Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks

Maurice Chu, Horst Haussecker and Feng Zhao

Based upon slides by:

Jigesh Vora, UCLA

Overview

- Introduction
- Sensing Model
- Problem Formulation & Information Utility
- Algorithm for Sensor Selection and Dynamic Routing
- Experiments
- Rumor Routing

Introduction

- how to dynamically query sensors and route data in a network so that information gain is maximized while power and bandwidth consumption is minimized
- two algorithms:
  - Information-Driven Sensor Querying (IDSQ)
  - Constrained Anisotropic Diffusion Routing (CADR)

What's new?

- Introduction of two novel techniques IDSQ and CADR for energy-efficient data querying and routing in sensor networks
- the use of general form of information utility that models the information content as well as the spatial configuration of a network
- generalization of directed diffusion that uses both the communication cost and the information utility to diffuse data

Problem Formulation

- $\mathbf{z}_i(t) = \mathbf{h}(\mathbf{x}(t),\ \ _i(t)),$ (1)

  $\mathbf{x}(t)$ is based on parameters of the sensor, $_i(t)$ and $\mathbf{z}_i(t)$ are characteristics and measurement of sensor *i* respectively.

- for sensors measuring sound amplitude

  $_i = [\ x_i\ ,\ \sigma_i^{\ 2}\ ]^{T}$ (3)

  x_i is the known sensor position and $\sigma_i^{\ 2}$ is the known additive noise variance

  $z_i = a\ /\ ||\ \mathbf{x}_i - \mathbf{x}\ ||^{\ /2} + w_i\ ,$ (4)

  a is target amplitude, is attenuation coefficient , w_i is Gaussian noise with variance $\sigma_i^{\ 2}$

Define Belief as ...

- representation of the current *a posteriori* distribution of **x** given measurement $\mathbf{z}_1, \ldots, \mathbf{z}_N$: $p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N)$
- expectation is considered estimate

$$\bar{\mathbf{x}} = \quad \mathbf{x} p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N) d\mathbf{x}$$

- covariance approximates residual uncertainty

$$= \quad (\mathbf{x} - \mathbf{x})(\mathbf{x} - \mathbf{x})^T p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N) d\mathbf{x}$$

Define Information Utility as …

The Information Utility function is defined as

Ψ: $P(R^d) \rightarrow R$

d is the dimension of x

Ψ assigns value to each element of $P(R^d)$ *indicating the uncertainty of the distribution*

smaller value -> more spread out distribution

larger value -> tighter distribution

Sensor Selection (in theory)

- $j_0 = \arg_{j \; A} \max \; (p(\mathbf{x}|\{\mathbf{z}_i\}_{i \; U} \; \{\mathbf{z}_j\}))$
  - $A = \{1, \ldots, N\} - U$ is set of sensors whose measurements not incorporated into belief
  - is information utility function defined on the class of all probability distributions of **x**
  - intuitively, select sensor *j* for querying such that information utility function of the distribution updated by $\mathbf{z}_j$ is maximum

Sensor Selection (in practice)

- $\mathbf{z}_j$ is unknown before it's sent back
- best average case

  $j' = \arg_{j\ \ A} \max E_{\mathbf{z}_j}[\ \ (p(\mathbf{x}|\{\mathbf{z}_i\}_{i\ \ U}\ \ \ \{\mathbf{z}_j\}))|$
  $\{\mathbf{z}_i\}_{i\ \ U}\ ]$
- maximizing worst case

  $j' = \arg_{j\ \ A} \max \min_{\mathbf{z}_j}\ \ (p(\mathbf{x}|\{\mathbf{z}_i\}_{i\ \ U}\ \ \ \{\mathbf{z}_j\}))$
- maximizing best case

  $j' = \arg_{j\ \ A} \max \max_{\mathbf{z}_j}\ \ (p(\mathbf{x}|\{\mathbf{z}_i\}_{i\ \ U}\ \ \ \{\mathbf{z}_j\}))$

Sensor Selection Example

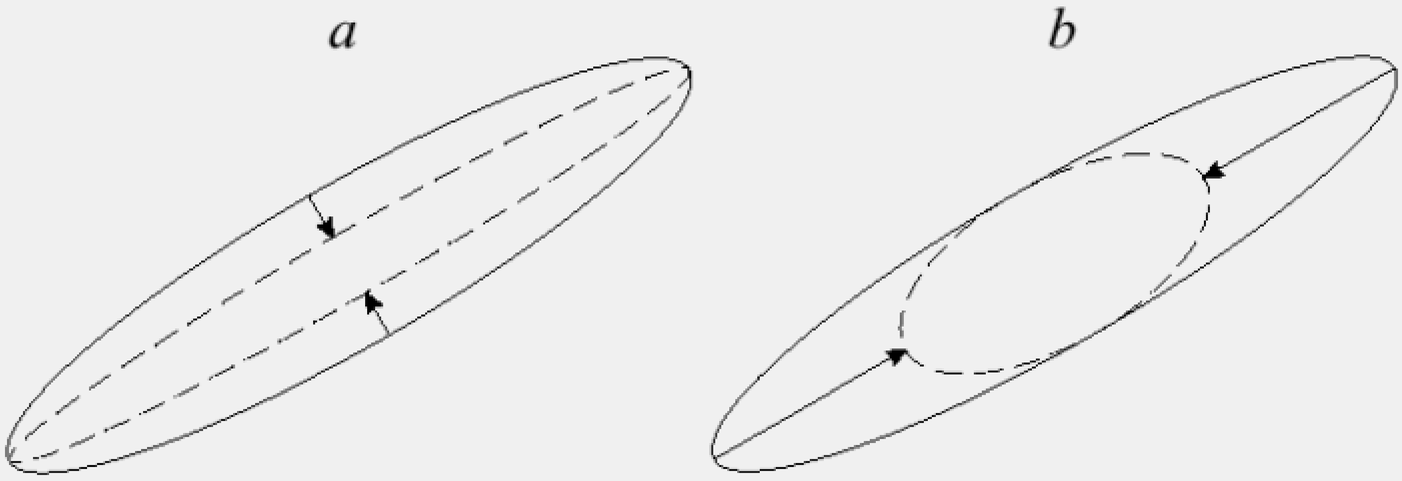


Figure 1: Illustration of sensor selection criteria based upon the information gain of individual sensor contributions. Here, the information gain is measured by the reduction in the error ellipsoid.

Information Utility Measures

- covariance-based

  $(p_X) = -\det(\ \ ), \quad (p_X) = -\mathrm{trace}(\ \ )$

- Fisher information matrix

  $(p_X) = \det(\mathbf{F}(\mathbf{x})), \quad (p_X) = \mathrm{trace}(\mathbf{F}(\mathbf{x}))$

- entropy of estimation uncertainty

  $(p_X) = -H(P), \quad (p_X) = -h(p_X)$

Information Utility Measures (Contd ...)

