Finding Convex Hulls of Agents with Sensor Networks

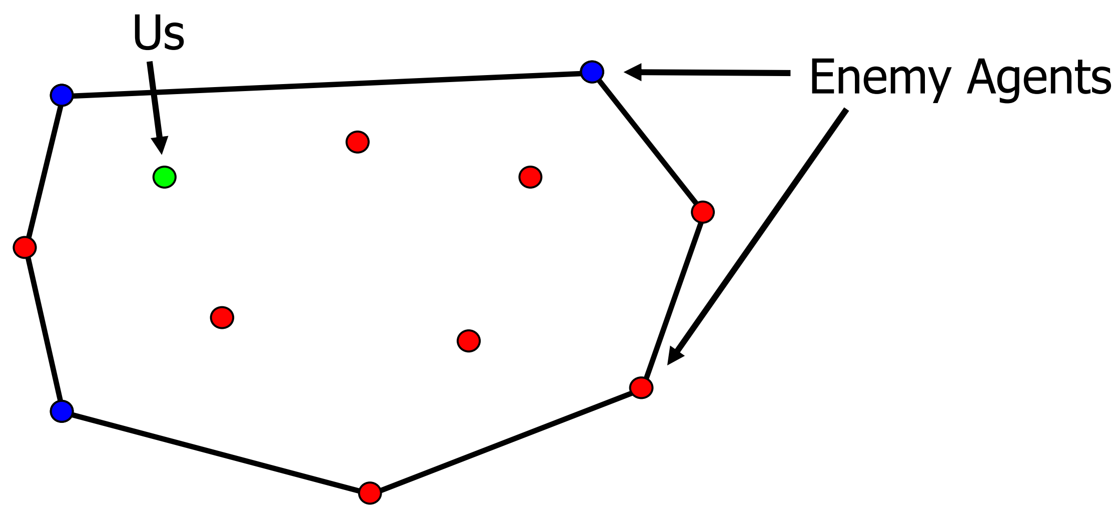
Rahul Biswas

CS428 Project

June 5, 2003

Introduction

- Relational approach to algorithm design
- Determine if agent is surrounded by enemies
- Minimize number of sensor queries required



Overview

- Introduction
- Problem Definition
- Generative Model
- Inference in the Network
- Integrated Sensor Selection Techniques
- Experimental Results
- Conclusion

Overview

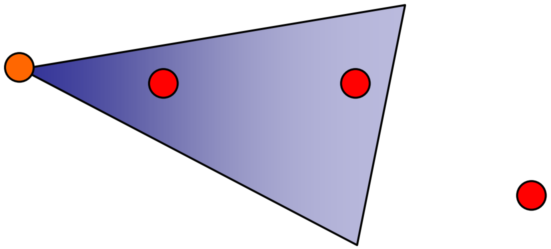
- Introduction
- Problem Definition
- Generative Model
- Inference in the Network
- Integrated Sensor Selection Techniques
- Experimental Results
- Conclusion

Problem Definition

- LF in convex hull with high probability?
- PIR and Amplitude sensors
- Known
  - LF (location of friendly agent)
  - $S_1 \ldots S_M$ (type of each sensor)
  - $LS_1 \ldots LS_M$ (locations of sensors)
  - $C_1 \ldots C_T$ (sensor observations)
- Unknown
  - $LE_1 \ldots LE_N$ (locations of enemy agents)

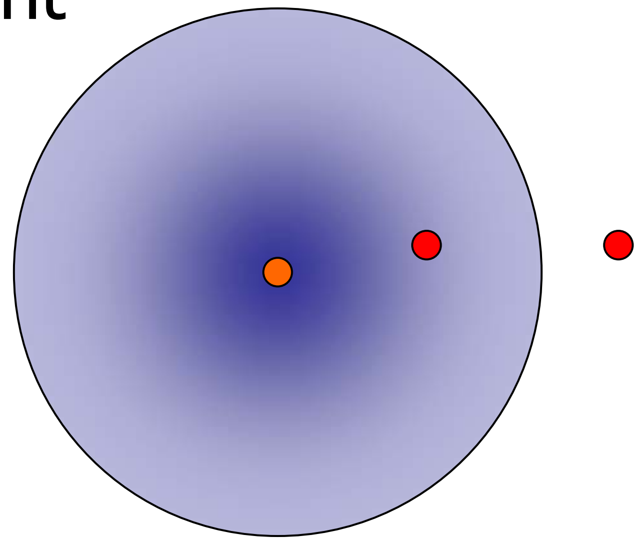
PIR Sensors

- Passive Infrared Intrusion Sensor
- Detects agents in triangle adjoining sensor
- Algorithm choose direction of sensing
- $P(d \mid d') = 1 - 0.2 * p$
- $P(d \mid \neg d') = 0.1$



Amplitude Sensors

- Detects enemy in circle surrounding sensor
- Returns presence and distance to nearest agent
- $P(d \mid d') = 1 - 0.2 * p$
- $P(d \mid \neg d') = 0.1$
- $s \sim N(s',u)$

