Brain network model dynamical simulations

Vince Varga, Géza Ódor

June 23, 2017
Budapest
GPU Day
Wigner Research Centre for Physics of the Hungarian Academy of Sciences

We ported our cellular automata spreading simulation from C to CUDA C to see if we could run simulations faster.

- Brain networks, connectome and criticality
- Cellular automata and spreading modell
- Port to CUDA C, what we learned and achieved
- Results of the simlations briefly
- New algorithm? More data?

Brain networks

- networks of neurons comprise the brain
- brain is more than the sum of their individual components
- the brain neuronal communication capacity and computational power relies on complex connectivity architecture in brain
- "wiring diagrams" might help us understand how brain underlies behavior and how brain malfunctions underlie behavioral disorders

Scales of analysis

- macroscale: SPECT, multimodal MRI
- microscale: array tomography, calcium imaging
- nanoscale: electron microscopy

Brain networks

- networks of neurons comprise the brain
- brain is more than the sum of their individual components
- the brain neuronal communication capacity and computational power relies on complex connectivity architecture in brain
- "wiring diagrams" might help us understand how brain underlies behavior and how brain malfunctions underlie behavioral disorders

Scales of analysis

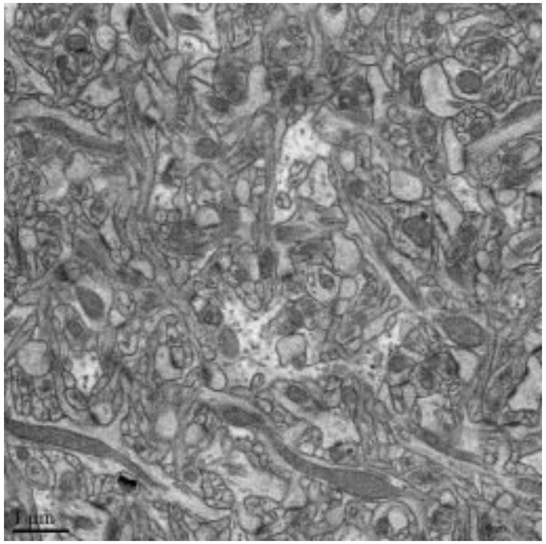
- macroscale: SPECT, multimodal MRI
- microscale: array tomography, calcium imaging
- nanoscale: electron microscopy

Brain networks

- networks of neurons comprise the brain
- brain is more than the sum of their individual components
- the brain neuronal communication capacity and computational power relies on complex connectivity architecture in brain
- "wiring diagrams" might help us understand how brain underlies behavior and how brain malfunctions underlie behavioral disorders

Scales of analysis

- macroscale: SPECT, multimodal MRI
- microscale: array tomography, calcium imaging
- nanoscale: electron microscopy



Connectome

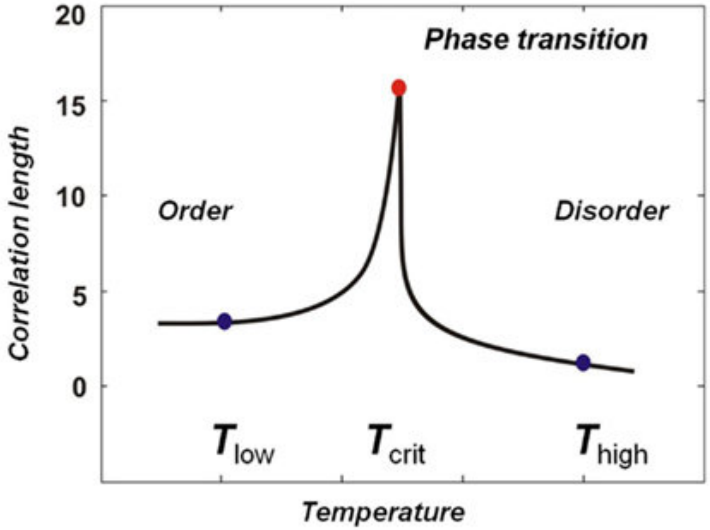
Terms referring to some large scale data in the area of biomedical research: genome, proteome, metabolome.

