Movement-Assisted Sensor Deployment

INFOCOM 2004

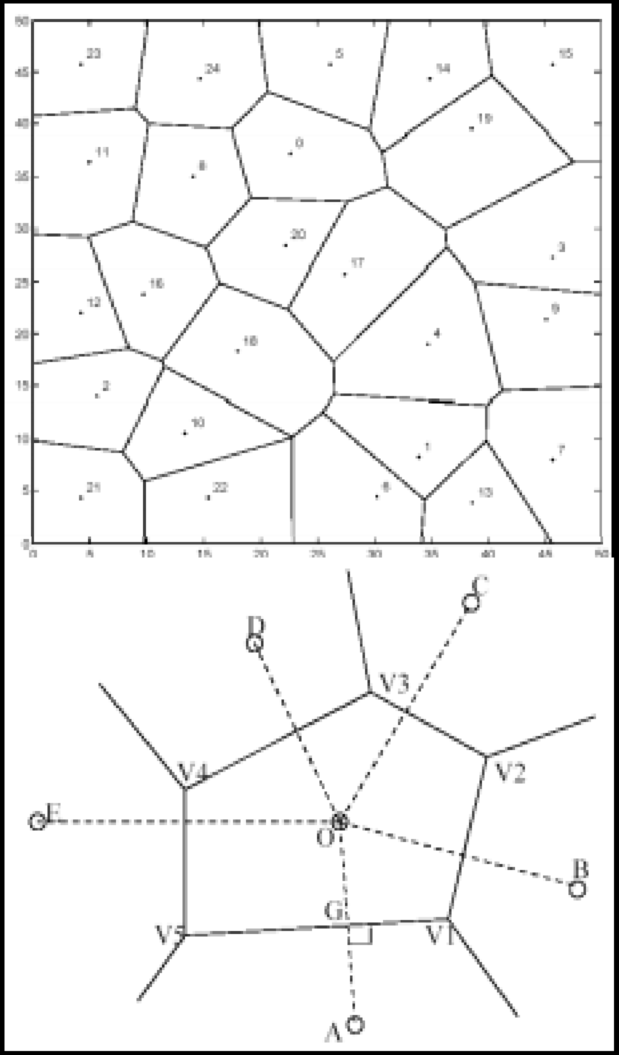
The problem of sensor deployment

- Given the target area, how to maximize the sensor coverage with less time, movement distance and message complexity
- The importance of the problem
- Distributed instead of centralized

Voronoi diagram

- Definition:

  Every point in a given polygon is closer to the node in this polygon than to any other node.



Overview of the proposed algorithm

- Sensors broadcast their locations and construct local Voronoi polygons
- Find the coverage holes by examine Voronoi polygons
- If holes exist, reduce coverage hole by moving
  - VEC, VOR, Minimax

Reduce coverage holes

- VEC：VECtor-based
  - Push sensors away from a densely covered area
- VOR：VORonoi-based
  - Pull sensors to the sparsely covered area
- Minimax
  - Move sensors to their local center area

VEC algorithm

- Motivated by the attributes of electro-magnetic particles
- $d(s_i, s_j)$: distance between sensor i and sensor j
- d_{ave} : average distance between two sensors
- one node fixed, the other will be pushed $d_{ave} - d(s_i, s_j)$