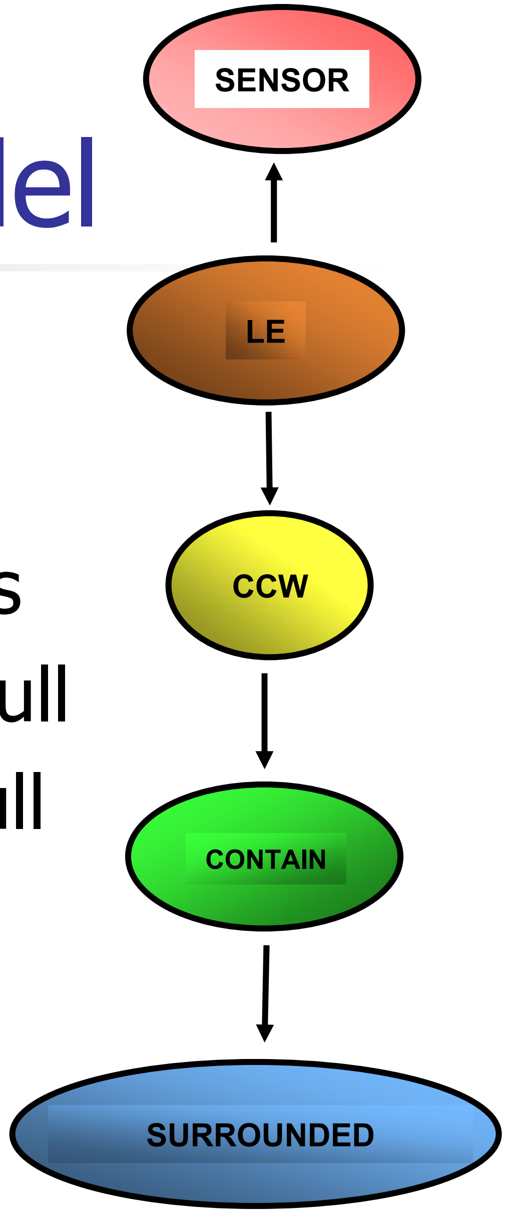


Overview

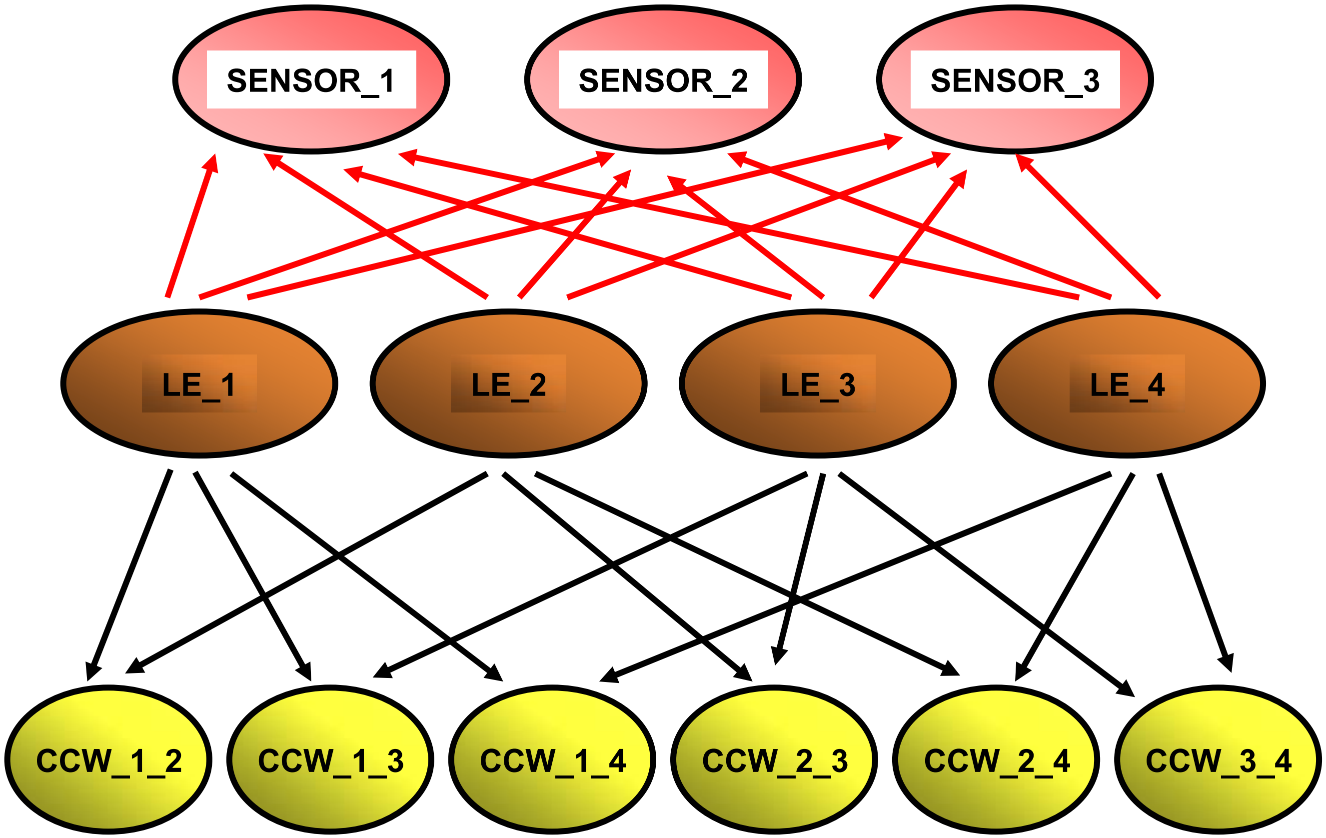
- Introduction
- Problem Definition
- Generative Model
- Inference in the Network
- Integrated Sensor Selection Techniques
- Experimental Results
- Conclusion

Generative Model

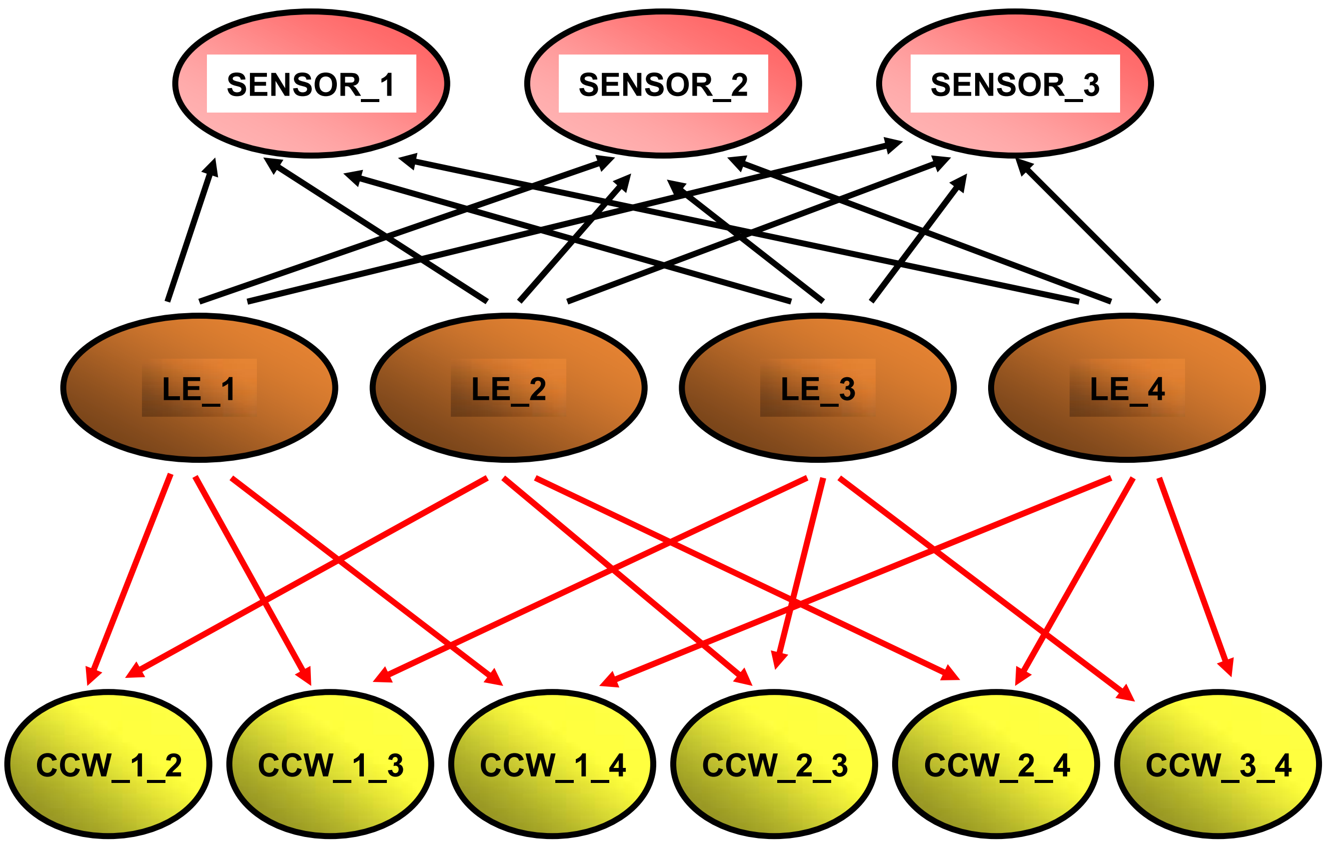
- SENSOR	Results of sensing
- LE	Locations on enemies
- CCW	CCW over LE variables
- CONTAIN	LF in subset convex hull
- SURROUNDED	LF in entire convex hull



