Spreading Processes on Networks: Models, Techniques and Algorithms

Hans De Sterck  
hans.desterck@monash.edu
School of Mathematical Sciences
Monash University, Melbourne, Australia
(formerly University of Waterloo)
“computational social science”:
- large-scale data is being collected on complex social systems
  - online social networks (Facebook, Twitter, ...)
  - email (gmail, ...)
  - travel patterns (public transportation, google maps, ...)
  - mobile phone connections, locations
  - shopping patterns
  - ...
- it is now possible to build, analyze, and simulate computational models of these systems
- research is possible that applies successful methods from the natural sciences, e.g. mathematical modelling and statistical mechanics, to produce novel insights in the social sciences — to apply “scientific method” to social science
- some of this can be high-performance computing / high-end computing (big data, combinatorial, ...), efficient algorithms, methods
I have started to set some steps in computational social science / network problems / social media (numerical PDEs, numerical linear algebra, numerical optimization, HPC)

- multigrid methods for computing stationary vectors of Markov chains – random walks on graphs (Google PageRank) – Nelly Litvak
- multilevel co-clustering for social networks
- location tagging for Twitter messages
- optimization methods for tensor decomposition, matrix completion, recommendation
- ODE and network models for social uprisings (Arab Spring)
- dynamical models for smoking epidemic and obesity epidemic
- propagation of Susceptible-Infectious-Recovered (SIR) disease on random networks with spatial structure
“propagation of Susceptible-Infectious-Recovered (SIR) disease on random networks with spatial structure”

- “introduction to network science”
- some new results on SIR propagation on random spatial networks

- Spreading Processes on Networks:
  - Models
  - Techniques
  - Algorithms (random network generation, stochastic simulation algorithms)
  - Applications

Random Spatial Networks: Small Worlds without Clustering, Traveling Waves, and Hop-and-Spread Disease Dynamics

John Lang, Hans De Sterck, Jamieson L. Kaiser, Joel C. Miller
collaborators

- Joel Miller, Institute for Disease Modeling, Seattle, USA
- John Lang, UCLA, Communications Studies (PhD Waterloo, July 2016)
Network Science

Class 3: Random Networks
(Chapter 3 in textbook)

Albert-László Barabási
with
Roberta Sinatra

WWW.BARABASILAB.COM

sources (background on network science, SIR disease propagation, ...)

- http://barabasi.com/networksciencebook/

- book by Kiss, Miller and Simon (2017)

Random Spatial Networks: Small Worlds without Clustering, Traveling Waves, and Hop-and-Spread Disease Dynamics

John Lang, Hans De Sterck, Jamieson L. Kaiser, Joel C. Miller
motivation: spread of 2013-2016 Ebola epidemic

- Guinea, Sierra Leone, Liberia

Virus genomes reveal factors that spread and sustained the Ebola epidemic

Gytis Dudas, Luiz Max Carvalho, Trevor Bedford, Andrew J. Tatem, Guy Baele,


- goal: develop modeling framework
- random networks
- spatial structure!
- disease propagation (stochastic, DEs for insight)
two parts of my presentation

- **part A: models and algorithms for networks**
  - “introduction to network science”
  - random spatial networks
  - algorithms for efficient network generation
  - application: small worlds with spatial structure

- **part B: disease propagation on networks**
  - propagation of Susceptible-Infectious-Recovered (SIR) disease
  - stochastic simulation algorithms
  - exact analytic models, and simulations
  - applications
A1: a brief overview of graphs and networks

- graph \( G = (V, E) \) (undirected, simple (no loops, no multiple edges))
  - vertices/nodes \( V = \{v_1, v_2, \ldots, v_n\} \) \( n = |V| \)
  - edges \( E = \{\{v_{i_1}, v_{j_1}\}, \ldots, \{v_{i_m}, v_{j_m}\}\} \) \( m = |E| \)

