

大阪大学大学院情報科学研究科

Osaka University

Graduate School of Information Science & Technology

Advanced Network Architecture Laboratory

<http://www.anarg.jp/>



<http://www.yuragi.osaka-u.ac.jp>

A Heuristic Approach for K-Coverage Extension with Energy-Efficient Sleep Scheduling in Sensor Networks

Kenji Leibnitz, Indika Suranjith Abeyweera,
Naoki Wakamiya, Masayuki Murata

Outline

- ▶ Introduction
- ▶ Problem formulation
- ▶ Deployment of sensor nodes with “Gas Bubble” algorithm
 - Algorithm formulation
 - Performance evaluation
- ▶ Integration with dynamic sleep scheduling
- ▶ Conclusion and Outlook

Introduction

- ▶ Future ambient information infrastructures require huge number of sensors with redundant coverage
- ▶ “Coverage” encompasses two parts:
 - Sensing coverage (type of sensor)
 - RF communication range (connectivity)
- ▶ Deployment of sensor nodes is an NP-complete problem to obtain K -coverage, especially when a partially existing network is extended

K-Coverage Problem



► Example scenario:

- Consider surveillance of a museum with sensors
- Some objects require more reliable monitoring with redundant coverage
- Failure of individual sensor can be compensated

K-Coverage Problem



► Example scenario:

- Consider surveillance of a museum with sensors
- Some objects require more reliable monitoring with redundant coverage
- Failure of individual sensor can be compensated

Assumptions and Definitions

- ▶ Rectangular monitoring window W
- ▶ Set of sensor nodes \mathcal{S} with $|\mathcal{S}| = M$
- ▶ Sensing coverage radius r
- ▶ Circular sensing coverage areas, coverage degree:

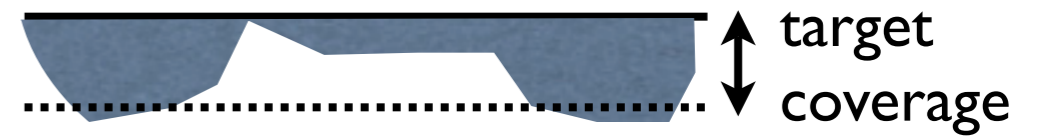
$$C(x) = |\{s \in \mathcal{S} : \|s - x\| < r\}|$$

- ▶ **Definition:**

We say that a region W has obtained K -coverage, iff $\forall x \in W \ C(x) \geq K$ where $C : \mathbb{R}^2 \rightarrow \mathbb{N}$ is the number of sensor nodes covering a point $x \in W$

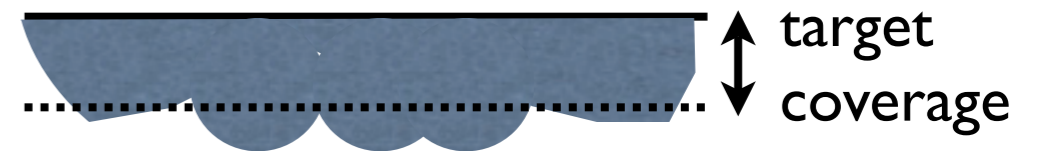
Gas Bubble Algorithm

- ▶ Mimics the behavior of gas bubbles slowly rising in a liquid before changing its state to solid
- ▶ Heuristic method based on self-organizing feature maps, especially *Growing Neural Gas*
- ▶ Reaction of neurons to stimuli in the given area
- ▶ The distance of each neuron is indicator for suitability of “cooling down” (as in *Simulated Annealing*)



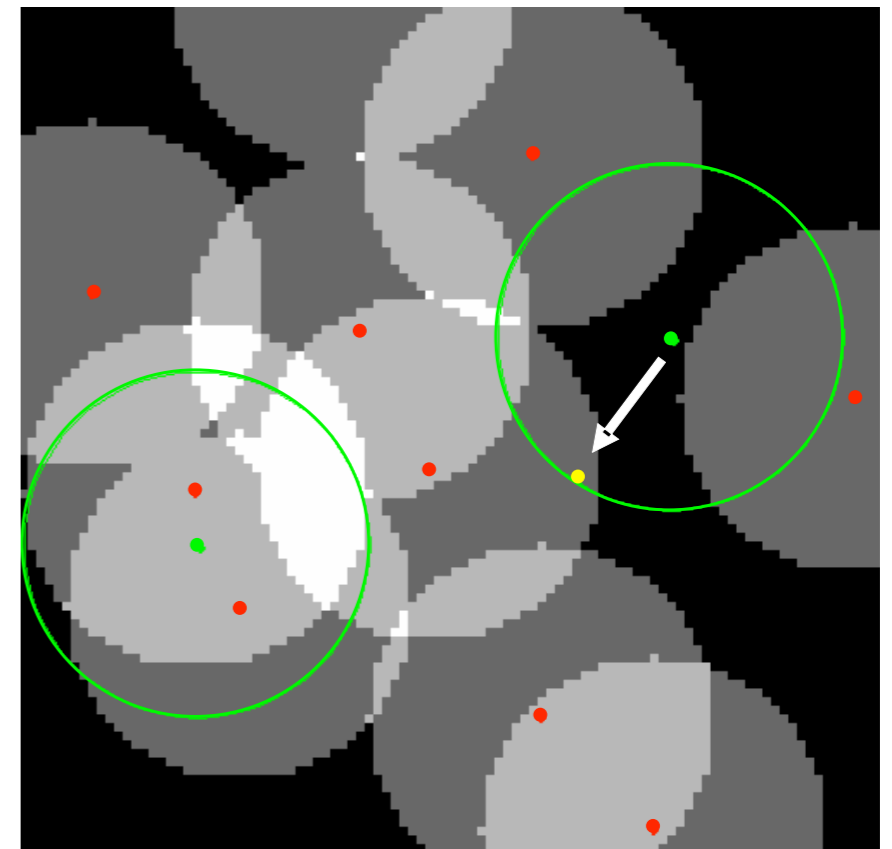
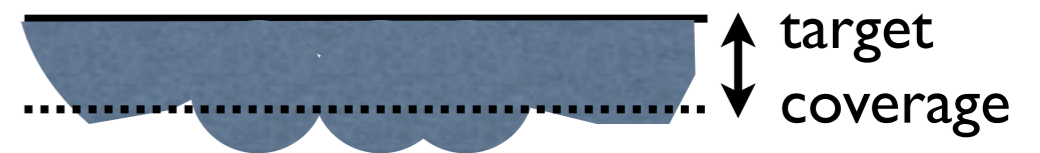
Gas Bubble Algorithm