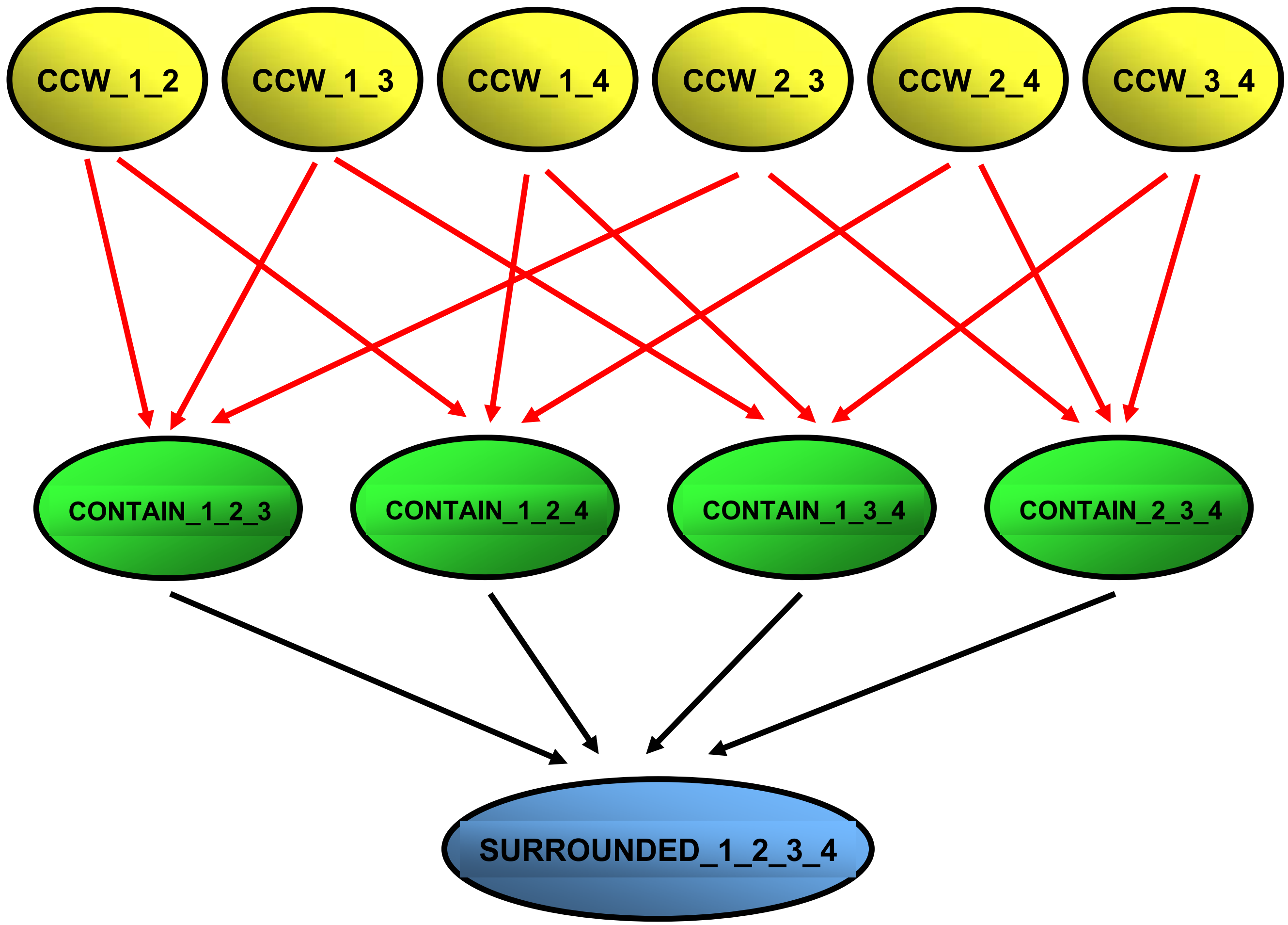
Generative Model



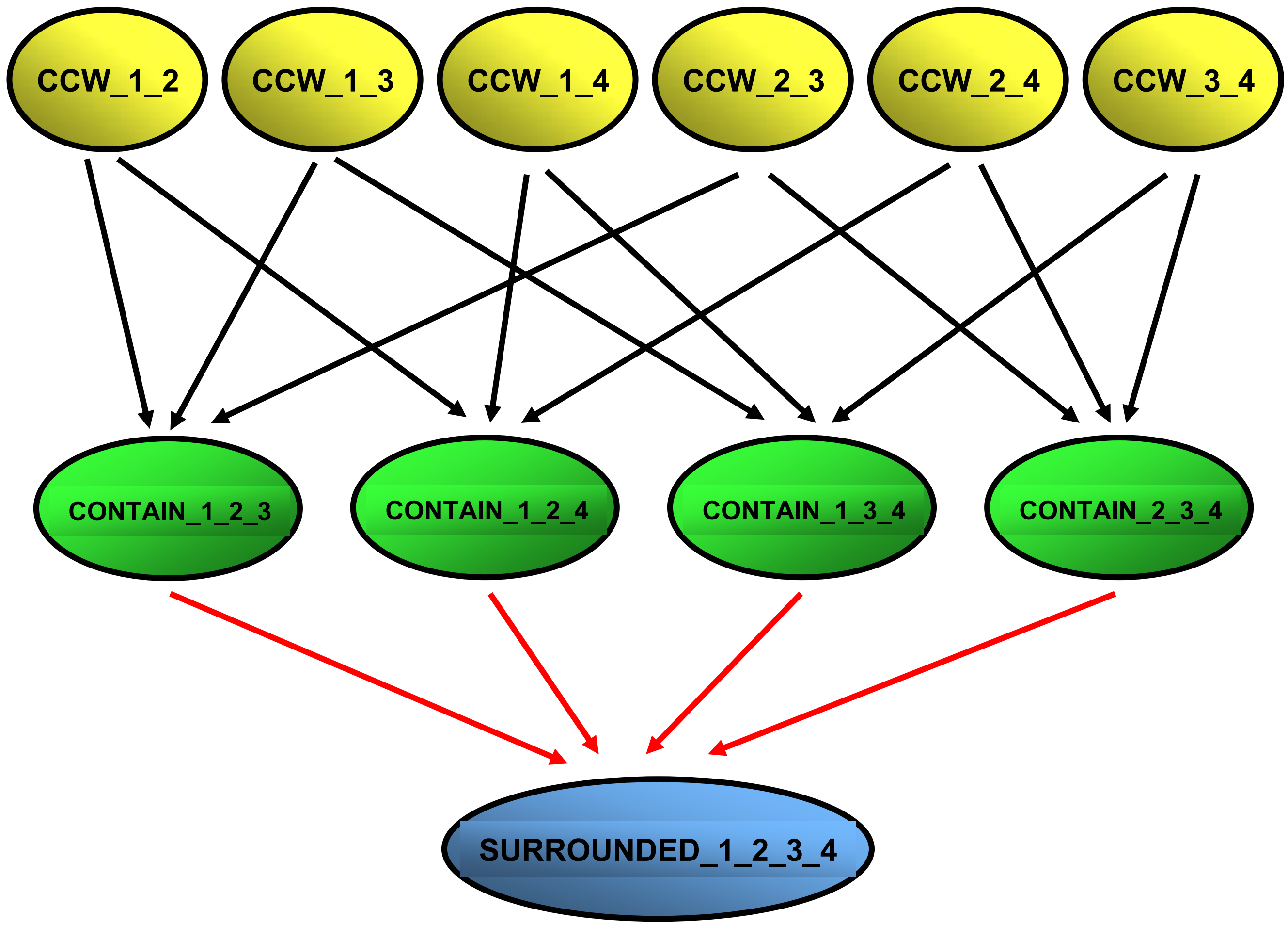
Generative Model



Generative Model



Generative Model



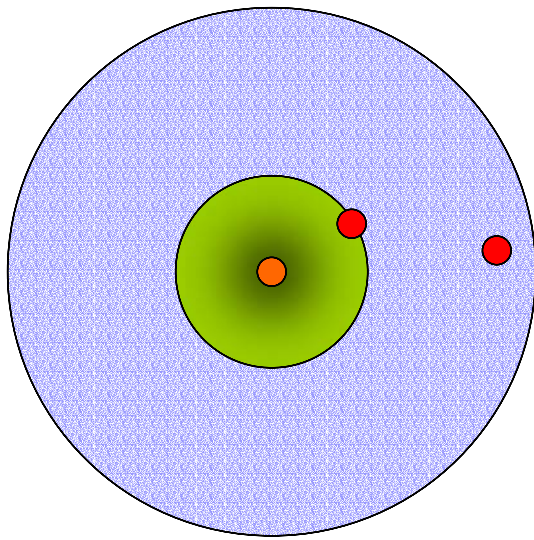
Overview

- Introduction
- Problem Definition
- Generative Model
- Inference in the Network
- Integrated Sensor Selection Techniques
- Experimental Results
- Conclusion

Inference in the Network

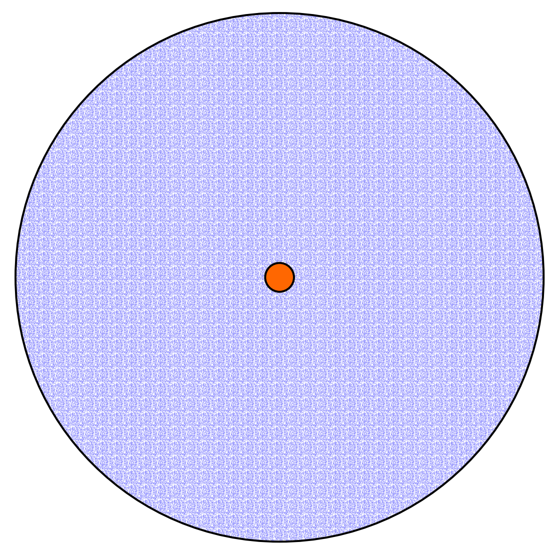
- Difficulties with traditional inference
- Posterior estimation via particle filters
  - Compute LE posteriors given SENSOR variables
  - Compute CONTAIN posteriors given LE
  - Bound SURROUNDED given CONTAIN

LE Posterior from SENSOR



Particles in green gain weight; other particles unaffected

Particles in blue lose weight

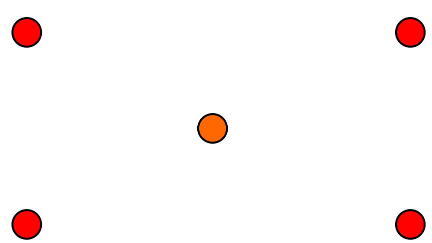


Data Association Problem

- Impossible to differentiate enemy agents
- Find joint posterior over LE given SENSOR
- Run K-means to use in later inference
- Hard assignment of points to clusters