- volume of high probability region

  = {**x** S : p(**x**) }, chose so that P( ) = , is given

  (p_X) = - vol( )

- sensor geometry based measures

  in cases utility is function of sensor location only

  (p_X) = - ($\mathbf{x}_i$-$\mathbf{x}_0$)^T $^{-1}$($\mathbf{x}_i$-$\mathbf{x}_0$), where $\mathbf{x}_0$ is the mean of the current estimate of target location also called *Mahalanobis distance*

Composite Objective Function

- $M_c(\ _l, \ _j, p(\mathbf{x}|\{\mathbf{z}_i\}_{i\ U})$
$= \ M_u(p(\mathbf{x}|\{\mathbf{z}_i\}_{i\ U}, \ _j) - (1 - \ )M_a(\ _l, \ _j)$
    - M_u is information utility measure
    - M_a is communication cost measure
    - [0, 1] balances their contributions
    - $_j$ is characteristics of current sensor j
- $j_0 = \arg_{j\ A} \max M_c(\ _i, \ _j, p(\mathbf{x}|\{\mathbf{z}_i\}_{i\ U})$

Incremental Update of Belief

- $p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N)$
$= c\, p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_{N-1})\, p(\mathbf{z}_N \mid \mathbf{x})$
  - $\mathbf{z}_N$ *is the new measurement*
  - $p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_{N-1})$ is previous belief
  - $p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N)$ is updated belief
  - c is normalizing constant
- for linear system with Gaussian distribution, *Kalman filter* is used

IDSQ Algorithm

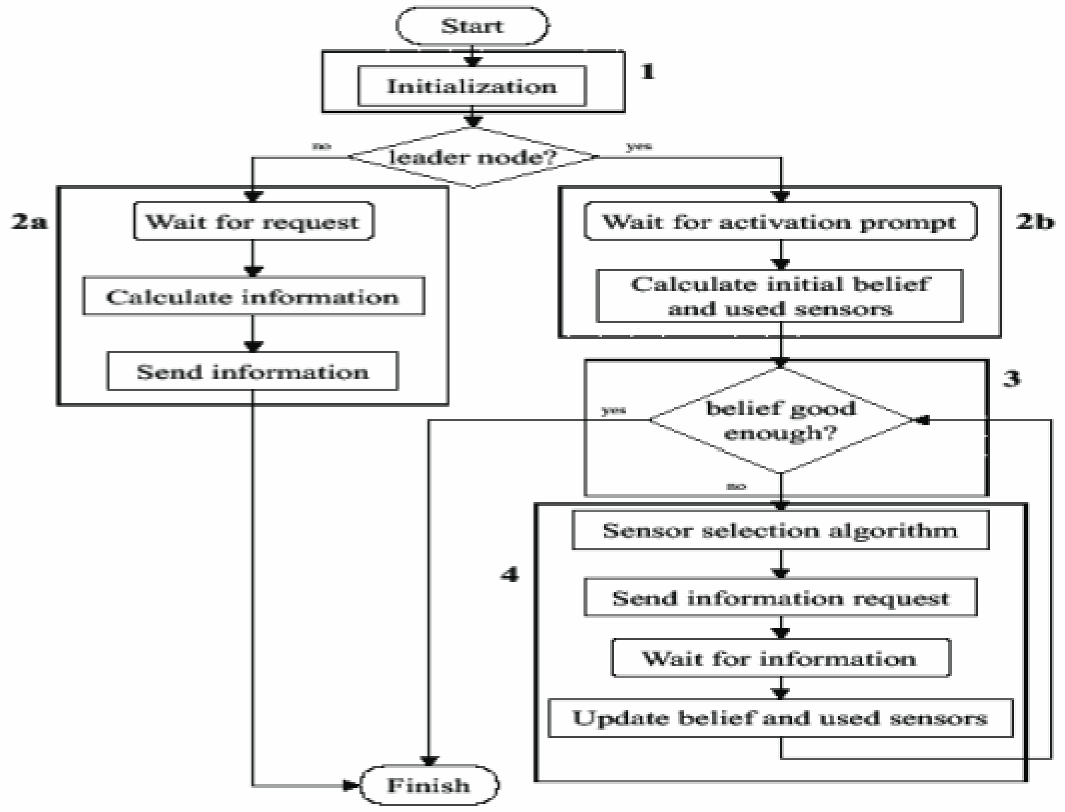


Figure 2: Flowchart of the information-driven sensor querying algorithm for each sensor.

CADR Algorithm

- global knowledge of sensor positions

- with global knowledge of sensor positions
- optimal position to route query to is given by

  $\mathbf{x}_o = \arg_{\mathbf{x}} [\ M_c = 0]$ (10)
- The routing is directly addressed to the sensor node that is closest to the optimal position

CADR Algorithm

- no global knowledge of sensor position

1. $j' = \arg_j \max(M_c(\mathbf{x}_i))$, j k
2. $j' = \arg_j \max((\ M_c)^T(\mathbf{x}_k - \mathbf{x}_j) / (|\ M_c||\mathbf{x}_k - \mathbf{x}_j|))$,
3. instead of following M_c only, follow

   $d = \quad M_c + (1 - \quad)(\mathbf{x}_o - \mathbf{x}_j)$,

   $= (|x_0 - x_j|^{-1})$

   for large distance to $\mathbf{x}_o$, follow $(\mathbf{x}_o - \mathbf{x}_k)$

   for small distance to $\mathbf{x}_o$, follow M_c

IDSQ Experiments

- Sensor Selection Criteria

- A. *nearest neighbor data diffusion*

$$j_0 = \arg_{j\ \{1, \ldots, N\}-U} \min ||\mathbf{x}_l - \mathbf{x}_j||$$

- B. *Mahalanobis distance*

$$j_0 = \arg_{j\ \{1, \ldots, N\}-U} \min (\mathbf{x}_i - \mathbf{x}_0)\ ^{-1}(\mathbf{x}_i - \mathbf{x}_0)$$

- C. maximum likelihood

$$j_0 = \arg_{j\ \{1, \ldots, N\}-U} \min p(\mathbf{x}_i |\ )$$

- D. best feasible region, upper bound

http://www.itl.nist.gov/div898/software/dataplot/refman2/auxillar/matrdist.htm

IDSQ Experiments

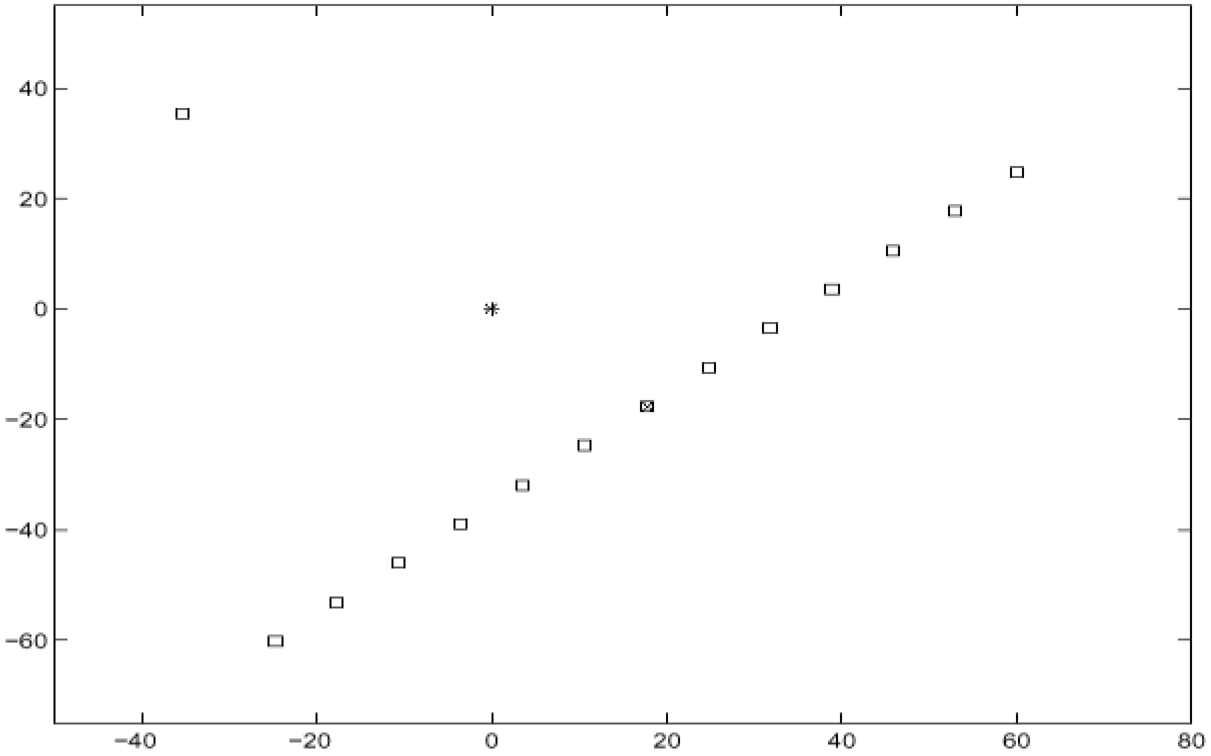


Figure 3: Layout of all-but-one-colinear sensors (squares) and target (asterisk). The leader node is denoted by an $\times$.