- ▶ Mimics the behavior of gas bubbles slowly rising in a liquid before changing its state to solid
- ▶ Heuristic method based on self-organizing feature maps, especially *Growing Neural Gas*
- ▶ Reaction of neurons to stimuli in the given area
- ▶ The distance of each neuron is indicator for suitability of “cooling down” (as in *Simulated Annealing*)



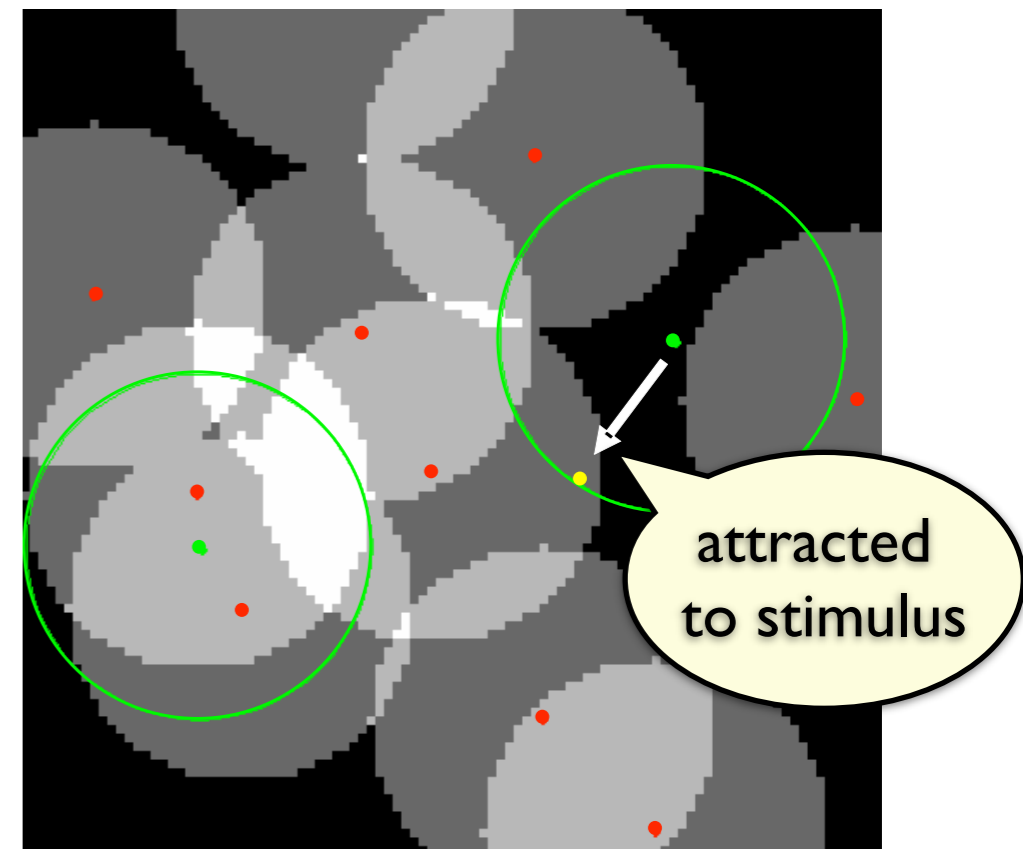
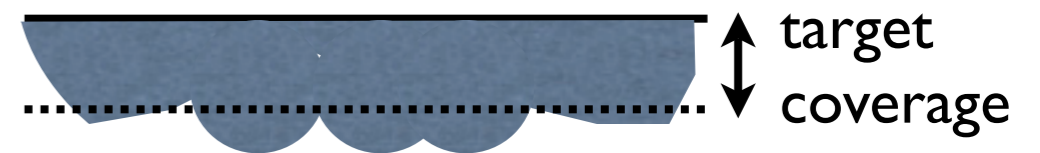
Gas Bubble Algorithm