- recall binomial coefficients
  \[
  \binom{n}{k} = \frac{n!}{(n-k)!k!} \quad (n \geq k)
  \]

- \( m_{\text{max}} = \binom{n}{2} = \frac{n(n-1)}{2} \)
  \[= 6 \quad (n=4)\]

- \( n = 4 \)
  \( m = 5 \)

\( V = \{v_1, v_2, v_3, v_4\} \)
\( E = \{\{v_1,v_2\}, \{v_1,v_3\}, \{v_2,v_3\}, \{v_2,v_4\}, \{v_3,v_4\}\} \)
• degree \( k_i = \text{number of edges incident on node } v_i \)

• average degree \( <k> = \frac{\sum_{i=1}^{n} k_i}{n} \)

• property: \( \sum_{i=1}^{n} k_i = 2m \)

\[
m = \frac{n <k>}{2}
\]

\[
S = \frac{\psi \cdot 2.5}{2}
\]

\( k_1 = 2 \)

\( k_2 = 3 \)

\( <k> = 2.5 \)
degree distribution

- degree distribution
  
  \[ p_k = \text{fraction of nodes with degree } k \]

- degree distribution for random graph model
  
  \[ P(K = k) = P(\text{a node in the random network has degree } k) \]

- many “real-world” networks approximately have power law degree distribution
  
  \[ p_k = c k^{-\gamma} \quad 2 \leq \gamma \leq 3 \]
  
  \[ \ln p_k = \ln c - \gamma \ln k \]
counting triangles - clustering coefficient

- **local clustering coefficient of node** $v_i$
  \[ c_i = \frac{\text{# of triangles node } v_i \text{ forms with its neighbors}}{\text{# of possible triangles node } v_i \text{ can form with its neighbors}} \]

- **neighbor set of node** $v_i$
  \[ N_i = \{v_{i1}, \ldots, v_{ik_i}\} \quad E_{Ni} = \{\text{edges between neighbors of node } v_i\} \]

  then
  \[ c_i = \frac{|E_{Ni}|}{\binom{k_i}{2}} = \frac{2|E_{Ni}|}{k_i(k_i - 1)} \]

- **average clustering coefficient**
  \[ <c> = \frac{\sum_{i=1}^{n} c_i}{n} \]
average shortest path

- \( d_{ij} = \# \) of edges between nodes \( v_i \) and \( v_j \), in a shortest path

- average (shortest) path length (in a connected graph)

\[
< d > = \frac{\sum_{i,j} d_{ij}}{n(n-1)/2}
\]