IDSQ Experiments

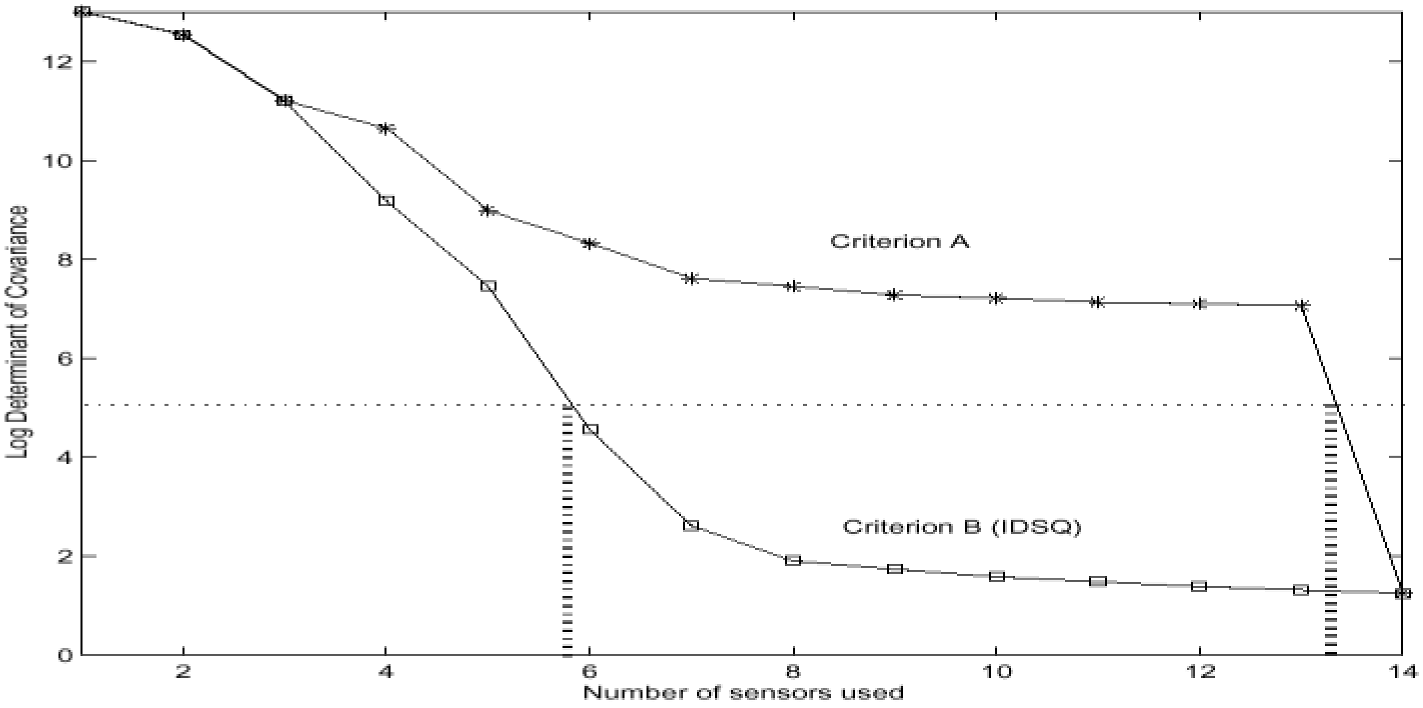
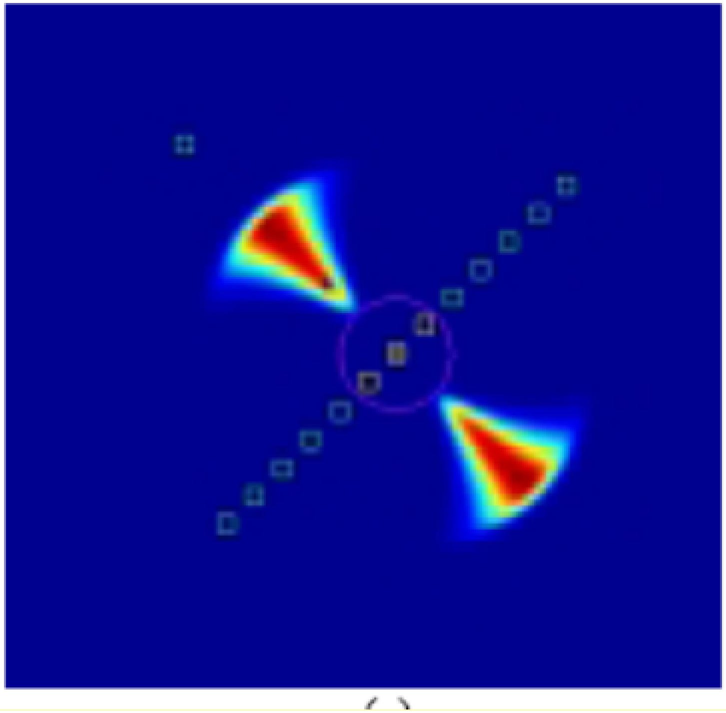
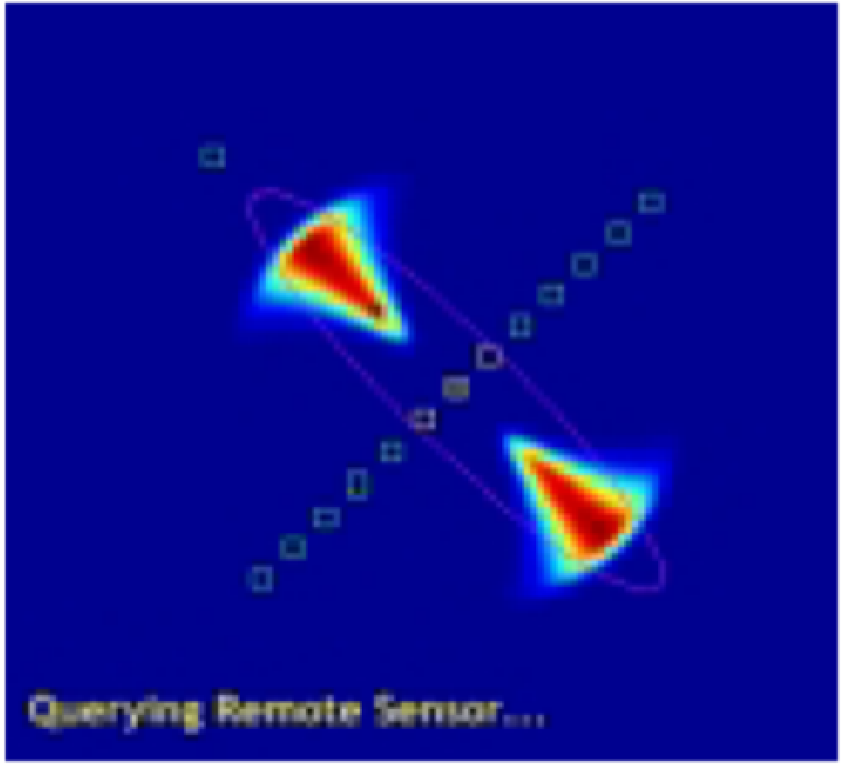


Figure 6: Determinant of the error covariance for selection criteria **A** and **B** (IDSQ) for the all-but-one-colinear sensor layout. **A** tasks 14 sensors while **B** tasks 6 sensors to be below an error threshold of 5 units.