- ▶ Mimics the behavior of gas bubbles slowly rising in a liquid before changing its state to solid
- ▶ Heuristic method based on self-organizing feature maps, especially *Growing Neural Gas*
- ▶ Reaction of neurons to stimuli in the given area
- ▶ The distance of each neuron is indicator for suitability of “cooling down” (as in *Simulated Annealing*)



Gas Bubble Algorithm

- ▶ Mimics the behavior of gas bubbles slowly rising in a liquid before changing its state to solid
- ▶ Heuristic method based on self-organizing feature maps, especially *Growing Neural Gas*
- ▶ Reaction of neurons to stimuli in the given area
- ▶ The distance of each neuron is indicator for suitability of “cooling down” (as in *Simulated Annealing*)



Formulation of Bubble Algorithm

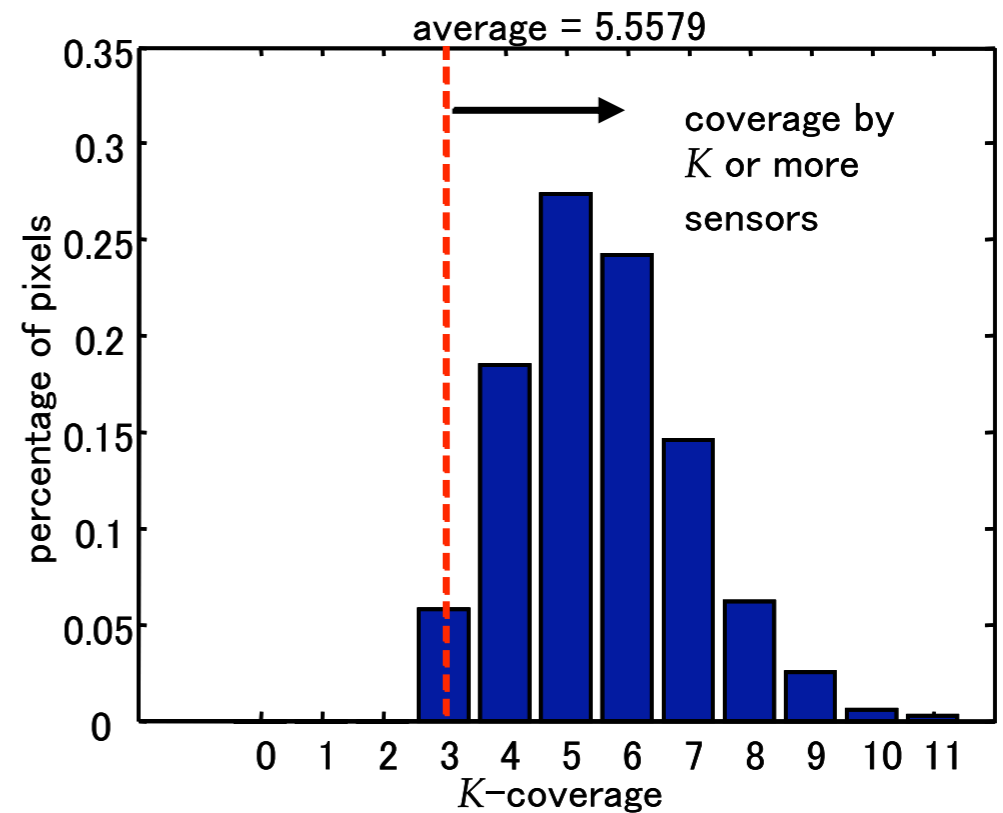
1. Initialize $step = 0$
2. While there are still locations with coverage less than K
 - 2.1. $step = step + 1$
 - 2.2. Start with two neurons n_1 and n_2 at random positions in the window
 - 2.3. Generate input stimulus ξ randomly among all uncovered points in W proportional to $K - C(\xi)$
 - 2.4. Find the nearest neuron b to ξ and update its error value and reset its age.
$$error(b) = error(b) + \|b - \xi\|$$
$$age(b) = 0$$
 - 2.5. Move b by ϵ towards ξ : $b = b + \epsilon(\xi - b)$
 - 2.6. If $t_{add} \bmod step == 0$, add a neuron at the location of the neuron with the highest error and split this total error among the old and new neuron.
 - 2.7. If $s_{add} \bmod step == 0$, the neuron with lowest error becomes a sensor node if there is a coverage gain.
 - 2.8. Remove all neurons n_i with $age(n_i) > t_{age}$
 - 2.9. Decrease the error of each neuron n_j by a decay factor δ and increment its age

