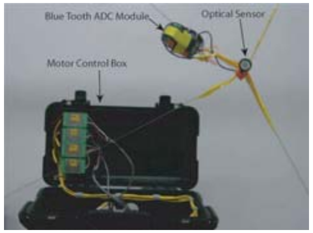
- Structural insights help us solve challenging problems
- **Shameless plug:** → www.submodularity.org
  - MATLAB Toolbox for optimizing submodular functions
  - Tutorial slides, References & Video
  - **"Intelligent Information Gathering and Submodular**

Conclusions

- Real sensing problems create challenges such as

  Robustness, complex constraints, dynamic environments

- Can exploit submodularity to find provably good solutions
- Presented algorithms with strong guarantees
- Perform well on real world problems



Thanks to:

Carlos Guestrin, Anupam Gupta, Jon Kleinberg, Jure Leskovec, Amarjeet Singh, William Kaiser, Jeanne VanBriesen, Christos Faloutsos, Brendan McMahan