- many “real-world networks” have
  - large clustering \( < c > \)
  - small average path length \( < d > \)

```
\begin{align*}
1-2 & \quad 1 \\
1-3 & \quad 1 \\
1-4 & \quad 2 \\
2-3 & \quad 1 \\
2-4 & \quad 1 \\
3-4 & \quad 1 \\
\end{align*}
\]

\[
< d > = \frac{7}{6}
\]
example: Watts-Strogatz “small world” networks

- ring, each node connected to four nearest neighbors; randomly rewire with probability $p$

Collective dynamics of ‘small-world’ networks

Duncan J. Watts & Steven H. Strogatz

* Nature 1998

14,000 citations
example: Watts-Strogatz “small world” networks

- small world:
  - high clustering $\langle c \rangle$
    (local structure) (unlike random graph)
  - small average (shortest) path length $\langle d \rangle$
    (good connectivity) (like random graph)

- many “real-world” graphs are small-world

“large world” (e.g., lattice)
  - local structure
  - large (average) shortest path length
  $\langle d \rangle = O(n)$
  1D lattice

$C(p) / C(0)$
$\langle c \rangle$

$L(p) / L(0)$
A2: some random network models

- **Erdos-Renyi networks**: $n$ nodes, and assign edges randomly with probability $p$

  \[ p_{ij} = P(\text{edge } \{v_i, v_j\} \text{ exists}) \]

  \[ p_{ij} = p \]

- degrees and edges:
  \[ k_i \approx (n - 1)p \]
  \[ E(k_i) = (n - 1)p \]
  \[ E(< k >) = (n - 1)p \]

\[ G(n, p) \]
Erdos-Renyi networks: degree distribution

- **binomial degree distribution**
  \[ p_k = P(K = k) = \binom{n-1}{k} p^k (1-p)^{(n-1)-k} \]
  \[ E(k) = (n-1)p \]
  \[ \text{Var}(k) = \sigma^2(k) = p(1-p)(n-1) \]

- **peaked distribution**
  \[ \frac{\sigma(k)}{E(k)} = O \left( \frac{1}{\sqrt{n-1}} \right) \]

- for large \( n \), small \( k \): approximately Poisson distributed
  \[ p_k \approx \exp(-\lambda) \frac{\lambda^k}{k!} \]
  \[ E(k) = \lambda \]
  \[ \lambda = (n-1)p \]
degree distributions of “real” networks: power law graphs – scale-free networks

\[ p_k = ck^{-\gamma} \]

\[ 2 \leq \gamma \leq 3 \]

[Barabasi]
“real world” networks with power law distribution (approximately)

<table>
<thead>
<tr>
<th>Network</th>
<th>$N$</th>
<th>$L$</th>
<th>$\langle k \rangle$</th>
<th>$\langle k_{in}^2 \rangle$</th>
<th>$\langle k_{out}^2 \rangle$</th>
<th>$\langle k^2 \rangle$</th>
<th>$\gamma_{in}$</th>
<th>$\gamma_{out}$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>192,244</td>
<td>609,066</td>
<td>6.34</td>
<td>-</td>
<td>-</td>
<td>240.1</td>
<td>-</td>
<td>-</td>
<td>3.42*</td>
</tr>
<tr>
<td>WWW</td>
<td>325,729</td>
<td>1,497,134</td>
<td>4.60</td>
<td>1546.0</td>
<td>482.4</td>
<td>-</td>
<td>2.00</td>
<td>2.31</td>
<td>-</td>
</tr>
<tr>
<td>Power Grid</td>
<td>4,941</td>
<td>6,594</td>
<td>2.67</td>
<td>-</td>
<td>-</td>
<td>10.3</td>
<td>-</td>
<td>-</td>
<td>Exp.</td>
</tr>
<tr>
<td>Mobile-Phone Calls</td>
<td>36,595</td>
<td>91,826</td>
<td>2.51</td>
<td>12.0</td>
<td>11.7</td>
<td>-</td>
<td>4.69*</td>
<td>5.01*</td>
<td>-</td>
</tr>
<tr>
<td>Email</td>
<td>57,194</td>
<td>103,731</td>
<td>1.81</td>
<td>94.7</td>
<td>1163.9</td>