$$error(n_j) = \delta error(n_j)$$

$$age(n_j) = age(n_j) + 1$$

Parameter Settings

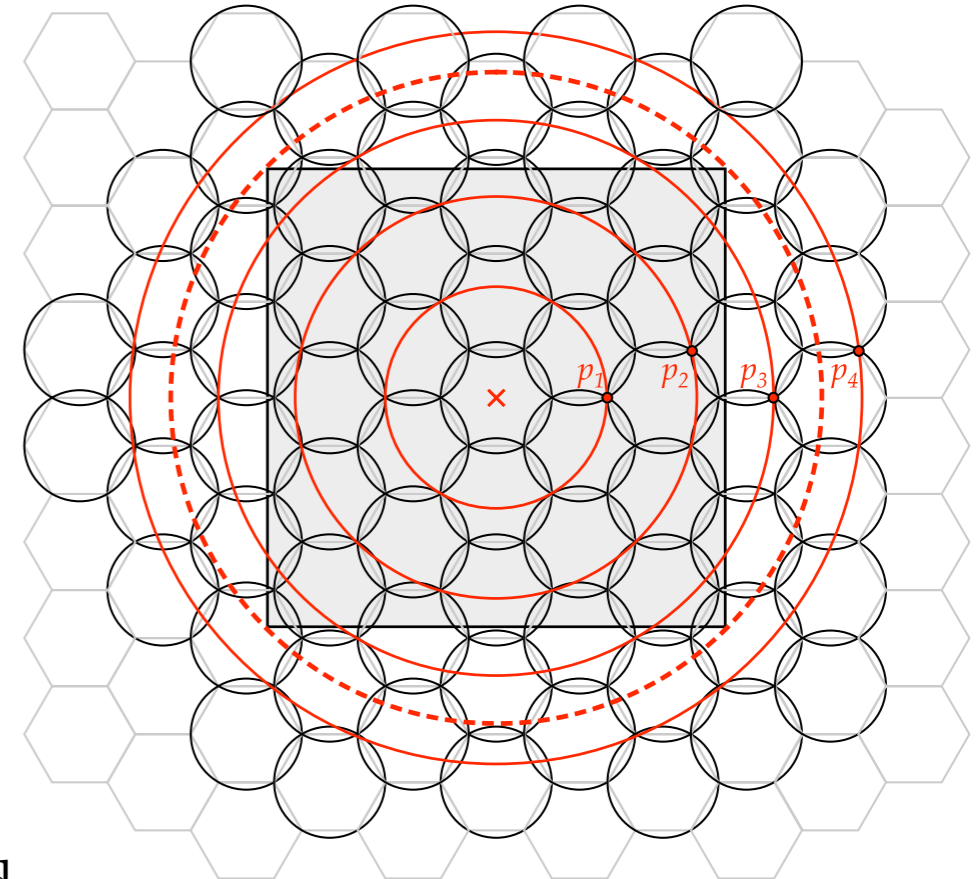
- ▶ Starting with 2 neurons, their error is indicator for adding new neurons
- ▶ Every t_{add} time steps a new neuron is added by splitting the node with the highest error value
- ▶ Every s_{add} time steps a neuron becomes fixed
- ▶ Circular sensor coverage areas cause much overlap, especially for large K



$$\begin{array}{ll} t_{add} = 200 & s_{add} = 500 \\ \epsilon = 0.2 & t_{age} = 200 \\ \delta = 0.95 & w = 100 \\ r = 13 & K = 3 \end{array}$$

Approximation of Optimal Coverage

- ▶ Best approximation of circular coverage areas by hexagon with superposition of K layers
- ▶ Beginning from center ('x'), calculate p_i and their radius y_i
- ▶ Number of nodes L_i for tier i



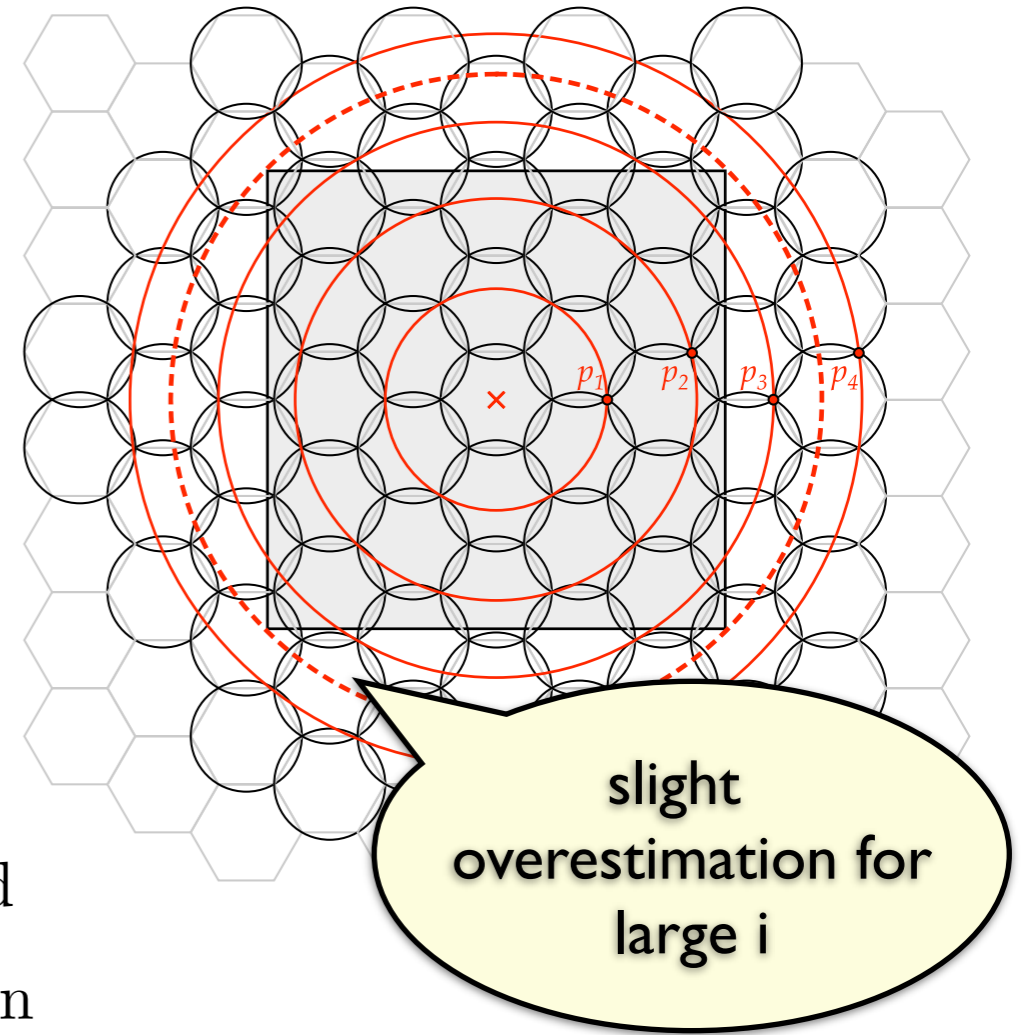
$$y_i = \begin{cases} r (2 + 3 (i - 1)) & \text{if } i \text{ is odd} \\ \frac{r}{2} \sqrt{(7 + 6 (i - 2))^2 + 3} & \text{if } i \text{ is even} \end{cases}$$