IDSQ - Comparison of NN and Mahalanobis distance method



NN distance method



Mahalanobis distance method

IDSQ Experiments

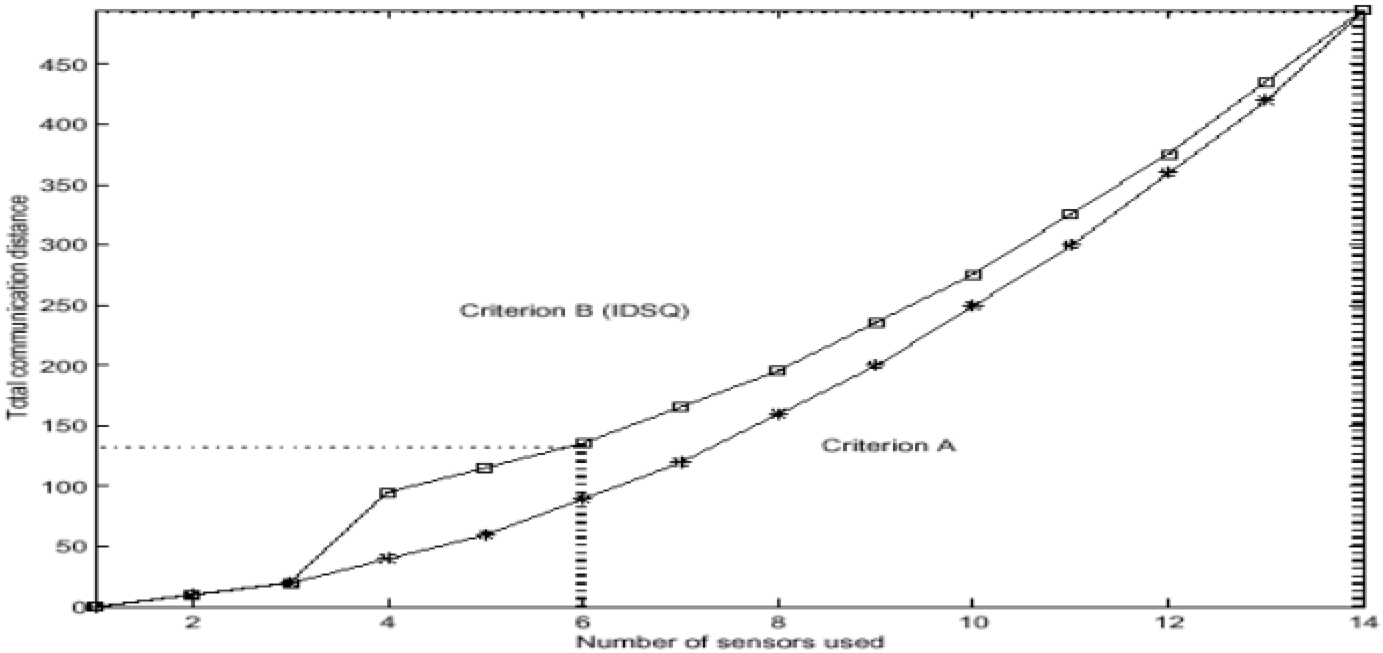


Figure 7: Total communication distance vs. the number of sensors queried for selection criteria **A** and **B** (IDSQ) for the all-but-one-colinear sensor layout. For achieving the same threshold of the error, **A** tasks 14 sensors and uses nearly 500 units of communication distance whereas **B** tasks 6 sensors and uses less than 150 units of communication distance.

IDSQ Experiments

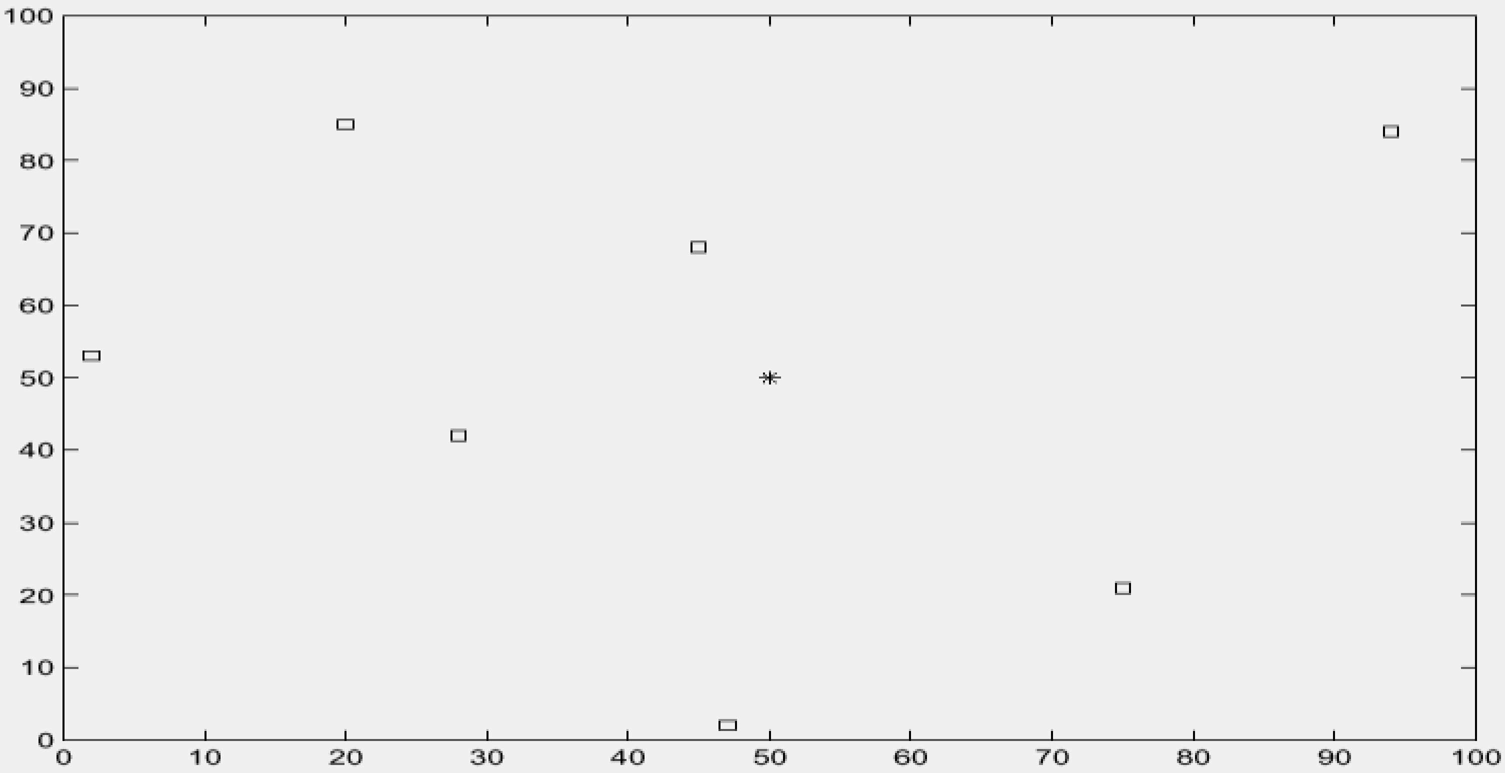


Figure 8: Layout of seven randomly placed sensors (squares) with target in the middle (asterisk).