- the connection matrix of the human brain
- set of neuronal connections
- the network of the human brain is still largely unknown
- the largest precisely explored structural network contains n = 302 neurons (C. elegans, roundworm), human brain is $n \approx 10^{11}$ neurons
- MRI: mm^3 resolution, $N \approx 10^6$ nodes

Connectome: today refers to essentially any neuroscientific investigation of the relationship between neurons or collections of neurons

Criticality

- Networks of neurons produce avalanches of activity with power law distribution.
- Brain might operate near a critical point, poised between a phase where activity rapidly dies out and phase where activity is amplified over time.
- Simulations suggest information processing functions would be optimized at the critical point.
    - communication (correlation length)
    - informational storage
    - computational power
    - dynamic range
- This hypothesis is still controversial.



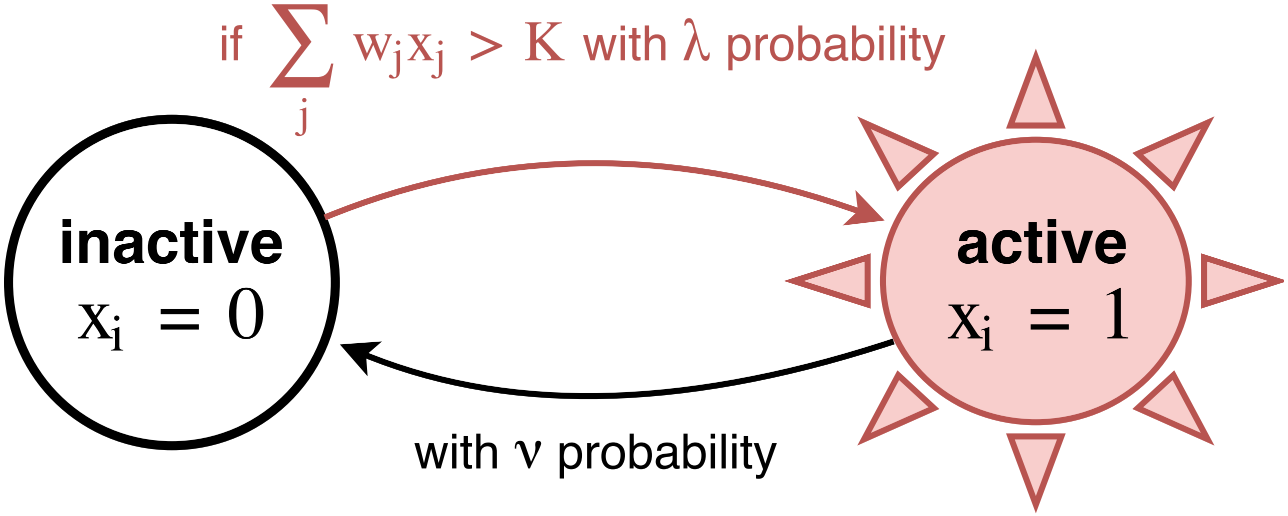
Cellular automata

Simple components together are capable of complex behaviour. Mathematical model.

- consists of simple identical components (nodes)
- nodes interact locally
- nodes with a finite set of possible values
- according to identical rules, the value of nodes evolve synchronously in discrete time steps
- value of a particular node is determined by the previous values of a neighborhood of sites around it

Spreading model

- activation probability λ
- deactivation probability ν
- threshold K
- undirected edges
- inhibitory edges