$$L_i = 1 + 6 \sum_{k=1}^i k = 1 + 3i(i + 1)$$

i	1	2	3	4	5	6	7
y_i	30.0	54.1	120.0	143.1	210.0	232.9	300.0

Approximation of Optimal Coverage

- ▶ Best approximation of circular coverage areas by hexagon with superposition of K layers
- ▶ Beginning from center ('x'), calculate p_i and their radius y_i
- ▶ Number of nodes L_i for tier i

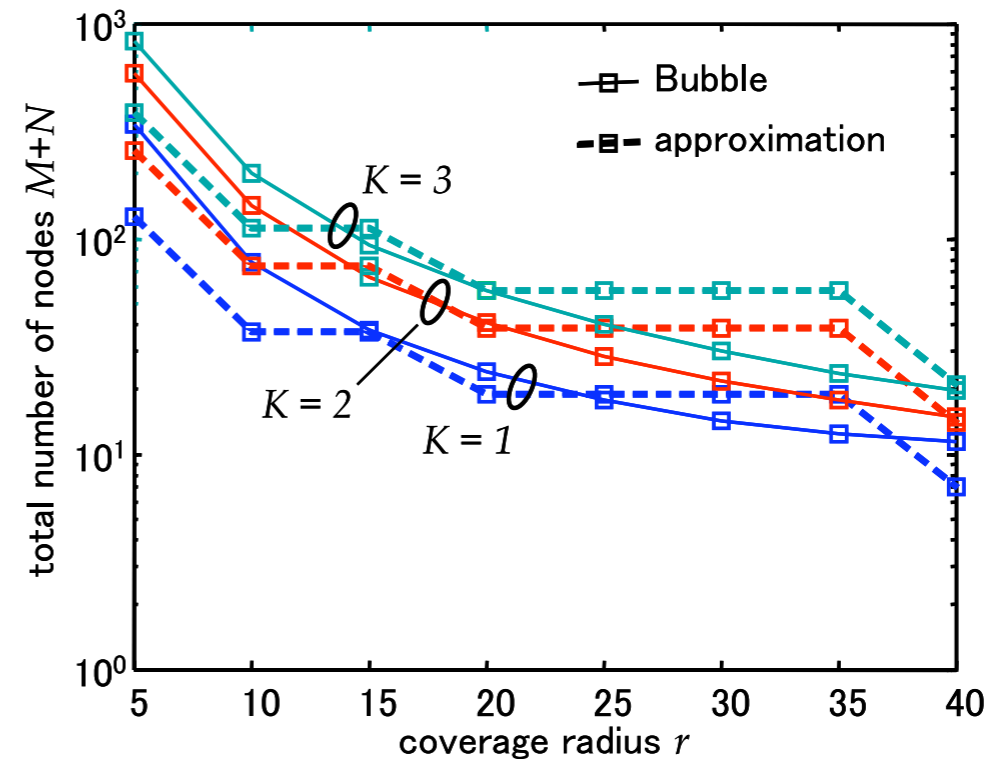
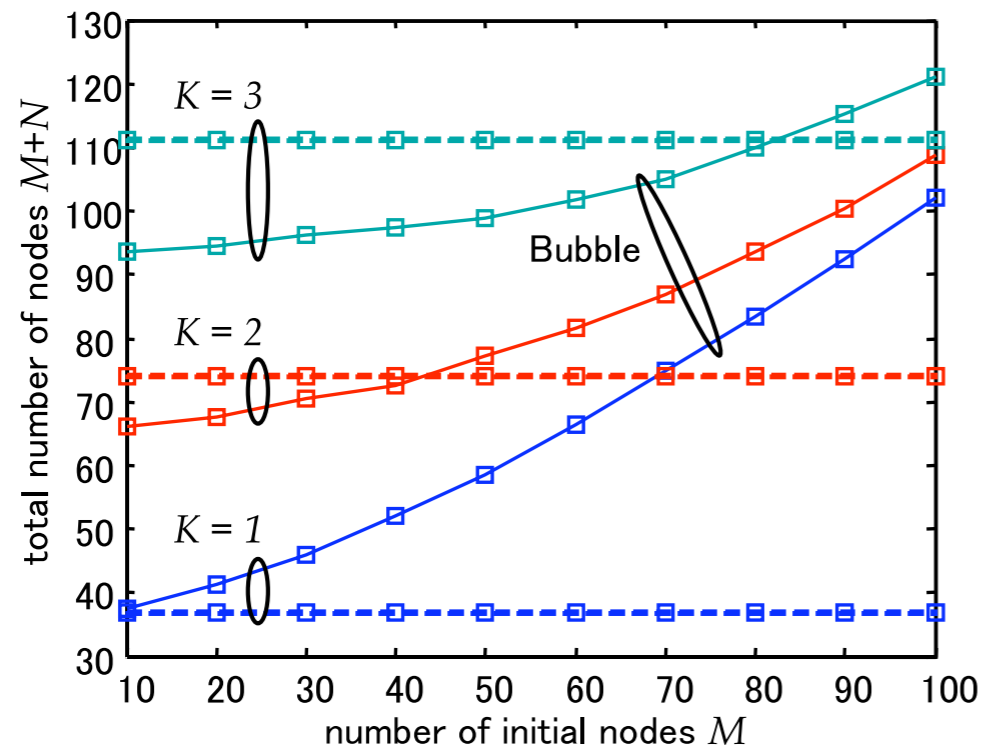