IDSQ Experiments

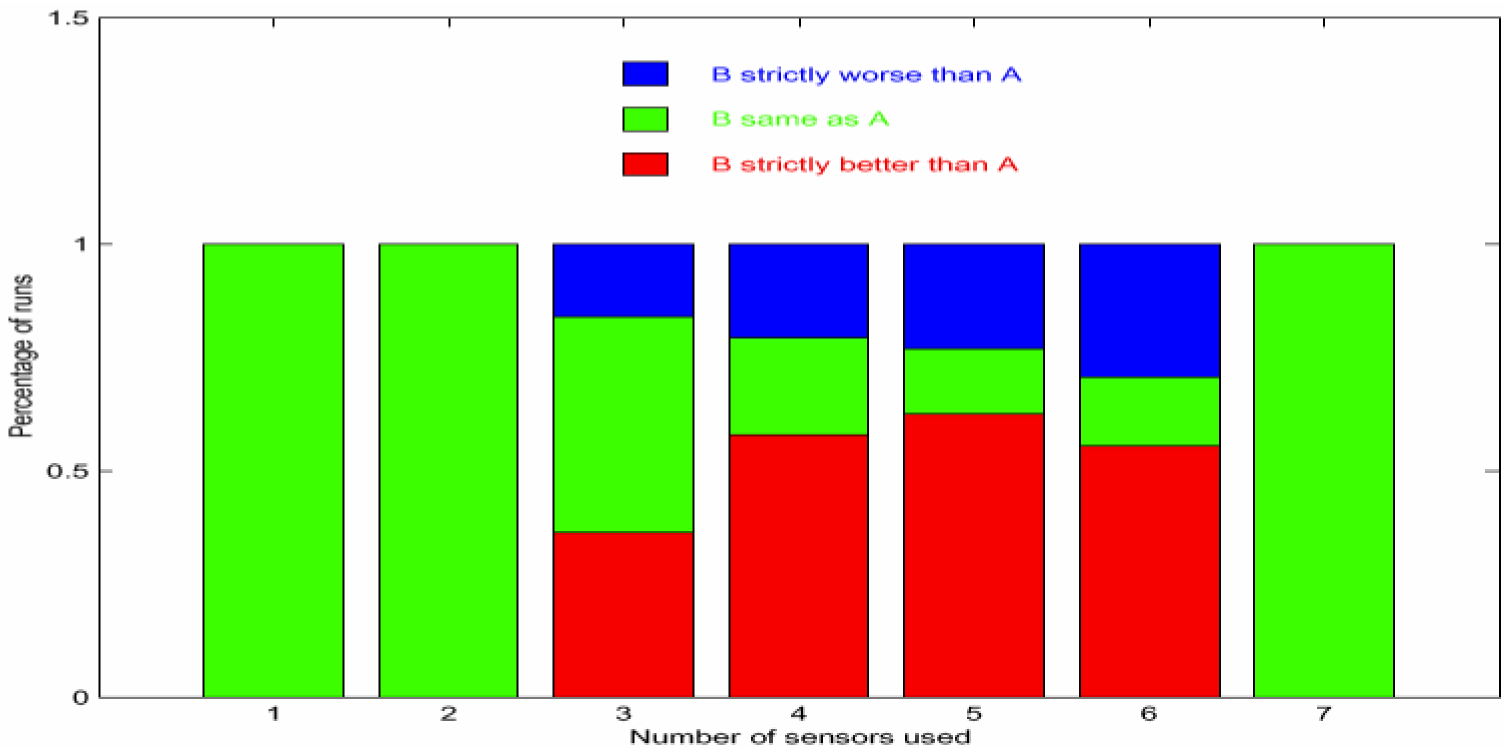


Figure 9: Percentage of runs where **B** performs better than **A** for seven randomly placed sensors.

B uses the Mahalanobis distance, A is Nearest Neighbor Diffusion. “Performs Better” means less error for given number of sensors used.

IDSQ Experiments

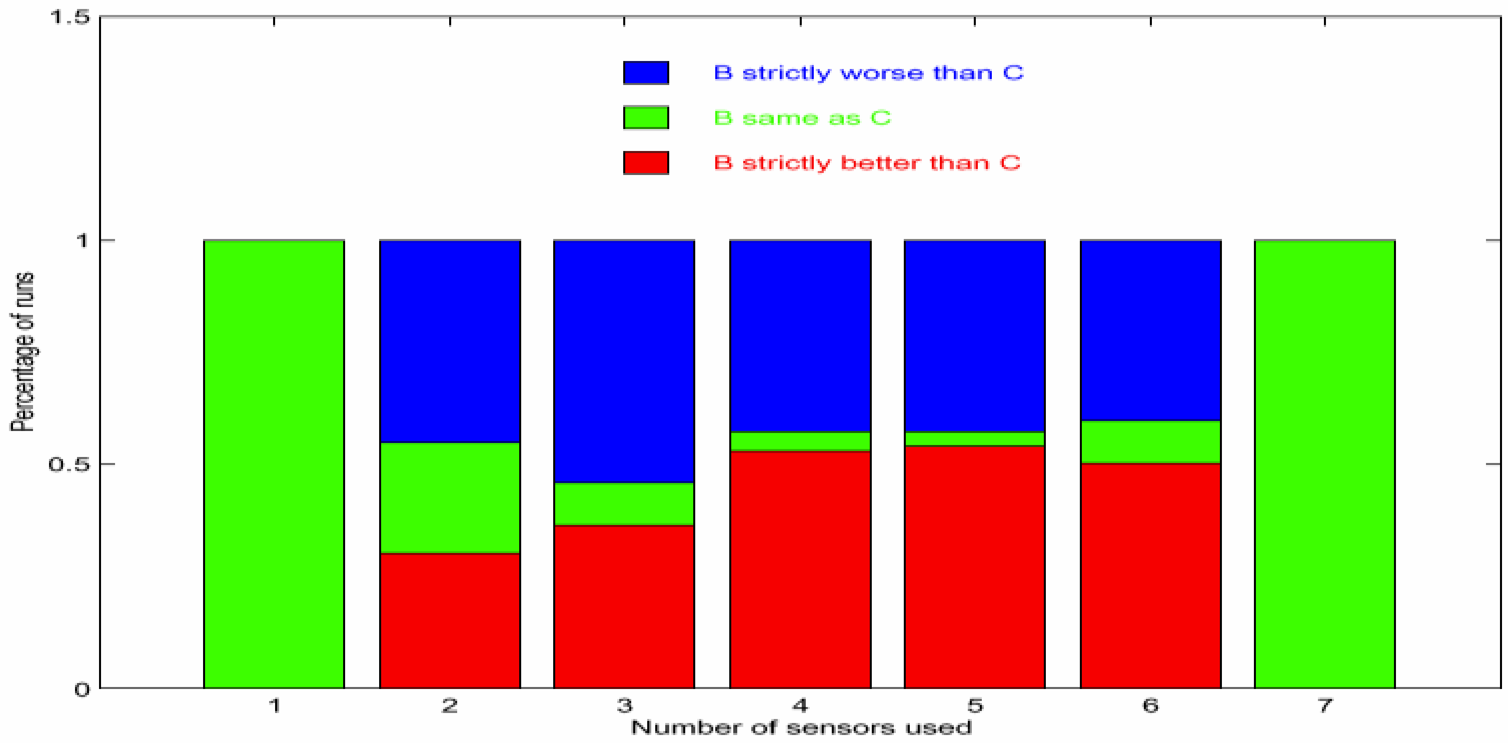


Figure 10: Percentage of runs where **B** performs better than **C** for seven randomly placed sensors.

C uses maximum likelihood.