<td>-</td>
<td>3.43*</td>
<td>2.03*</td>
<td>-</td>
</tr>
<tr>
<td>Science Collaboration</td>
<td>23,133</td>
<td>93,437</td>
<td>8.08</td>
<td>-</td>
<td>-</td>
<td>178.2</td>
<td>-</td>
<td>-</td>
<td>3.35*</td>
</tr>
<tr>
<td>Actor Network</td>
<td>702,388</td>
<td>29,397,908</td>
<td>83.71</td>
<td>-</td>
<td>-</td>
<td>47,353.7</td>
<td>-</td>
<td>-</td>
<td>2.12*</td>
</tr>
<tr>
<td>Citation Network</td>
<td>449,673</td>
<td>4,689,479</td>
<td>10.43</td>
<td>971.5</td>
<td>198.8</td>
<td>-</td>
<td>3.03*</td>
<td>4.00*</td>
<td>-</td>
</tr>
<tr>
<td>E. Coli Metabolism</td>
<td>1,039</td>
<td>5,802</td>
<td>5.58</td>
<td>535.7</td>
<td>396.7</td>
<td>-</td>
<td>2.43*</td>
<td>2.90*</td>
<td>-</td>
</tr>
<tr>
<td>Protein Interactions</td>
<td>2,018</td>
<td>2,930</td>
<td>2.90</td>
<td>-</td>
<td>-</td>
<td>32.3</td>
<td>-</td>
<td>-</td>
<td>2.89*</td>
</tr>
</tbody>
</table>

Table 4.1

[Barabasi]
degree distributions of “real” networks: power law graphs – scale-free networks

\[ p_k = ck^{-\gamma} \quad 2 \leq \gamma \leq 3 \]

\[ E(k^l) \quad l > \gamma - 1 \quad \text{diverges} \]

\[ \gamma < 3 : l = 2, 3, \ldots \text{ diverge} \]

\[ E(k^2) \quad \text{diverges} \]

**Random Network**
Randomly chosen node: \( k = \langle k \rangle \pm \langle k \rangle^{1/2} \)
Scale: \( \langle k \rangle \)

**Scale-Free Network**
Randomly chosen node: \( k = \langle k \rangle \pm \infty \)
Scale: none

[Barabasi]
Our network.

Road network.

(a) POISSON
Most nodes have the same number of links
No highly connected nodes

(b) Map of the US with cities labeled.

(c) POWER LAW
Many nodes with only a few links
A few hubs with large number of links

(d) More detailed map with additional connections.

[Barabasi]
Erdos-Renyi networks: average shortest path

- small-world network!

\[ d_{ij} = \# \text{ of edges between nodes } v_i \text{ and } v_j, \text{ in a shortest path} \]

\[ E(d) \approx \frac{\log(n)}{\log(E(k))} \]

- compare
  - 1D lattice: \( E(d) = O(n) \)
  - 2D lattice: \( E(d) = O(\sqrt{n}) \)

average shortest path: \( E(d) = O(\log(n)) \), on slower

large worlds
“real-world” networks: average shortest path

- small worlds!

\[ E(d) \approx \frac{\log(n)}{\log(E(k))} \]

<table>
<thead>
<tr>
<th>Network</th>
<th>N</th>
<th>L</th>
<th>\langle k \rangle</th>
<th>\langle d \rangle</th>
<th>d_{\text{max}}</th>
<th>lnN/ln\langle k \rangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>192,244</td>
<td>609,066</td>
<td>6.34</td>
<td>6.98</td>
<td>26</td>
<td>6.58</td>
</tr>
<tr>
<td>WWW</td>
<td>325,729</td>
<td>1,497,134</td>
<td>4.60</td>
<td>11.27</td>
<td>93</td>
<td>8.31</td>
</tr>
<tr>
<td>Power Grid</td>
<td>4,941</td>
<td>6,594</td>
<td>2.67</td>
<td>18.99</td>
<td>46</td>
<td>8.66</td>
</tr>
<tr>
<td>Mobile-Phone Calls</td>
<td>36,595</td>
<td>91,826</td>
<td>2.51</td>
<td>11.72</td>
<td>39</td>
<td>11.42</td>
</tr>
<tr>
<td>Email</td>
<td>57,194</td>
<td>103,731</td>
<td>1.81</td>
<td>5.88</td>
<td>18</td>
<td>18.4</td>
</tr>
<tr>