$$y_i = \begin{cases} r (2 + 3 (i - 1)) & \text{if } i \text{ is odd} \\ \frac{r}{2} \sqrt{(7 + 6 (i - 2))^2 + 3} & \text{if } i \text{ is even} \end{cases}$$

$$L_i = 1 + 6 \sum_{k=1}^i k = 1 + 3i(i + 1)$$

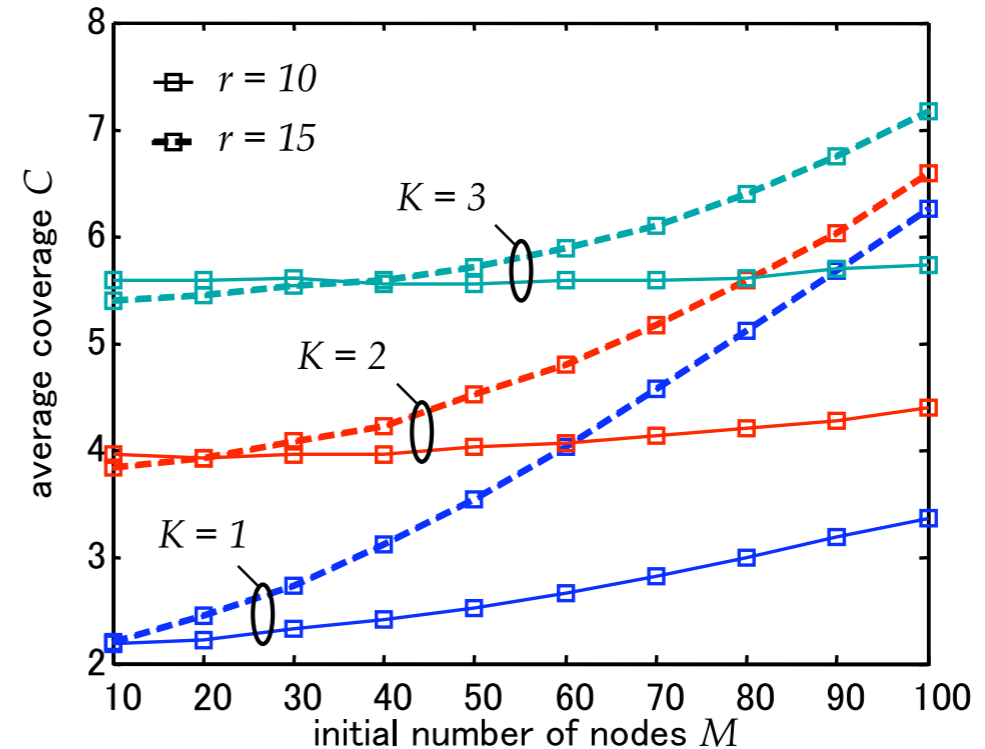
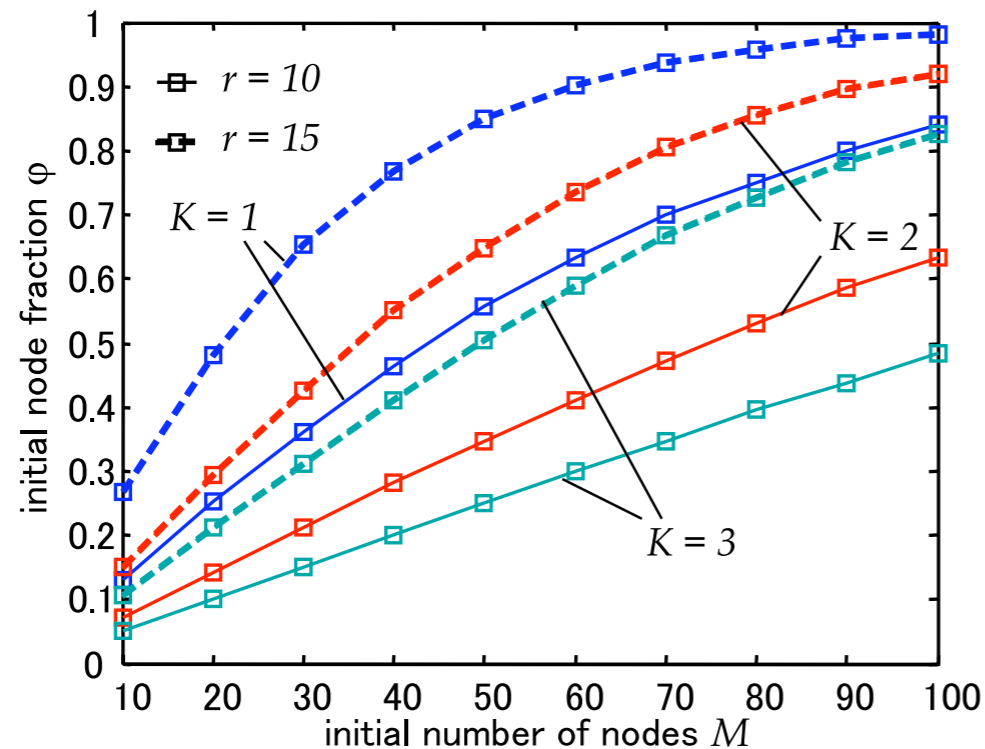
i	1	2	3	4	5	6	7
y_i	30.0	54.1	120.0	143.1	210.0	232.9	300.0