- Model selection for number of enemies

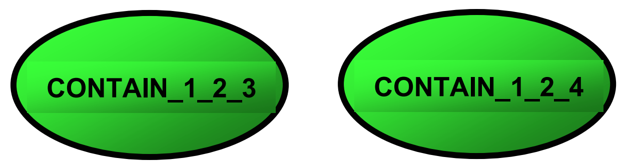
CONTAIN Posteriors Given LE

- Using CCW would yield inaccurate posteriors
- Sample from point clouds of each LE variable
- Count number of times assertion holds
- N-sided polygons often helpful

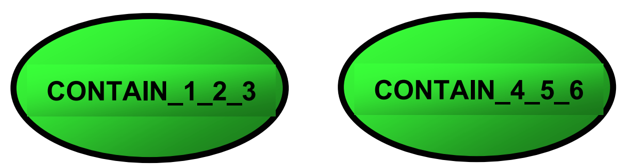


SURROUNDED Given CONTAIN

- CONTAIN variables generally not independent



- Independent if LE variables used disjoint



- Greedily choose pairwise disjoint variables
- Calculate posterior over chosen sets

Overview

- Introduction
- Problem Definition
- Generative Model
- Inference in the Network
- Integrated Sensor Selection Techniques
- Experimental Results
- Conclusion

Intrinsic Difficulty

- Inherently difficult scenarios:
  - When agent is not surrounded
  - When required agents outside sensor range
  - Friendly agent collinear with enemy agents
- Certain configurations confound particle filter
- Simple metric:
  - Inverse of distance to convex hull
  - Infinite if not surrounded