<td>Science Collaboration</td>
<td>23,133</td>
<td>93,437</td>
<td>8.08</td>
<td>5.35</td>
<td>15</td>
<td>4.81</td>
</tr>
<tr>
<td>Actor Network</td>
<td>702,388</td>
<td>29,397,908</td>
<td>83.71</td>
<td>3.91</td>
<td>14</td>
<td>3.04</td>
</tr>
<tr>
<td>Citation Network</td>
<td>449,673</td>
<td>4,707,958</td>
<td>10.43</td>
<td>11.21</td>
<td>42</td>
<td>5.55</td>
</tr>
<tr>
<td>E. Coli Metabolism</td>
<td>1,039</td>
<td>5,802</td>
<td>5.58</td>
<td>2.98</td>
<td>8</td>
<td>4.04</td>
</tr>
<tr>
<td>Protein Interactions</td>
<td>2,018</td>
<td>2,930</td>
<td>2.90</td>
<td>5.61</td>
<td>14</td>
<td>7.14</td>
</tr>
</tbody>
</table>

Table 3.2

[Barabasi]
Erdos-Renyi networks: clustering coefficient

\[ c_i = \frac{2|E_{N_i}|}{k_i(k_i - 1)} \approx p = \frac{E(k)}{n - 1} \]

- Low clustering as \( n \) grows

- But: many “real-world” networks have high average clustering coefficients

<table>
<thead>
<tr>
<th>Network</th>
<th>( C )</th>
<th>Erdös-Rényi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web [2]</td>
<td>0.081</td>
<td>7.71</td>
</tr>
<tr>
<td>Flickr</td>
<td>0.313</td>
<td>47,200</td>
</tr>
<tr>
<td>LiveJournal</td>
<td>0.330</td>
<td>119,000</td>
</tr>
<tr>
<td>Orkut</td>
<td>0.171</td>
<td>7,240</td>
</tr>
<tr>
<td>YouTube</td>
<td>0.136</td>
<td>36,900</td>
</tr>
</tbody>
</table>

(A. Bonato)
random networks with more realistic degree distributions: Chung-Lu networks

- $n$ nodes, desired degree sequence $\{\kappa_1, \ldots, \kappa_n\}$

- add edges randomly according to
  
  $$p_{ij} = \frac{\kappa_i \kappa_j}{n \langle \kappa \rangle}$$  
  $$\langle \kappa \rangle = \frac{\sum_{i=1}^{n} \kappa_i}{n}$$

  $$p_{ij} = \max\left(\frac{\kappa_i \kappa_j}{n \langle \kappa \rangle}, 1\right)$$

  (advantage: edges assigned independently, which keeps the model “analyzable” ... see later)
Chung-Lu networks: expected degree

- assume

\[ \frac{\kappa_i \kappa_j}{n < \kappa >} \leq 1 \]

\[
E(k_i) = \sum_{j=1, j \neq i}^{n} p_{ij} \\
= \sum_{j=1, j \neq i}^{n} \frac{\kappa_i \kappa_j}{n < \kappa >} \\
= \sum_{j=1}^{n} \frac{\kappa_i \kappa_j}{n < \kappa >} - \frac{\kappa_i^2}{n < \kappa >} \\
= \kappa_i - \frac{\kappa_i^2}{n < \kappa >} \\
\approx \kappa_i
\]

- Chung-Lu as a model for “real-world” networks:
  - path length \( < d > \) OK!
  - degree distribution \( p_k \) OK!
  - clustering coefficient \( < c > \) too low ...
Chung-Lu networks: choose desired degrees from distribution

- note: desired degree sequence \( \{\kappa_1, \ldots, \kappa_n\} \) can be chosen from (continuous) distribution

\[
E(\kappa) = \int_0^\infty \kappa p(\kappa) \, d\kappa
\]

\[
p_{ij} = \frac{\kappa_i \kappa_j}{n < \kappa >}
\]

\[
E(k_i) = \int_0^\infty \frac{\kappa_i \kappa}{n \int_0^\infty \kappa p(\kappa) \, d\kappa} \, np(\kappa) \, d\kappa = \kappa_i
\]

- if node \( v_i \) is assigned desired degree \( \kappa_i \),
  its expected degree is \( \kappa_i \)
A3: random spatial networks (RSNs)

- Chung-Lu:
  \[ p_{ij} = \frac{\kappa_i \kappa_j}{n < \kappa >} \]