Comparison of Total Nodes



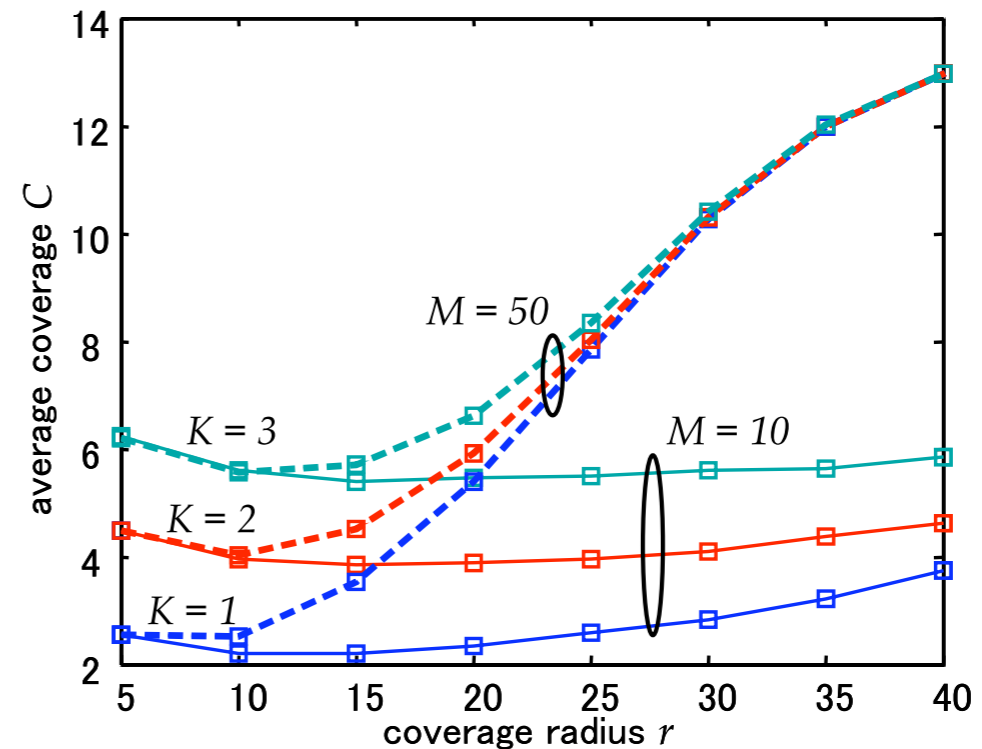
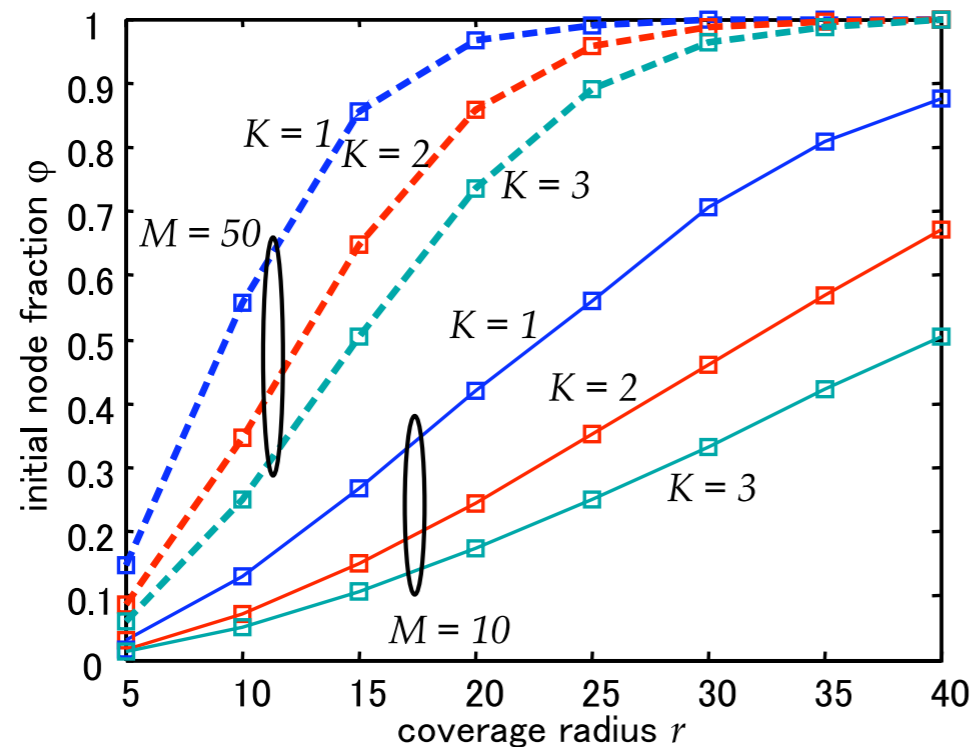
- ▶ Total nodes: M (initial nodes) + N (added nodes)
- ▶ Proposal is better when K gets larger, whereas total nodes remain constant for hexagonal approximation
- ▶ Similar benefit for variation of sensing radius r