IDSQ Experiments

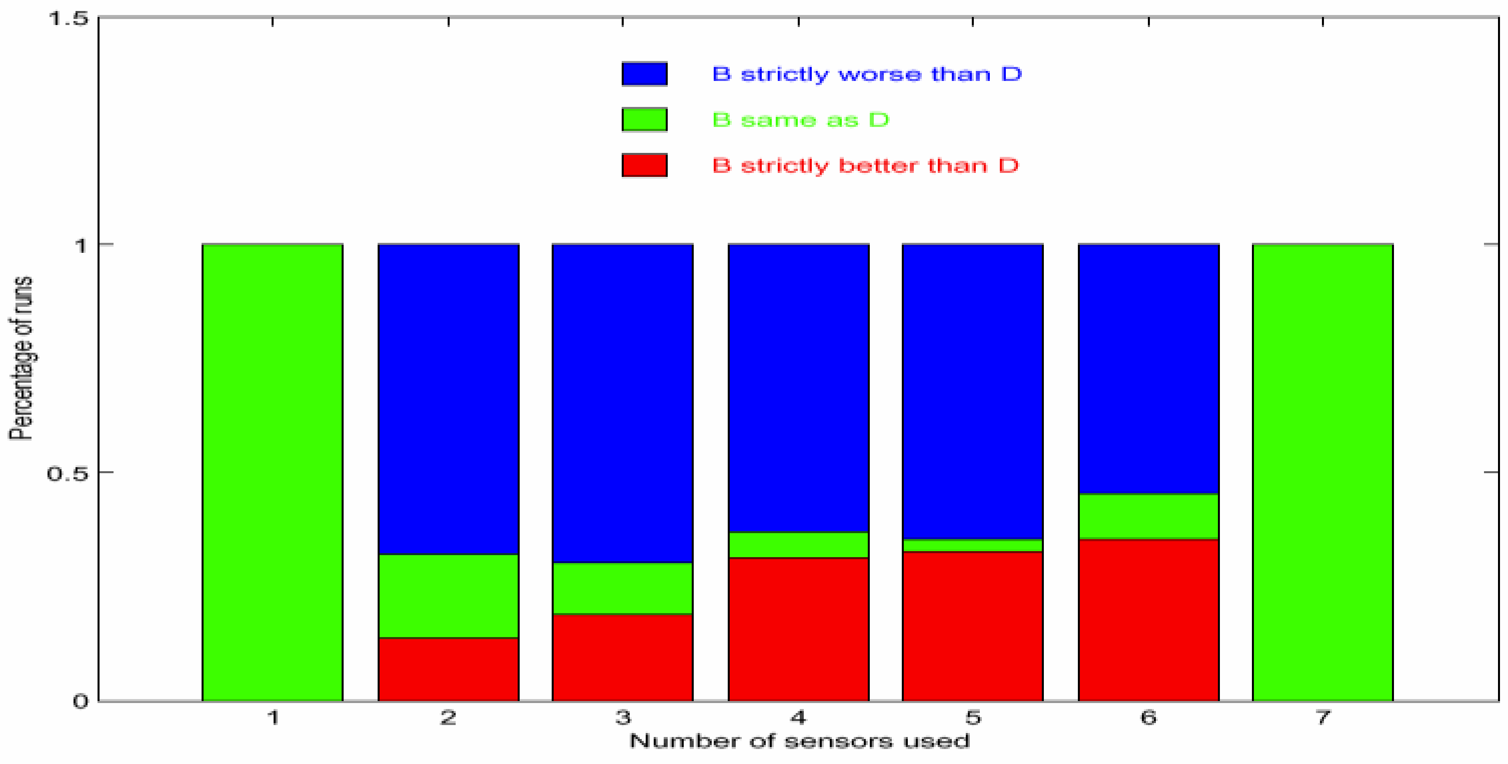
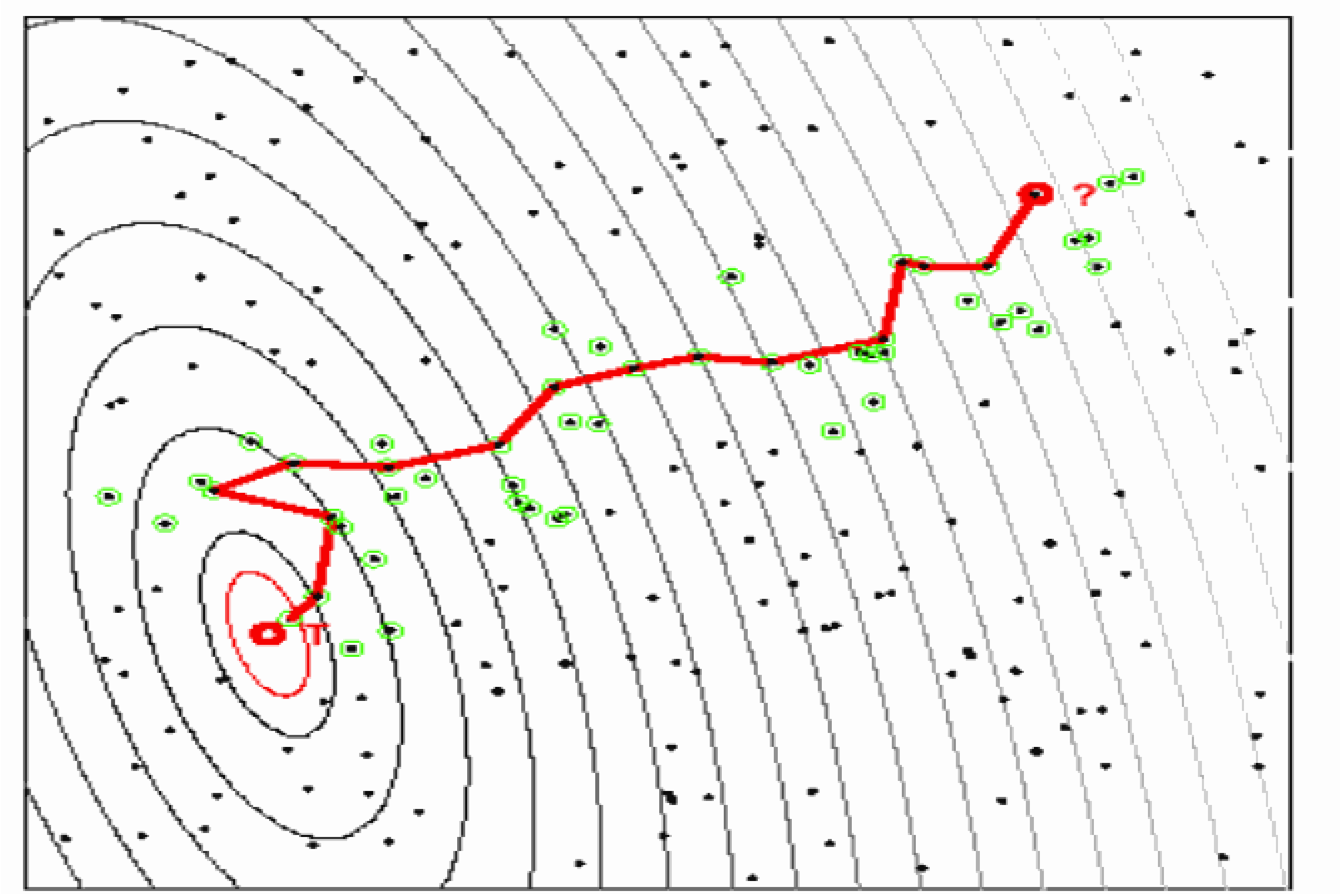


Figure 11: Percentage of runs where **B** performs better than **D** for seven randomly placed sensors.

D is the best feasible region approach. Requires knowledge of sensor data **before** querying.