if $\sum_j w_j x_j > K$ with λ probability

inactive

$x_i = 0$

active

$x_i = 1$

with ν probability

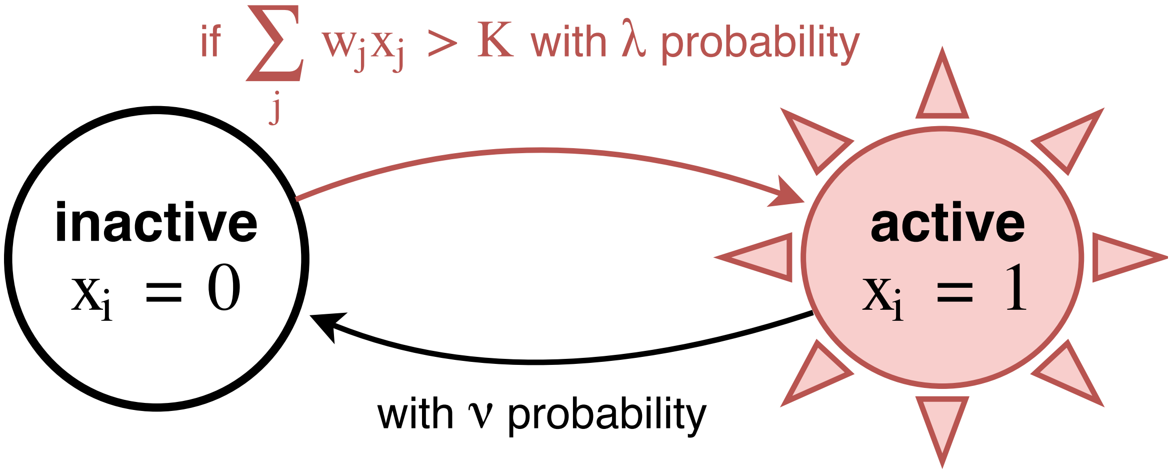
High Level Overview

1. read source graph
2. transform data structure into a more convenient format
3. run simulations for all parameters
4. write relevant results to file

```
network, weight = read_source_network('filename.dat',
                                      undirected_edges,
                                      inhibitory_edges)
network, weight, pos = transform_data_struct(network, weight)
for p in params:
    results = run_simulations(network, weight, pos,
                              p.repeat, p.max_time_steps
                              p.deactivate, p.activate,
                              p.threshold)
    write_relevant_results(results, p)
print("That's all Folks!")
```

Spreading simulation

```
# This is one discrete time step
p = parameters  # activation, deactivation probability, etc...
for node in network.nodes: ## PARALLEL ##
    if node.active:
        node.deactivate_for_next_step(with_probability=p.nu)
    else:
        active_neighbors = 0
        for neighbor in node.neighbors:
            if neighbor.active:
                active_neighbors += node.weights(neighbor)
        if active_neighbors > p.threshold:
            node.activate_for_next_step(with_probability=p.lambda)
```



Benchmarks

- Calculating the state of a node is done in one thread.
- **Calculations in one thread are independent of other threads, no synchronization necessary in a discrete time step**
- **Network is large and inhomogeneous, therefore we stored the network and the state of the nodes in global memory**
- **Need to calculate sum of neighbors, but degree of nodes vary as network is inhomogeneous. Some thread may be waiting for other where node has a high degree (many neighbors)**