- with spatial structure: in domain \( \Omega \)
  \[ p_{ij} = \frac{\kappa_i \kappa_j}{n < \kappa >} \left| \Omega \right| f(d_{ij}) \]
  \[ d_{ij} = \| \vec{x}_i - \vec{x}_j \|_2 \]

\[ \int_{\Omega} f(\| \vec{x}_0 - \vec{x} \|_2) d\vec{x} = 1 \quad \forall \vec{x}_0 \in \Omega \]

- more generally:
  \[ p_{ij} = \min \left( \frac{\kappa_i \kappa_j}{\rho < \kappa >} f(d_{ij}), 1 \right) \]
  \[ \rho = \frac{n}{\left| \Omega \right|} \]
random spatial networks (RSNs)

\[ p_{ij} = \min \left( \frac{\kappa_i \kappa_j}{\zeta}, f(d_{ij}), 1 \right) \]

FIG. 1. An example RSN and its properties. The distance kernel is a Gaussian, \( f(d) = \exp(-d^2/2\sigma^2)/2\pi\sigma^2 \) with \( \sigma = 0.03 \). The imposed distribution of expected degrees is \( P(2) = P(15) = 0.5 \). The density is \( \rho = 10000 \). One node and its neighbors are highlighted. A random network without spatial structure would exhibit neighbors throughout the domain.
random spatial networks (RSNs): expected degree

\[ p_{ij} = \min \left( \frac{\kappa_i \kappa_j}{\rho \kappa} f(d_{ij}), 1 \right) \]

\[
E(k_i) = \int_{\kappa=0}^{\infty} \left( \int_{\omega} \frac{k_i \kappa}{\rho} \int_{\kappa'}^{\infty} k p(\kappa') d\kappa' \int_{\Omega} f(||\vec{x}_i - \vec{x}'||_2) \rho d\vec{x}' \right) p(\kappa) d\kappa \\
= k_i \int_{0}^{\infty} \frac{k p(\kappa)}{\kappa p(\kappa')} d\kappa' \int_{\Omega} f(||\vec{x}_i - \vec{x}'||_2) d\vec{x}' \\
= k_i \]

If node \( v_i \) is assigned desired degree \( \kappa_i \), its expected degree is \( \kappa_i \)
A4: efficient algorithms for network generation

- **Erdos-Renyi:** \( G(n, p) \)
  
  \[ p_{ij} = p \]

- **naive algorithm**
  - for each possible edge \( \{v_i, v_j\} \), draw a random number

  \[
  m_{\text{max}} = \binom{n}{2} = \frac{n(n - 1)}{2}
  \]

  - complexity \( O(n^2) \)
efficient algorithms for network generation

- **Erdos-Renyi**: $O(n+m)$ algorithm \( p_{ij} = p \)

  - naive algorithm has many failures (when \( p \) is small)

  - idea: we know the distribution of successes and failures! (geometric); so sample the number of failures according to the appropriate distribution

\[
\delta = \text{number of failures before success}
\]

\[
P(\delta) = P(\text{first success happens at trial } \delta + 1) = (1 - p)^\delta p
\]

\[
P(D \leq \delta) = P(\text{first success happens in trial } 1, 2, \ldots, \text{ or } \delta + 1)
= 1 - P(\text{no success in first } \delta + 1 \text{ trials})
= 1 - (1 - p)^{\delta + 1}
\]
efficient algorithms for network generation

- **Erdos-Renyi: \(O(n+m)\) algorithm**
  
  \[ p_{ij} = p \quad q = 1 - p \]

  \[ \delta = \text{number of failures before success} \]

  \[ P(D \leq \delta) = P(\text{first success happens in trial 1, 2, \ldots, or } \delta + 1) \]

  \[ = 1 - P(\text{no success in first } \delta + 1 \text{ trials}) \]

  \[ = 1 - (1 - p)^{\delta+1} \]

- **Choose** \( r \in [0, 1] \)

  \[ \delta = \frac{\ln(1 - r)}{\ln(1 - p)} \leq \delta + 1 \]

