How to generate sensor-actor graphs ?

Atay, Stojmenovic IEEE WoWMoM 2007

Atay, Stojmenovic, Yanikomeroglu, Pervasive and Mobile Computing, to appear.

Generating Random Graphs for the

Simulation of Wireless

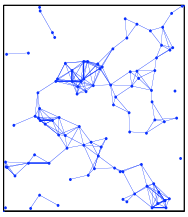
Ad Hoc, Actuator, Sensor, and Internet Networks

Existing generation of ad hoc, sensor, actuator/actor graphs

Standard Algorithm *N = 100 nodes, density d = 5*

choose x and y coordinates of each node at random

CRUG: Connected Random Unit disk Graph



Problems with CRUG

- Did **not** look like **evenly spread** over area
- Would students seat like that in a classroom?
- **Slow** to generate sparse connected networks
- So we want **fast** generation of sparse connected networks, and we want these networks to look more natural, like robots moving as **collaborative team** and fairly dividing area to control

CAG: Connected Actuator Graphs Generation Algorithms (Common Properties)

- Candidate graph:
  - Calculate an approximate transmission range such that *(expected node degree) = d* $r = (Ad/((N-1)\pi))^{1/2}$
  - place *N* nodes sequentially, in *N* rounds.
  - Place the *i*-th node based on the positions of the *(i-1)* previous nodes
- Proximity constraint is satisfied if node-*i* is
  - not isolated from the previous nodes based on the approximate range r and
  - it is no closer than d_{min} to any of the previous nodes.

CAG Generation Algorithms

- Furuzan Atay, Ivan Stojmenovic IEEE WoWMoM 2007
- Center node based Algorithms:
  - Eligible Proximity Algorithm (**EPA**)
  - Weighted Proximity Algorithm (**WPA**)
  - Minimum Degree Proximity Algorithm (**MIN-DPA**)
- Maximum Degree Proximity Algorithm (**MAX-DPA**)

Center node based algorithms

- Distribute degrees more uniformly while maintaining connectivity
- Place the first node randomly in *A*
- In round *i*, choose a center node and place node *i* within the transmission range (*r*) of the center node
- After all the nodes are placed, check for connectivity

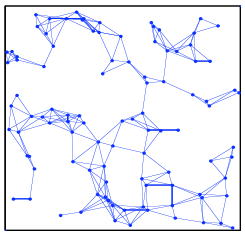
Center node based algorithms

- Center node selection:
  - Calculate the approximate degrees (d_i) of all the nodes already placed based on r
  - **MIN-DPA**: Choose the node with the minimum d_i
  - **EPA**: All the nodes with $d_i < d$ are eligible to accept more neighbors. Choose one of them at random
  - **WPA**: Assign weights to nodes proportional to $(d_i - d)$; Choose one at random according to these weights

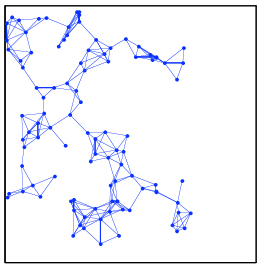
MAX-DPA

- In round i, choose a random position for node i.
- Calculate the approximate degrees (d_i) of all the nodes already placed based on r
- Accept this position if
  - it satisfies proximity test
  - it does not result in $d_i > d_{max}$ where $d_{max} = d+n$ and n is a parameter of the algorithm
- check for connectivity at the end

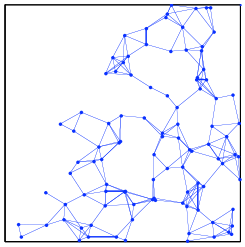
N = 100, d = 5 **Standard Algorithm**



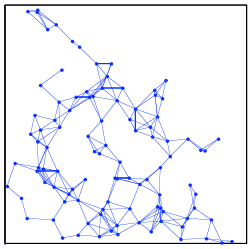
N = 100, d = 5 **MIN-DPA**



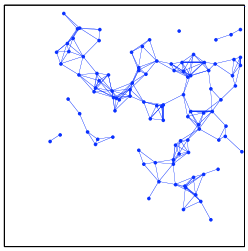
Sample candidate graphs
N = 100, d = 5
MAX-DPA$_2$ (d_{max} = 7)



N = 100, d = 5 **MAX-DPA$_4$** (d_{max} = 9)



N = 100, d = 5 **MAX-DPA$_8$** *(d_{max} = 13)*



Generating sensor networks?

- Add more proximity constraints, e.g.:
- Do not place new sensor if its covering circle is covered already by other sensors
- How to generate realistic sensor and actor networks?
- Wireless Internet networks:
- generate gateways, then new nodes must be connected to one of gateways.

More on graph generation ?

- Faster generation with smaller degree deviations
- Average size of the largest connected component increased
- How well new algorithms model realistic actuator networks?
- Connectivity analysis by formal methods?
- Theoretical differences from random unit graphs?