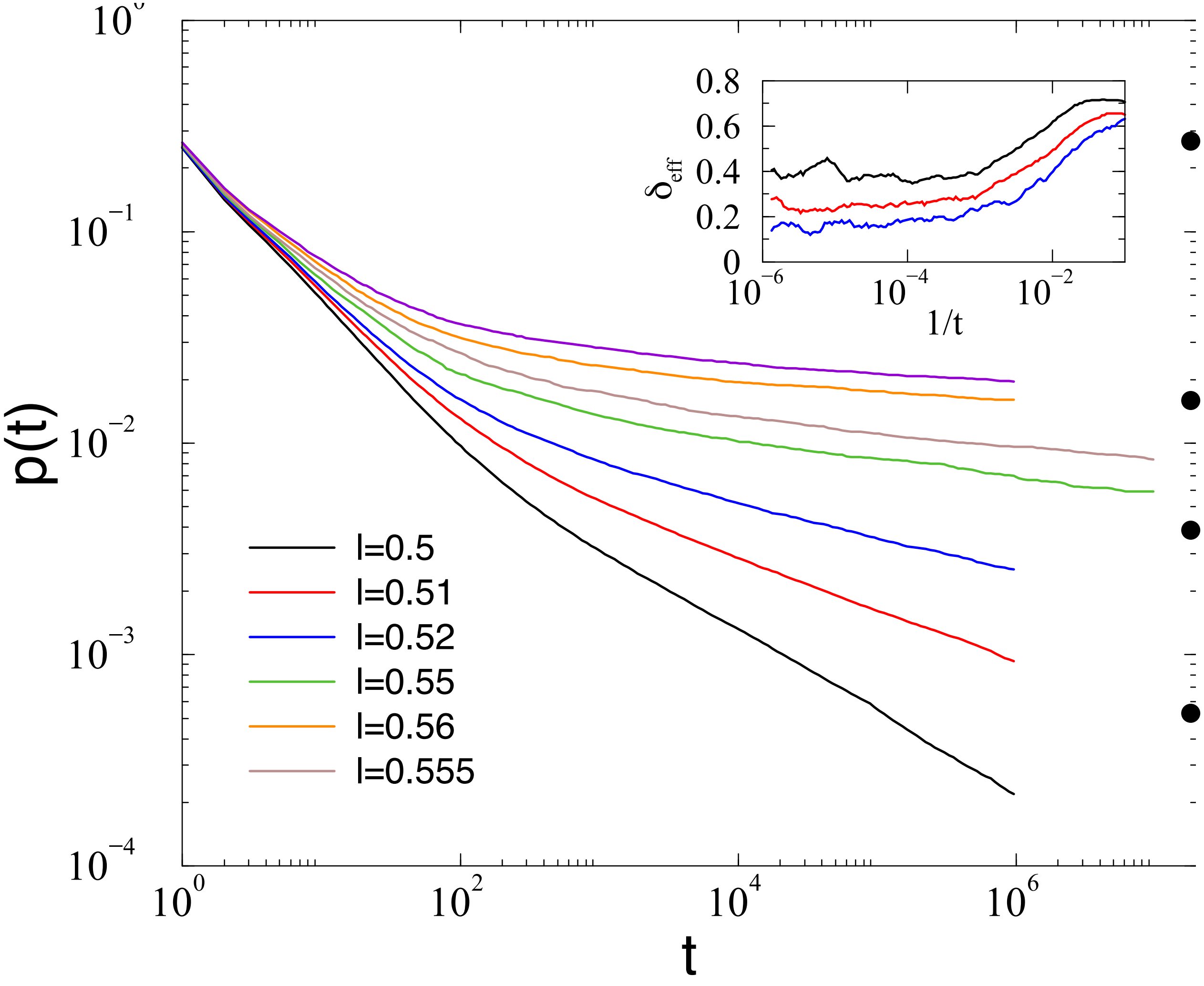
6-12x

Good, but not great.
14 days → 28 hours

$$P(t) \propto t^{-\delta}$$

$$\delta_{\mathrm{eff}} = -\frac{\ln P(t) - \ln P(t')}{\ln t - \ln t'}$$

$$t - t' = 8$$

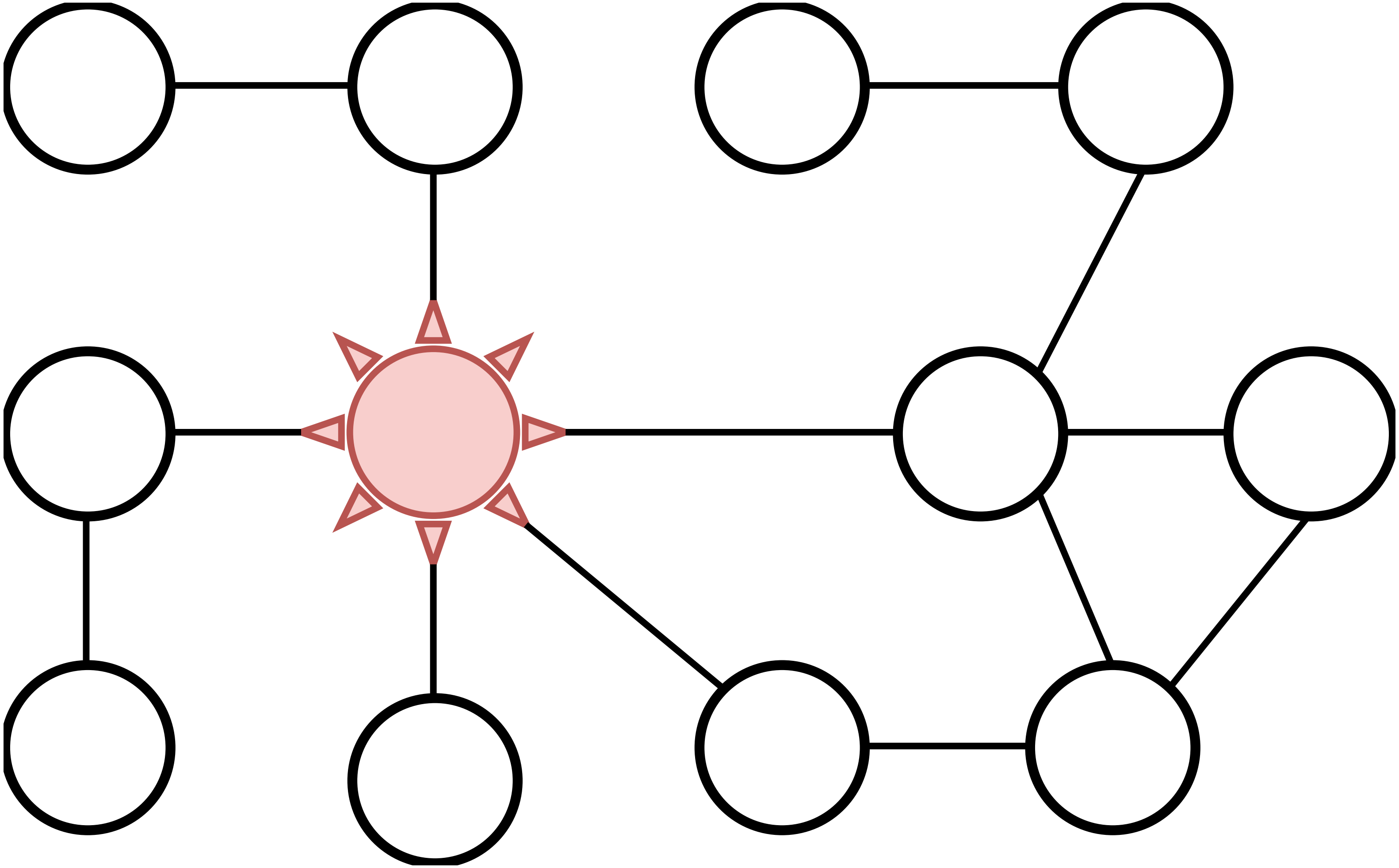