Influence of Initial Configuration



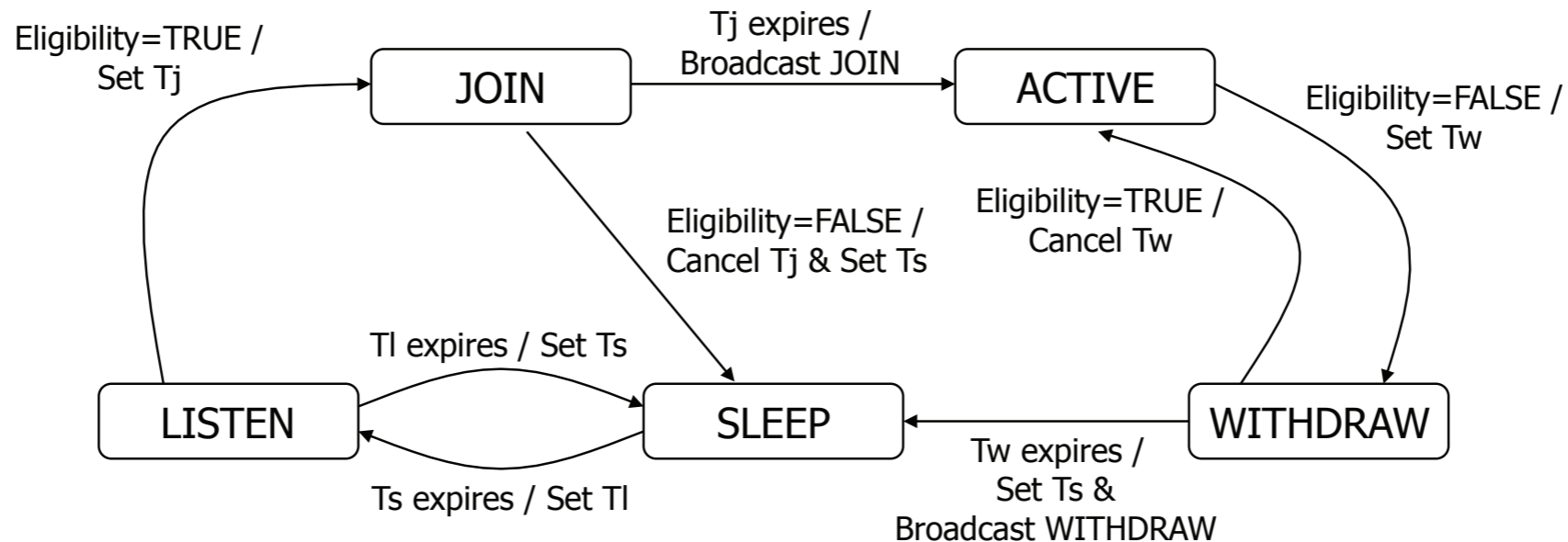
- ▶ Initial coverage fraction defined as $\varphi = M/(M + N)$
- ▶ Smaller coverage radius results in less total nodes
- ▶ Larger target coverage has less influence in conjunction with radius on average coverage C

Influence of Sensing Coverage Radius



- ▶ If initial nodes M is large, only few additional nodes required, especially for large r
- ▶ Proposal is more efficient the less initial nodes exist. For large number of initial nodes and radius, we have much (unnecessary) overlap

Conclusion



- ▶ Fast Gas Bubble method for sensor node deployment
- ▶ Heuristic is based on Neural Gas/Simulated Annealing
- ▶ Redundant coverage planning permits unnecessary nodes to enter sleep state, prolonging network lifetime
- ▶ Combined with energy-dependent state sojourn timers and message filtering