Sensor Utility Computation

- Want to maximize sensor utility
- Exact computation
  - Build search tree of sensor choices
  - Choose first sensor of best branch
  - Exponential in number of sensors
- Greedy computation
  - Depth limited search
  - Linear in inference time

Algorithm RANDOM

- Chooses sensors randomly
- Repetition of sensors never allowed
- Better than sensing at all sensors

Algorithm TRIANGLE

- Uses RANDOM until three affirmative readings
- Build triangles:
  - Two pre-determined enemy agent locations
  - One sensor location as potential agent location
  - Triangle contains friendly agent
- Choose unused sensor with most triangles

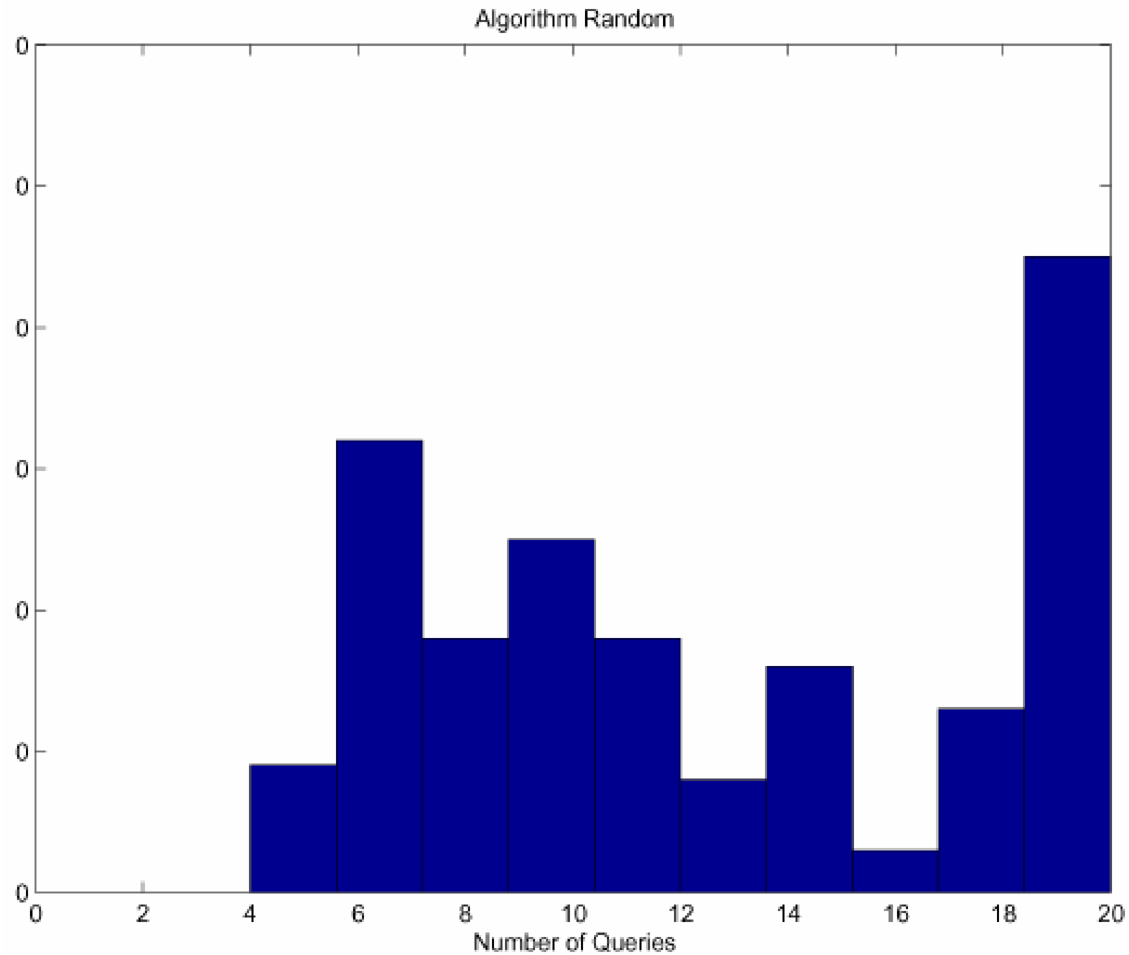
Algorithm ENTROPY

- Uses RANDOM until three affirmative readings
- Calculate entropy for each expected reading
  - Do calculation for each unused sensor
  - Only affirmative response calculated
- Choose unused sensor with lowest entropy
- Running time linear with number of sensors