CADR Experiments

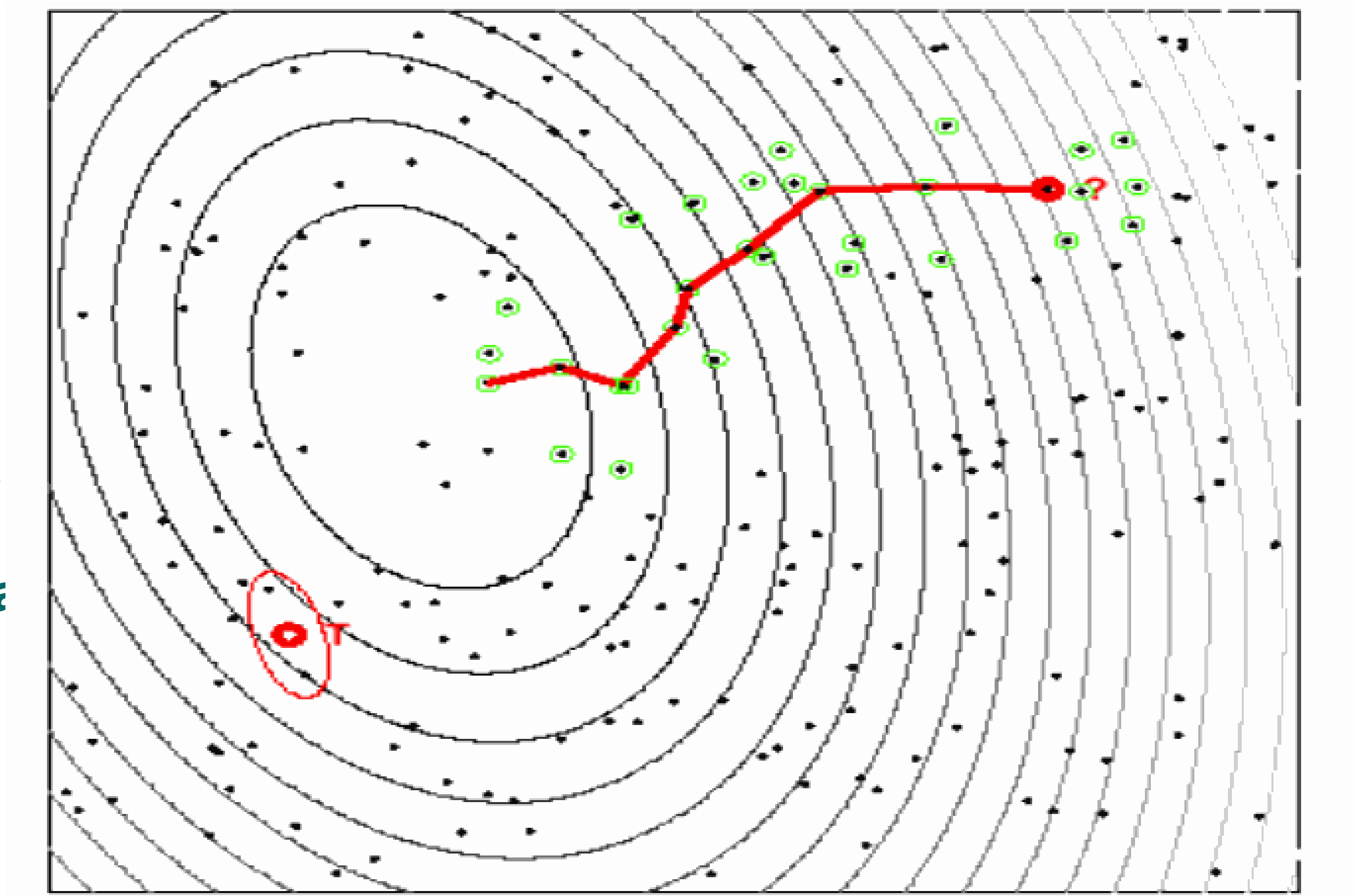
Figure 12-1



- $M_c = \quad M_u - (1 - \quad) M_a$
- $= 1$

CADR Experiments

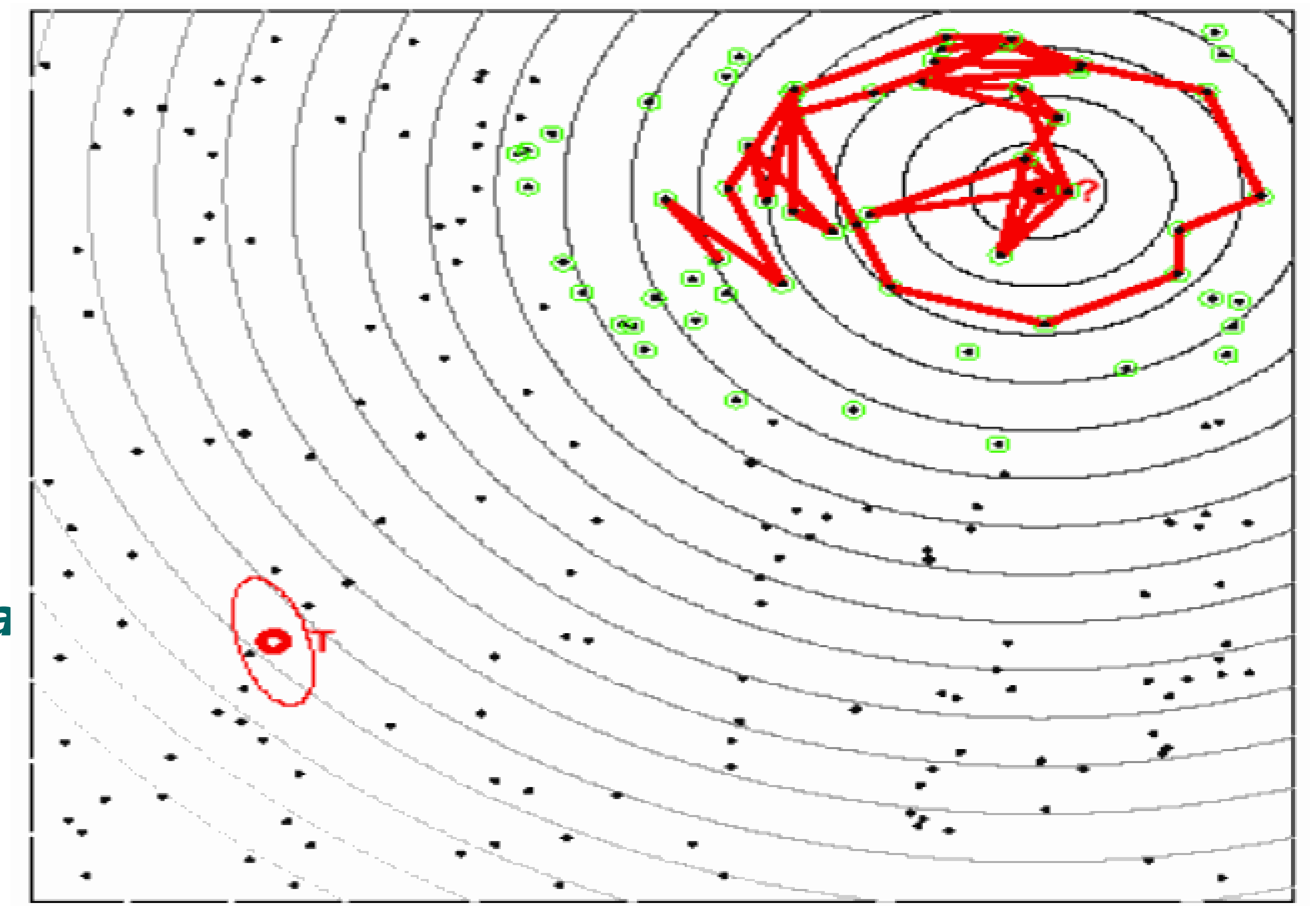
Figure 12-2



- $M_c = \quad M_u - (1 - \quad) M_a$
- = 0.2

CADR Experiments

Figure 12-3



- $M_c = M_u - (1 - \ )M_a$
- $= 0$

Critical Issues

- Use of Directed Diffusion to implement IDSQ/CADR
- Belief Representation
- Impact of Choice of Representation
- Hybrid Representation