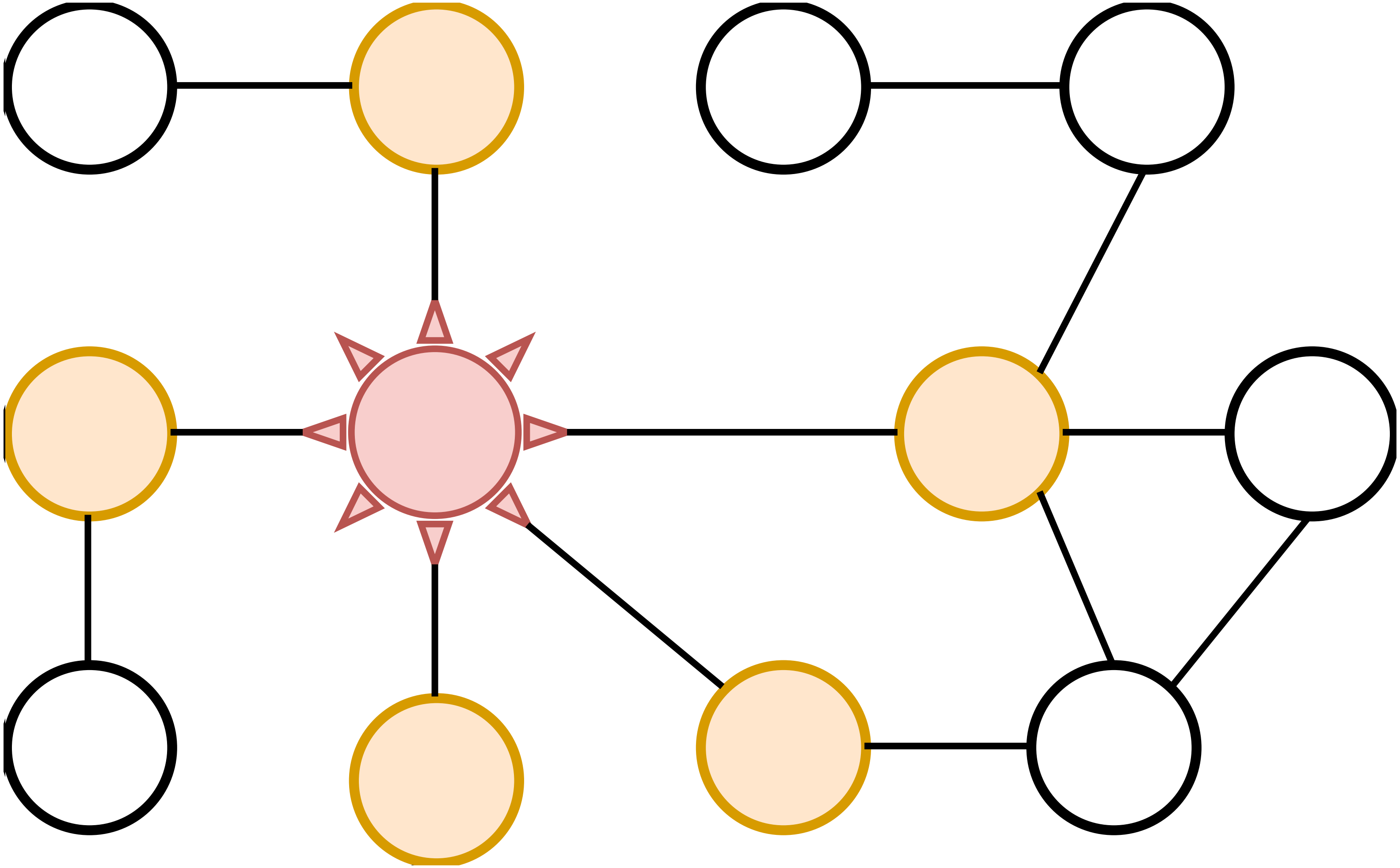
- $P(t)$ is survival probability: probability of the network having at least one active node at t time step
- λ activation probability changes from 0.5 to 0.555
- 30% of the nodes were made inhibitory (negative weight).
- power law distribution