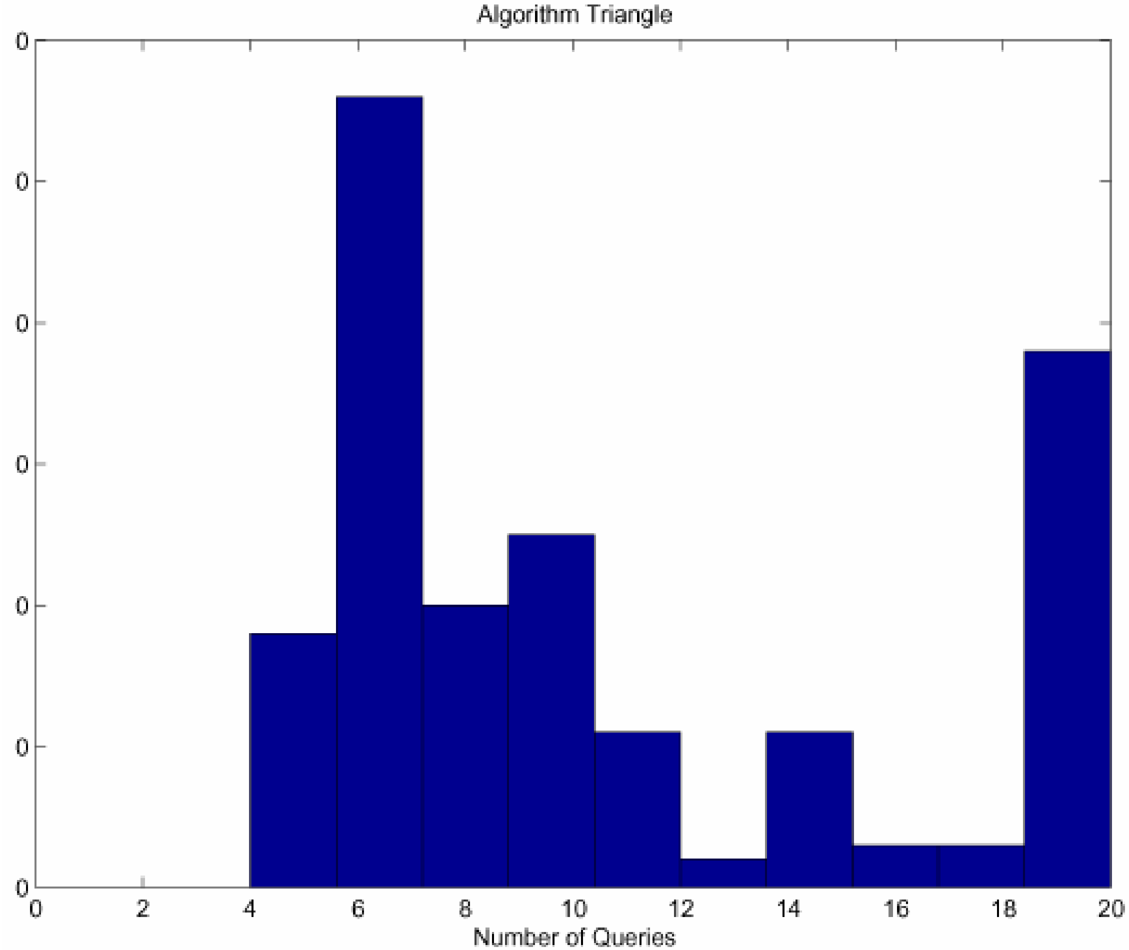
Overview

- Introduction
- Problem Definition
- Generative Model
- Inference in the Network
- Integrated Sensor Selection Techniques
- Experimental Results
- Conclusion