  \[ \delta = \left\lfloor \frac{\ln(1 - r)}{\ln(1 - p)} \right\rfloor \]
Algorithm 1. $G(N, p)$ Graph

Input: number of nodes $N$, and probability $0 < p < 1$

Output: $G(N, p)$ graph $G(V, E)$ with $V = \{0, \ldots, N - 1\}$

$E \leftarrow \emptyset$

for $u = 0$ to $N - 2$

$v \leftarrow u + 1$

while $v < N$

choose $r \in (0, 1)$ uniformly at random

$v \leftarrow v + \left\lfloor \frac{\log(r)}{\log(1-p)} \right\rfloor$

if $v < N$

$E \leftarrow E \cup \{u, v\}$

$v \leftarrow v + 1$

end

end

$O(n + m)$ (random numbers)
Chung-Lu: $O(n+m)$ algorithm

- Order nodes $v_i$ in order of decreasing desired degree $\kappa_i$
- For fixed $i$:
  - Fix $p = p_{i,i+1}$
  - Determine number of failures $\delta$ as before (using $p$)
  - $v_{i+1+\delta}$ is a potential neighbor: accept with probability $\frac{q}{p} \leq 1$ where $q = p_{il} \leq p$
  - (Then $p \frac{q}{p} = q = p_{il}$) ("rejection sampling")
  - Set $p = q$, compute next $\delta$ (repeat...)
- Acceptance remains high! (increasingly large jumps) $O(n+m)$ can be shown
Algorithm 2. Chung-Lu Graph

Input: list of $N$ weights, $W = w_0, \ldots, w_{N-1}$, sorted in decreasing order
Output: Chung-Lu graph $G(V, E)$ with $V = \{0, \ldots, N - 1\}$

$E \leftarrow \emptyset$
$S \leftarrow \sum_u w_u$

for $u = 0$ to $N - 2$ do
    $v \leftarrow u + 1$
    $p \leftarrow \min(w_u w_v / S, 1)$
    while $v < N$ and $p > 0$ do
        if $p \neq 1$ then
            choose $r \in (0, 1)$ uniformly at random
            $v \leftarrow v + \left\lceil \frac{\log(r)}{\log(1 - p)} \right\rceil$
        endif
        if $v < N$ then
            $q \leftarrow \min(w_u w_v / S, 1)$
            choose $r \in (0, 1)$ uniformly at random
            if $r < q/p$ then
                $E \leftarrow E \cup \{u, v\}$
            endif
        endif
    endwhile
endfor

$O(n + m)$ can be shown (acceptance is kept high)

Extra notes:
- $p = 1$: $S = 0$ (nothing added)
Random Spatial Networks (RSNs): $O(n+m)$ algorithm

- $p_{ij} = \min \left( \frac{\kappa_i \kappa_j}{\rho < \kappa > f(d_{ij})}, 1 \right)$

  - Assume $f(d_{ij})$ bounded, decreasing in $d_{ij}$
  - Divide $\Omega$ into $\ell$ subdomains $\Omega_j$ ($1 \leq j \leq \ell$)
  - Order nodes in each subdomain by decreasing desired degree

\[ p = \frac{\kappa_i \kappa_j}{\rho < \kappa > f_{\text{max}}} \]

Determine number of failures $\delta$ as before (using $p$)
Random Spatial Networks (RSNs): $O(n+m)$ algorithm

$p = \frac{\kappa_i \kappa_j}{\rho < \kappa > f(d_{ij})} f_{\text{max}} \quad \text{make sure } p \geq P_{ij}$

determine number of failures $\delta$ as before (using $p$)
node $u$ in region $\Omega_1$

» edges within region 1:

$f_{\text{max}} = f(0)$

» edges to region $\Omega_2$

find point $\vec{x}$ in $\Omega_2$ nearest to node $u$ in region $\Omega_1$

$f_{\text{max}} = f(\|\vec{x} - \vec{x}_u\|_2) \quad \text{f}_{\text{max}} \text{ is chosen small, but } p > P_{ij}$

– acceptance remains high! (increasingly large jumps) $O(n + m)$