VEC algorithm

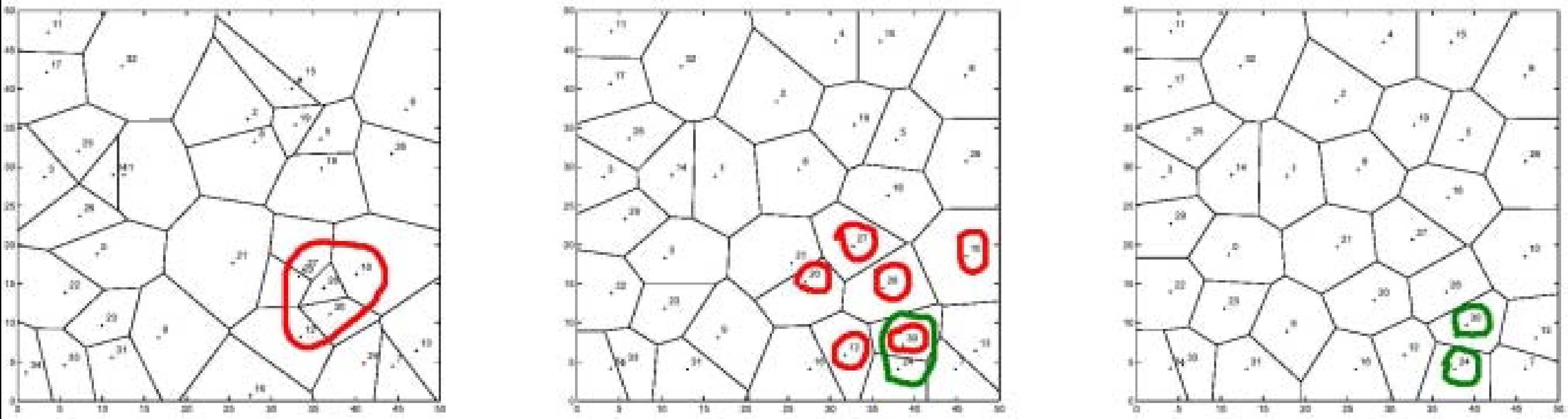
- Broadcast hello to neighbors
- Receive hello messages from neighbors and update Voronoi polygon
- Do vector summation
- Do movement adjustment

Movement adjustment

- To reduce the error of virtual-force
- It checks whether the local coverage will be increased by moving to the target location
- If local coverage increased at target location, the sensor will move, otherwise, it will stay

Execution of VEC

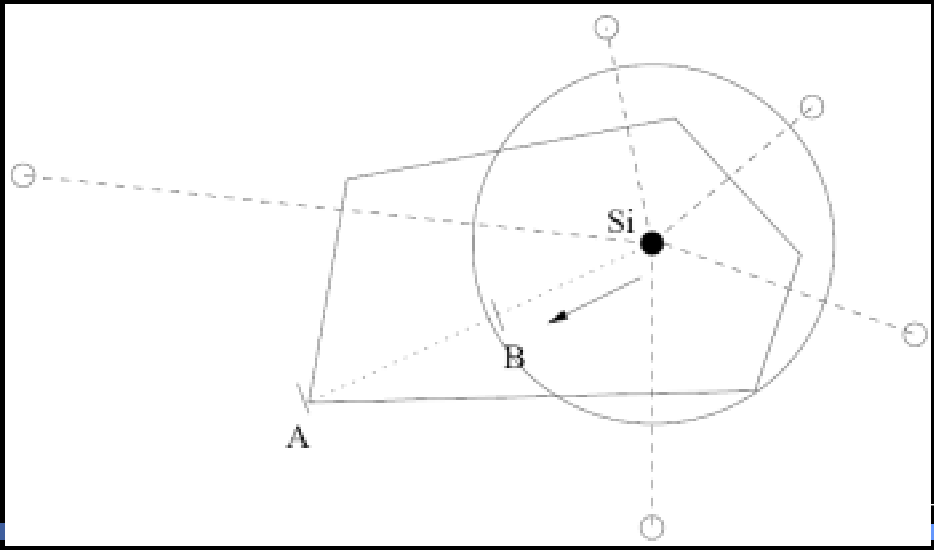
75.7 % → 92.2 % → 94.7 %



(a) Round 0 (b) Round 1 (c) Round 2

VOR algorithm

- Pull sensors to get local maximum coverage
- If a sensor detects coverage holes, it will move toward its farthest Voronoi vertex