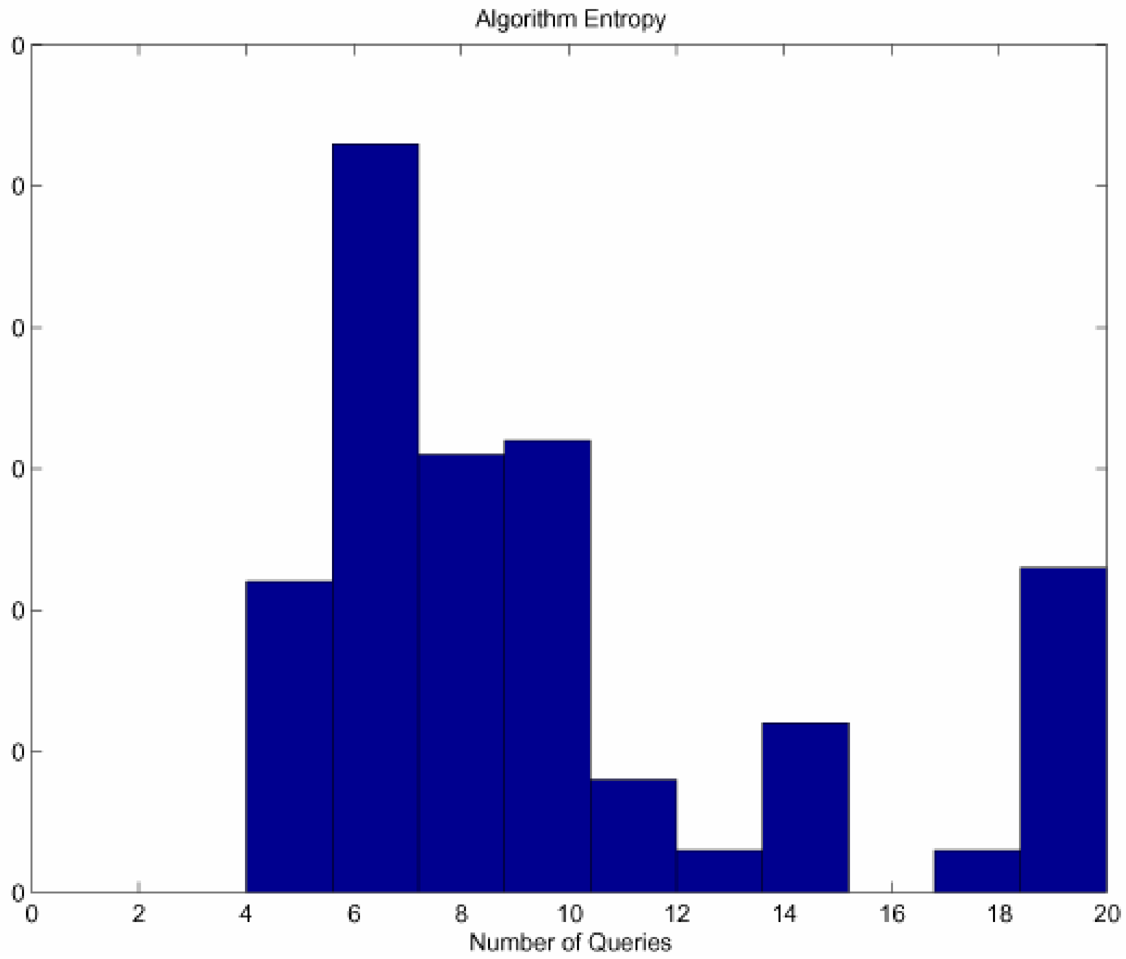
Algorithm RANDOM



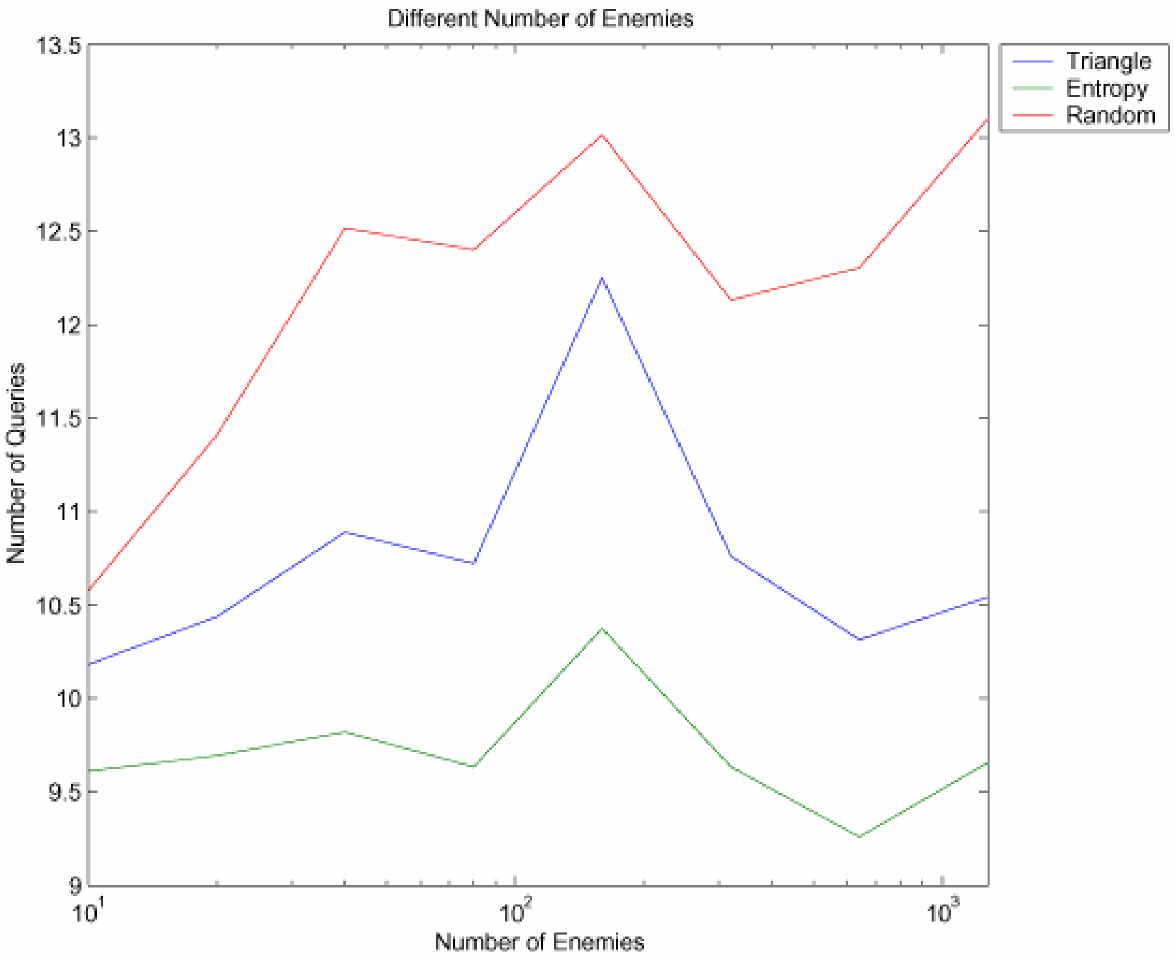
Algorithm TRIANGLE



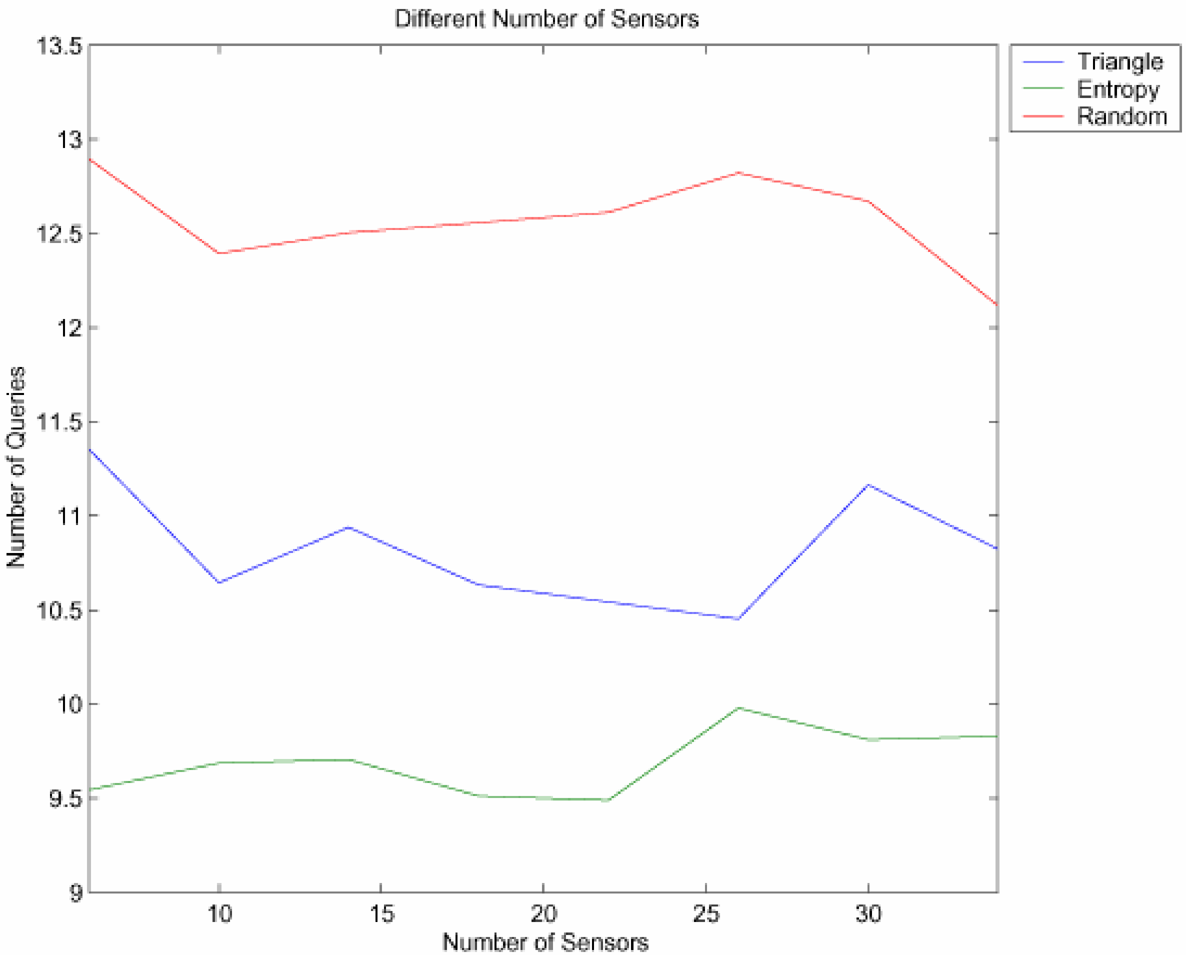
Algorithm ENTROPY



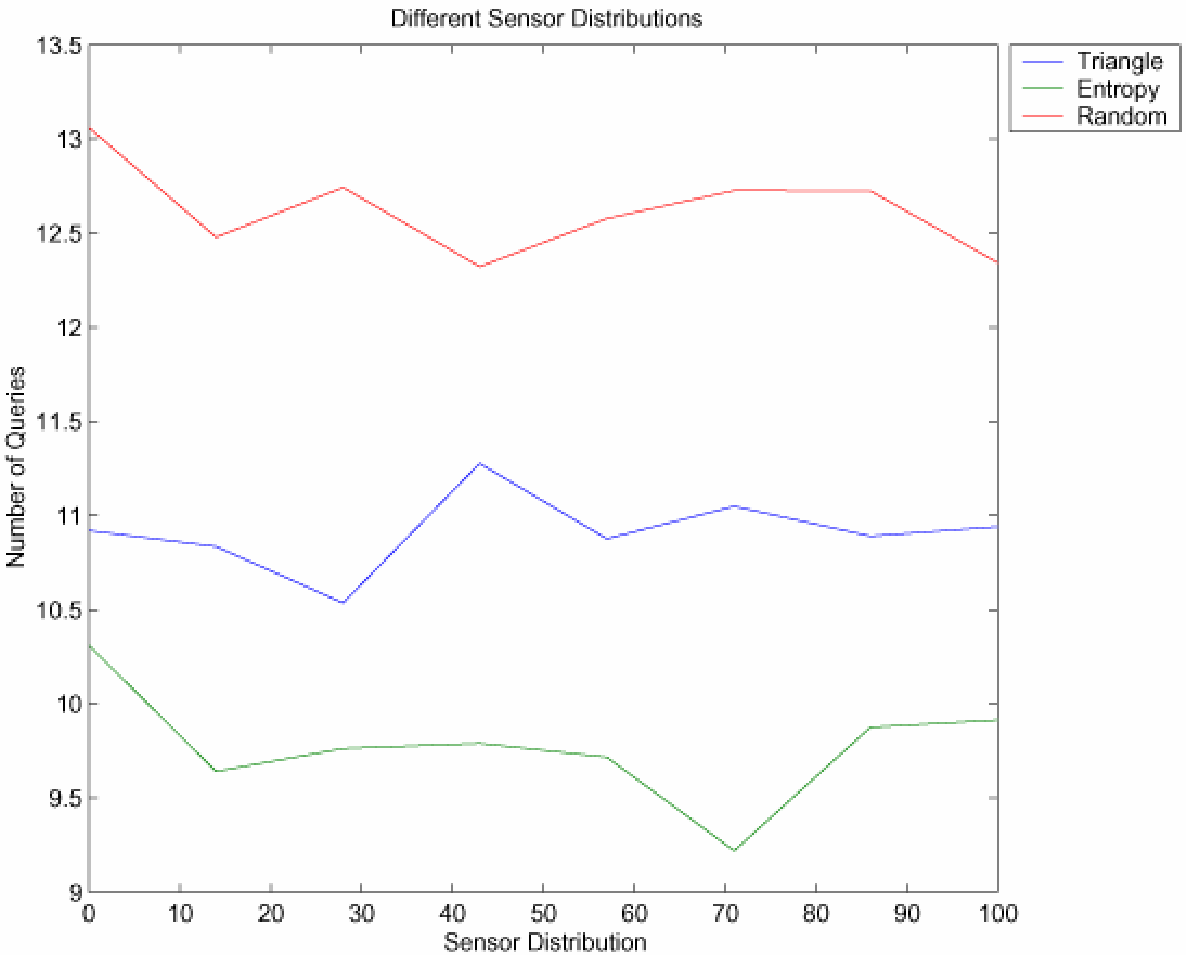
Varying Number of Enemies



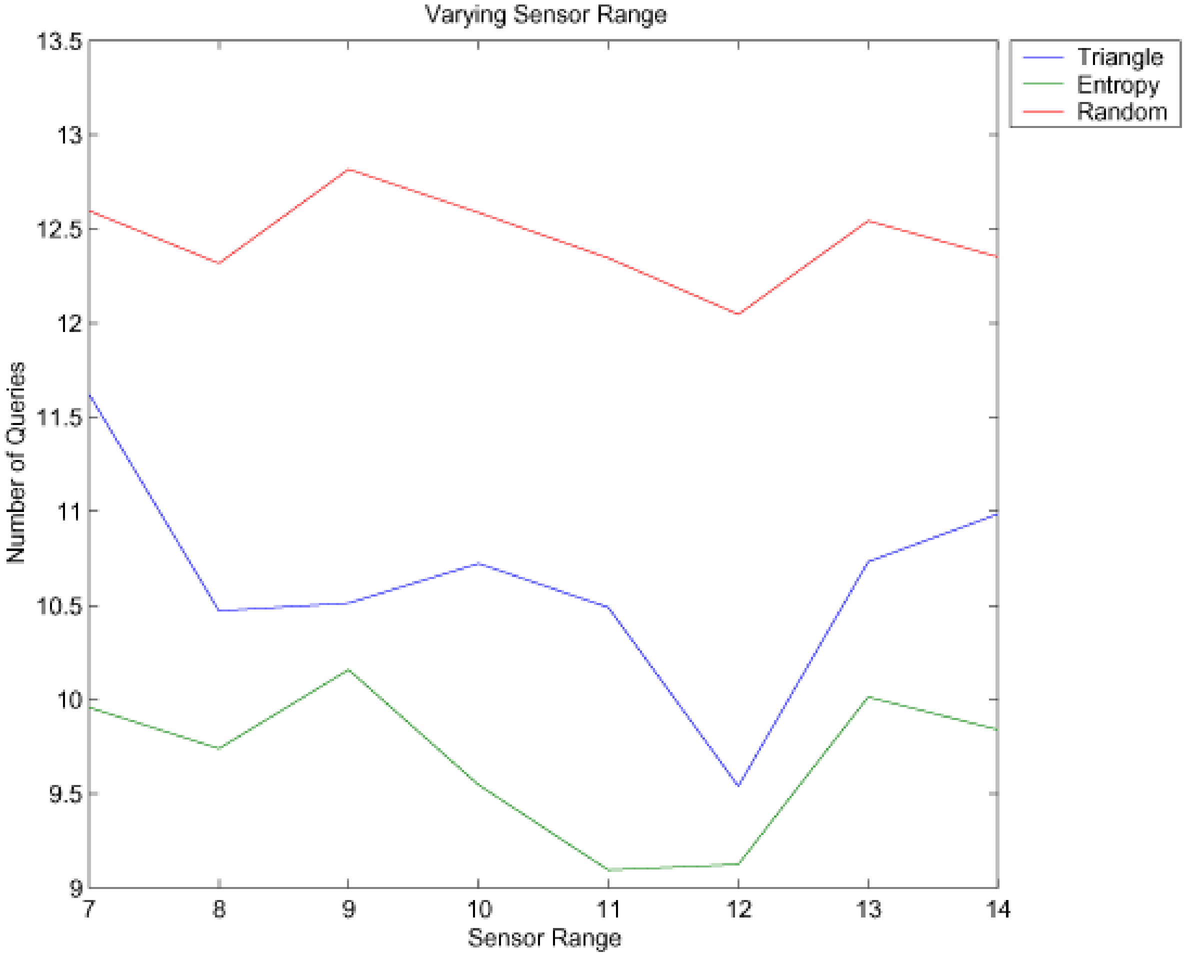
Varying Number of Sensors



Varying Sensor Composition



Varying Field of View



Overview

- Introduction
- Problem Definition
- Generative Model
- Inference in the Network
- Integrated Sensor Selection Techniques
- Experimental Results
- Conclusion

Conclusion

- Probabilistic model for Surrounded problem
- Inference technique for calculating posteriors
- Algorithms to choose subsequent sensors