Belief Representation

- *Parametric*, where each distribution is described by a set of parameters, poor quality but light-weighted e.g. Gaussian distributions
- *Non-parametric*, where each distribution is approximated by point samples, more accurate but more costly e.g. grid approach or histogram type approach

Impact of Choice of Representation

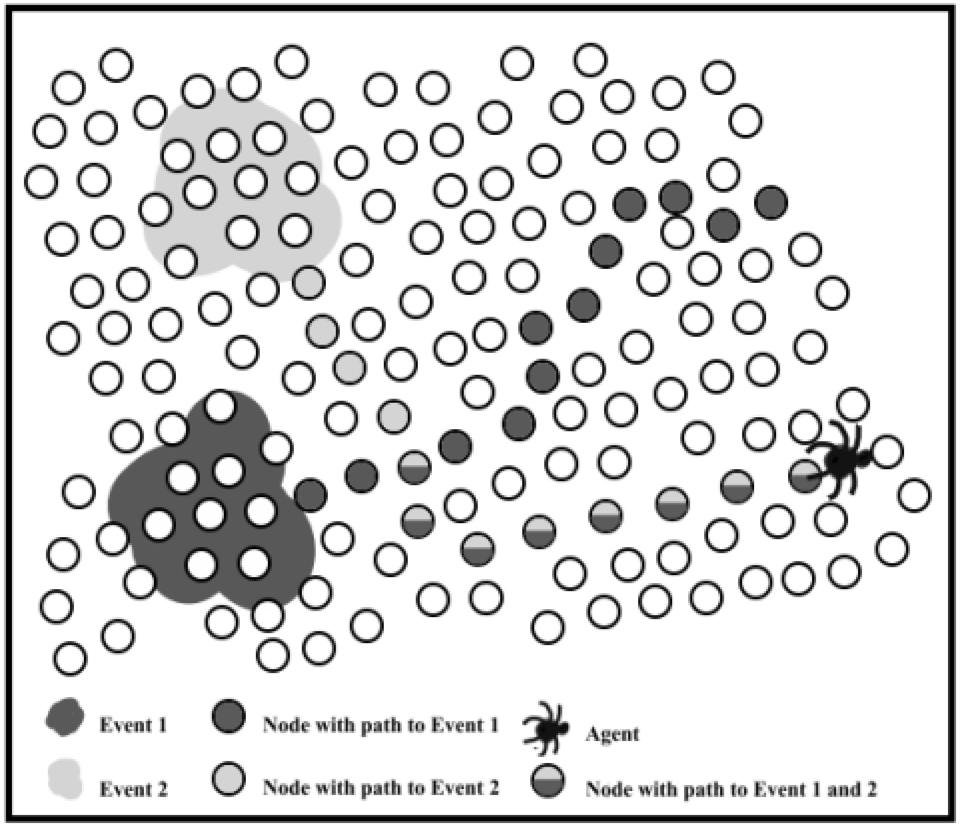
- Representation using a non-parametric approximation will result in a more accurate approximation but more bits will be required to transmit
- Representation using a parametric approximation will result in poor approximation but fewer bits will be required to transmit
- Hybrid Approach
  - Initially, the belief is parameterized by a history of measurements
  - Once the belief looks unimodal, it can be approximated using a parametric approximation like the Gaussian distributions

Other Issues

- The paper initially talks about the mitigation of link/node failures but no experiments are performed to prove it or no mention of it in the algorithms.

Rumor Routing

- Nodes having observed an event send out agents which leave routing info to the event as state in nodes
- Agents attempt to travel in a straight line
- If an agent crosses a path to another event, it begins to build the path to both
- Agent also optimizes paths if they find shorter ones.



Paper: David Braginsky and Deborah Estrin. Slide adapted from Sugata Hazarika, UCLA

Algorithm Basics

- All nodes maintain a neighbor list.
- Nodes also maintain a event table
  - When it observes an event, the event is added with distance 0.
- Agents
  - Packets that carry local event info across the network.
  - Aggregate events as they go.

Agent Path

- Agent tries to travel in a "somewhat" straight path.
  - Maintains a list of recently seen nodes.
  - When it arrives at a node adds the node's neighbors to the list.
  - For the next tries to find a node not in the recently seen list.
  - Avoids loops
  - -important to find a path regardless of "quality"

Following Paths

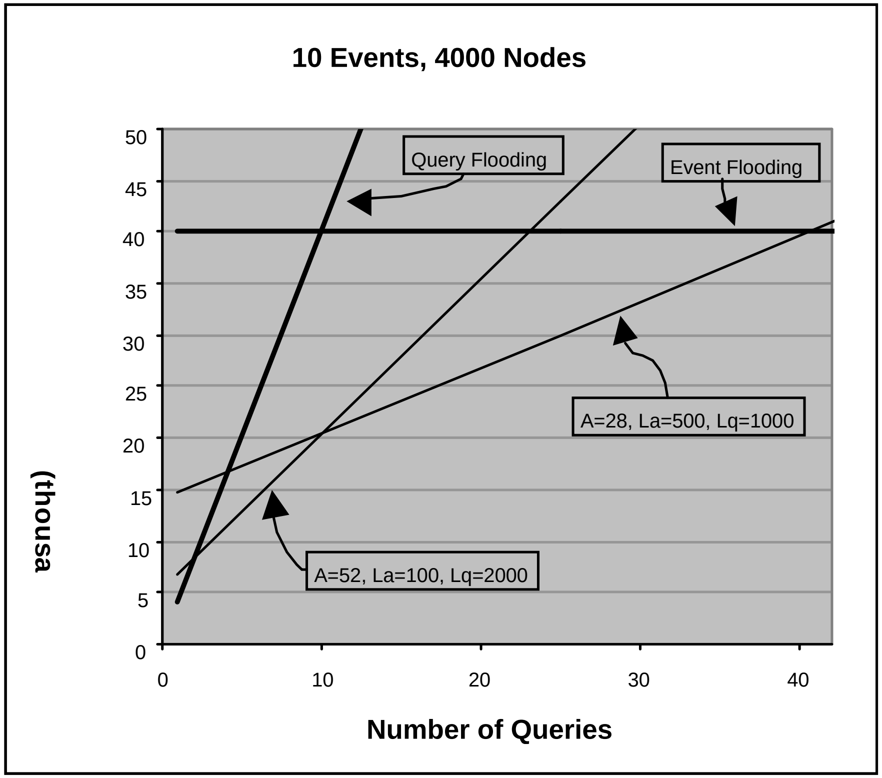
- A query originates from source, and is forwarded along until it reaches it's TTL
- Forwarding Rules:
  - If a node has seen the query before, it is sent to a random neighbor
  - If a node has a route to the event, forward to neighbor along the route
  - Otherwise, forward to random neighbor using straightening algorithm

Some Thoughts

- The effect of event distribution on the results is not clear.
- The straightening algorithm used is essentially only a random walk … can something better be done.
- The tuning of parameters for different network sizes and different node densities is not clear.
- There are no clear guidelines for parameter tuning, only simulation results in a particular environment.

Simulation Results

- Assume that undelivered queries are flooded
- Wide range of parameters allow for energy saving over either of the naïve alternatives
- Optimal parameters depend on network topology, query/event distribution and frequency
- Algorithm was very sensitive to event distribution



Questions?