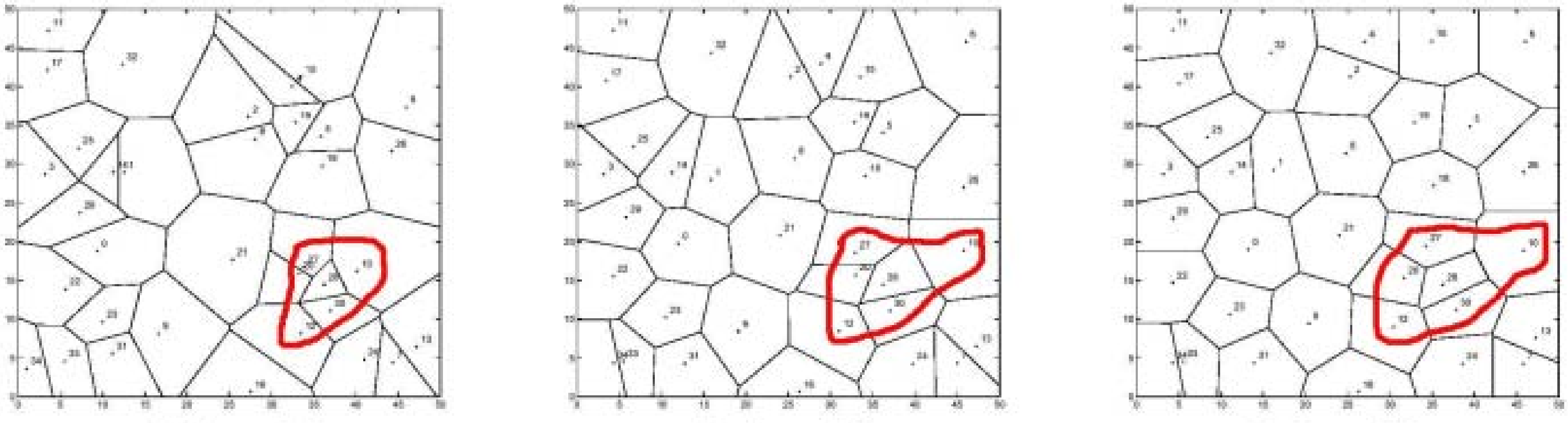


VOR algorithm

- VOR is a greedy algorithm
- Try to fix the largest hole
- Should have oscillation control
- Movement adjustment is also applied

Execution of VOR

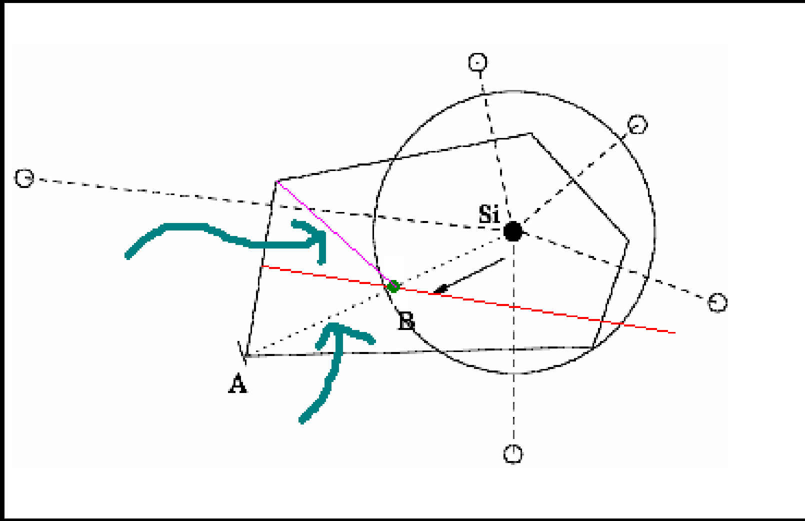
75.7 % → 89.2 % → 95.6 %



(a) Round 0 (b) Round 1 (c) Round 2

Minimax algorithm

- Fix holes by moving closer to the farthest Voronoi vertex
- Choose the target location whose distance to the farthest Voronoi vertex is minimized