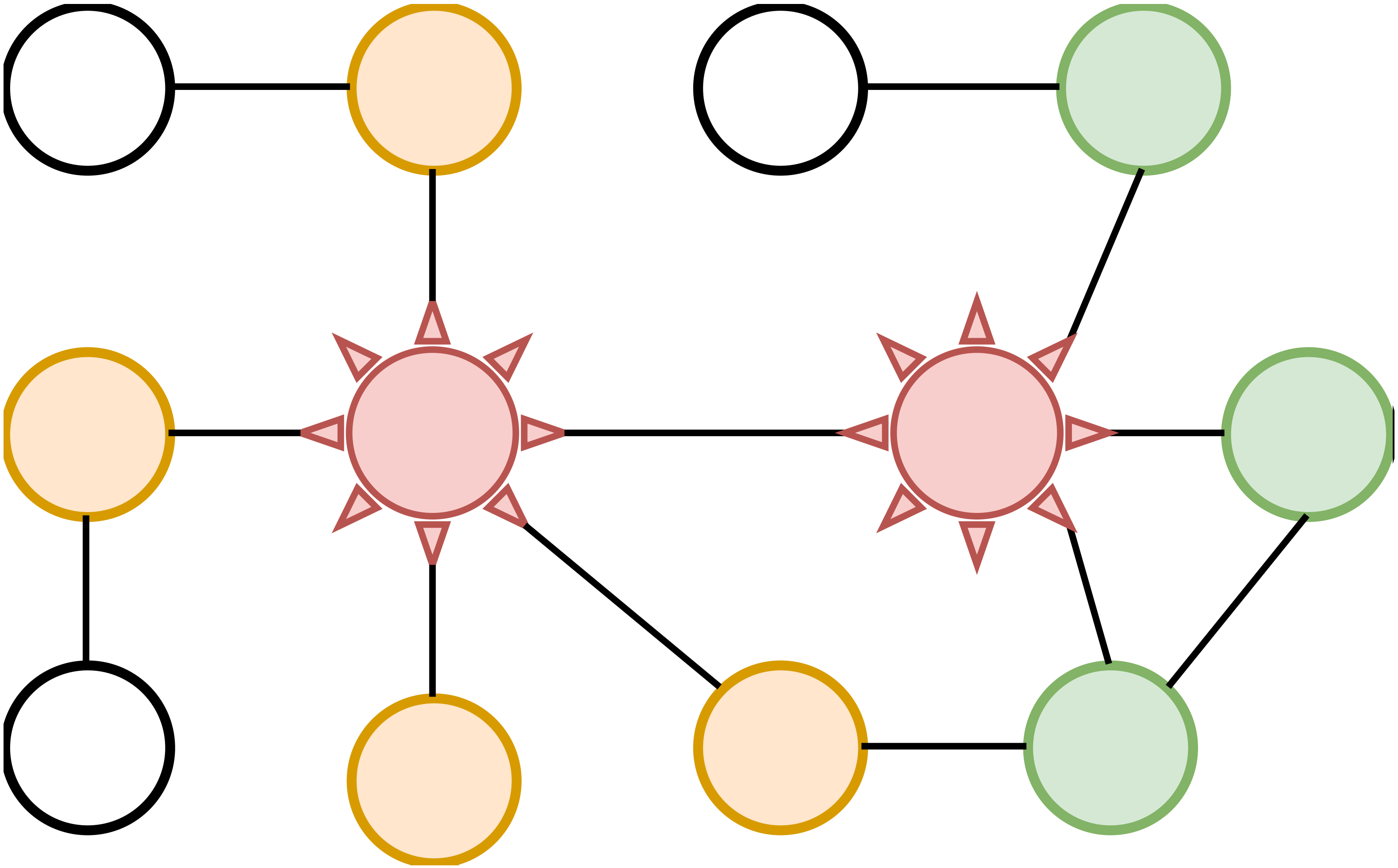
Plans