$p_{ij} = \min \left( \frac{\kappa_i \kappa_j}{\rho < \kappa > f(d_{ij})}, 1 \right)$
A5: small world networks with spatial structure (RSNs)

- recall: Watts-Strogatz “small world” networks

- small world:
  - large average local clustering coefficient
  - small average (shortest) path length
small world networks with spatial structure (RSNs)

- RSNs: define local proximity coefficient
  - first normalize all distances: \( \overline{d}_{ij} \) normalized to \([0, 1]\)
  
  - define local proximity coefficient:
    \[
p_i = 1 - \operatorname{avg}(\overline{d}_{ij} \text{ of graph neighbors } j \text{ of } i) \quad p_i \in [0, 1]
    \]
    
    \( p_i \approx 1 \): graph neighbors of \( i \) are located close
    \( p_i \approx 0 \): graph neighbors of \( i \) are located far
  
  - average local proximity coefficient:
    \[
    \langle p \rangle
    \]
small worlds with spatial structure

- a specific class of RSNs: mostly local connections, few global connections

\[
\kappa_i = k \quad |\Omega = 1| \quad \text{choose} \quad r_0
\]

\[
p_{ij} = \min \left( \frac{\kappa_i \kappa_j}{\rho \kappa} f(d_{ij}), 1 \right)
\]

\[
f(d_{uv}) = \begin{cases} 
\frac{N_0}{k} \left[ 1 - \frac{1 - \pi r_0^2}{\pi r_0^2} \right] & d_{uv} < r_0 \\
\frac{N_0}{k} \epsilon & d_{uv} \geq r_0
\end{cases}
\]

\[
p_{uv} = \min \left( k \frac{f(d_{uv})}{N}, 1 \right) = \frac{k f(d_{uv})}{N}
\]

\[
p_{uv} = \begin{cases} 
\frac{N_0}{N} \left( 1 - \epsilon \frac{1 - \pi r_0^2}{\pi r_0^2} \right) & d_{uv} < r_0 \\
\frac{N_0}{N} \epsilon & d_{uv} \geq r_0
\end{cases}
\]
small worlds with spatial structure

- a specific class of RSNs: mostly local connections, few global connections

\[ f(d_{uv}) = \begin{cases} \frac{N_0}{k} \left[ 1 - \epsilon \frac{1 - \pi r_0^2}{\pi r_0^2} \right] & d_{uv} < r_0 \\ \frac{N_0}{k} \epsilon & d_{uv} \geq r_0 \end{cases} \]

\[ p_{uv} = \begin{cases} \frac{N_0}{N} \left( 1 - \epsilon \frac{1 - \pi r_0^2}{\pi r_0^2} \right) & d_{uv} < r_0 \\ \frac{N_0}{N} \epsilon & d_{uv} \geq r_0 \end{cases} \]

- case \( n = N_0, \epsilon = 0 \):
  \[ k \frac{N_0}{N} = \frac{\pi r_0^2}{1} \]
  \[ \langle c \rangle \approx 0.59 \]
  all \( k \) neighbors in disc are connected to \( u \)

- case \( n = N_0, \epsilon \neq 0 \): some long-range connections

- case \( n = N \gg N_0, \epsilon \neq 0 \)
  \[ \langle c \rangle = O(1/N) \]

- low density, low clustering, but proximity remains the same
small worlds with spatial structure

- low density, $N=N_0$: small world
  - high clustering, high proximity
  - low (average) shortest path

- high density, $N>>N_0$: unclustered small world
  - low clustering
  - high proximity: local
  - low (average) shortest path: global
  - still small world! (local, and global, structure)
small-world effect:
- local propagation (traveling wave) due to local structure (proximity, not clustering)
- long-range jumps due to small-world property

for spatial networks, proximity is important in determining whether small-world effects occur, rather than clustering.