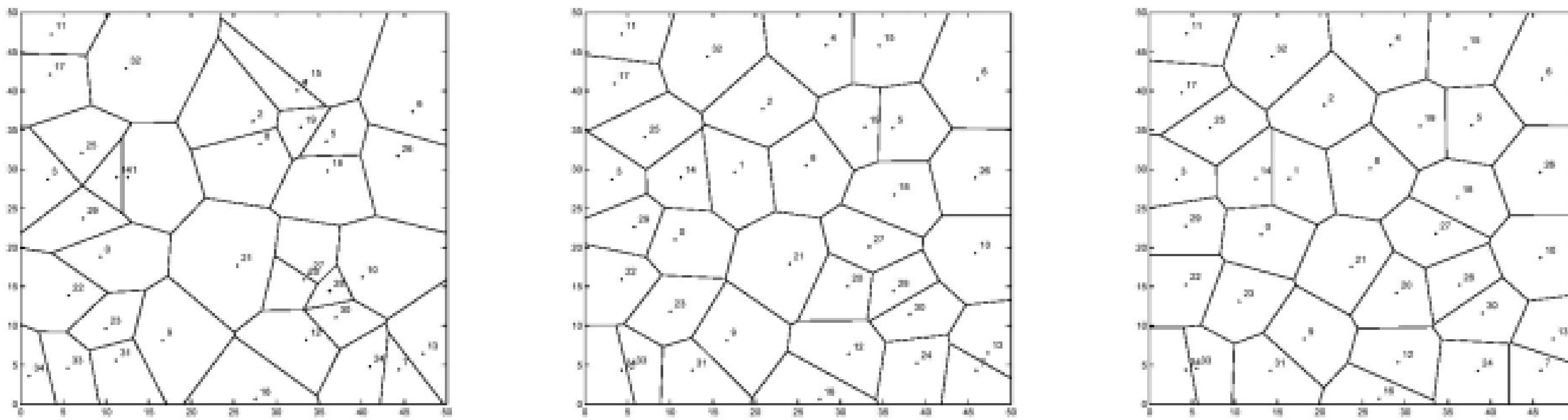


Minimax algorithm

- Based on the belief that a sensor should not be too far away from any of its Voronoi vertices when sensors are evenly distributed
- Result in a more regular shaped Voronoi polygon

Execution of Minimax

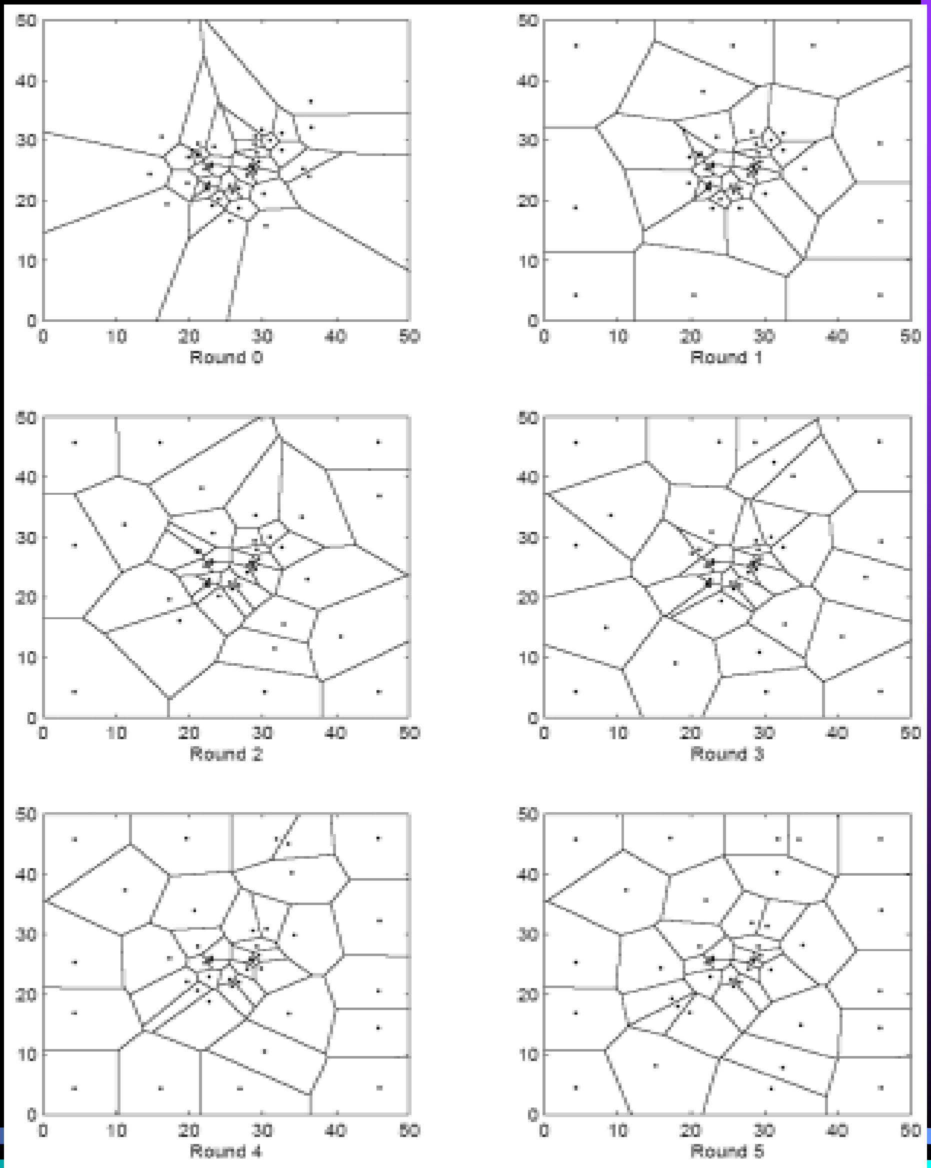
75.5 % → 92.7 % → 96.5 %



(a) Round 0 (b) Round 2 (c) Round 3

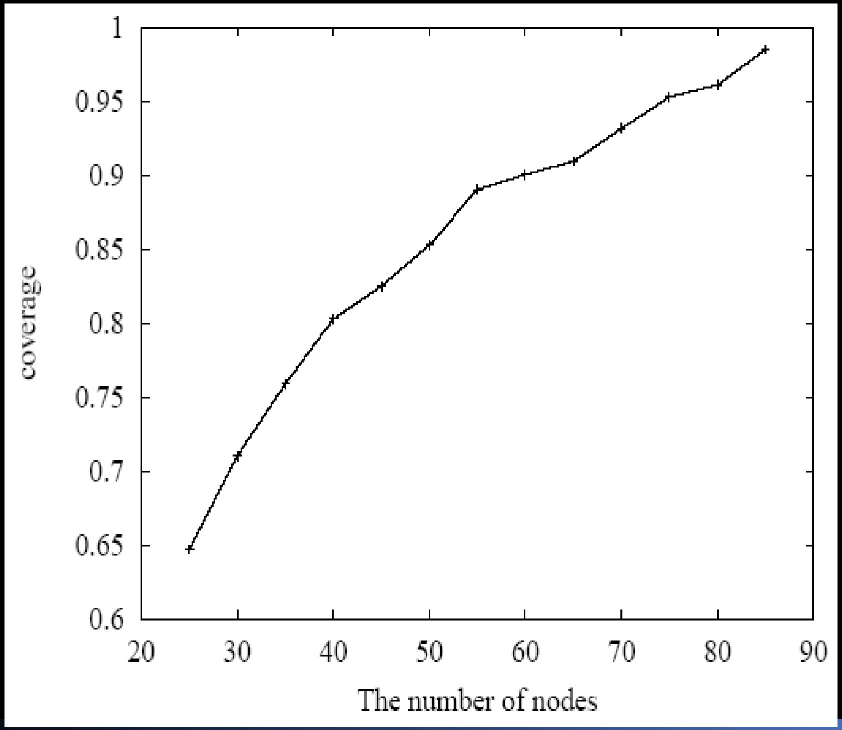
Optimizations

- Initial deployment may form clusters
- Explode!



Simulation results

Randomly deployed : about 85 sensors are required to reach 98 % coverage



Simulation results

The proposed algorithms : only 40 sensors are needed to reach the same coverage