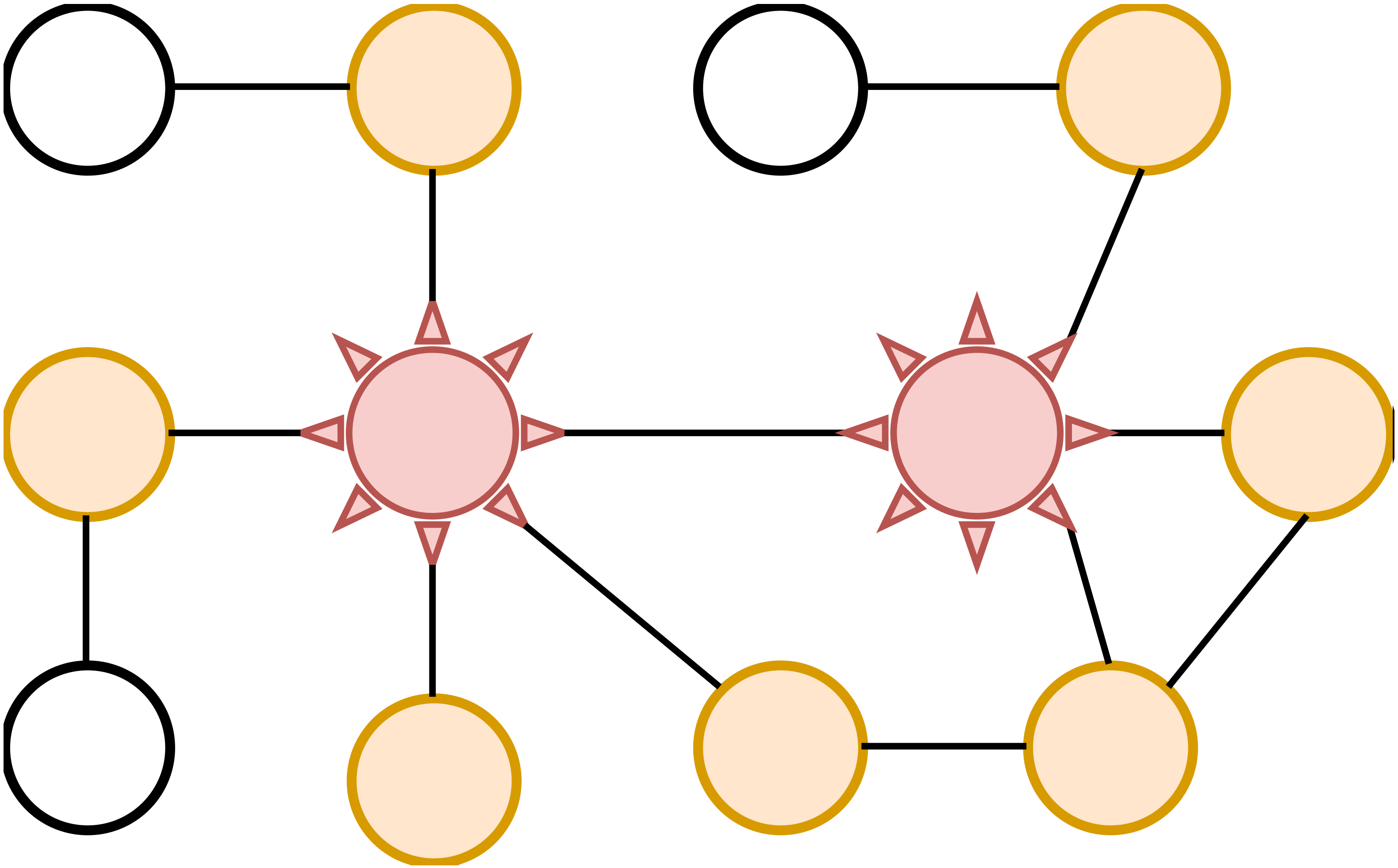
New algorithm?

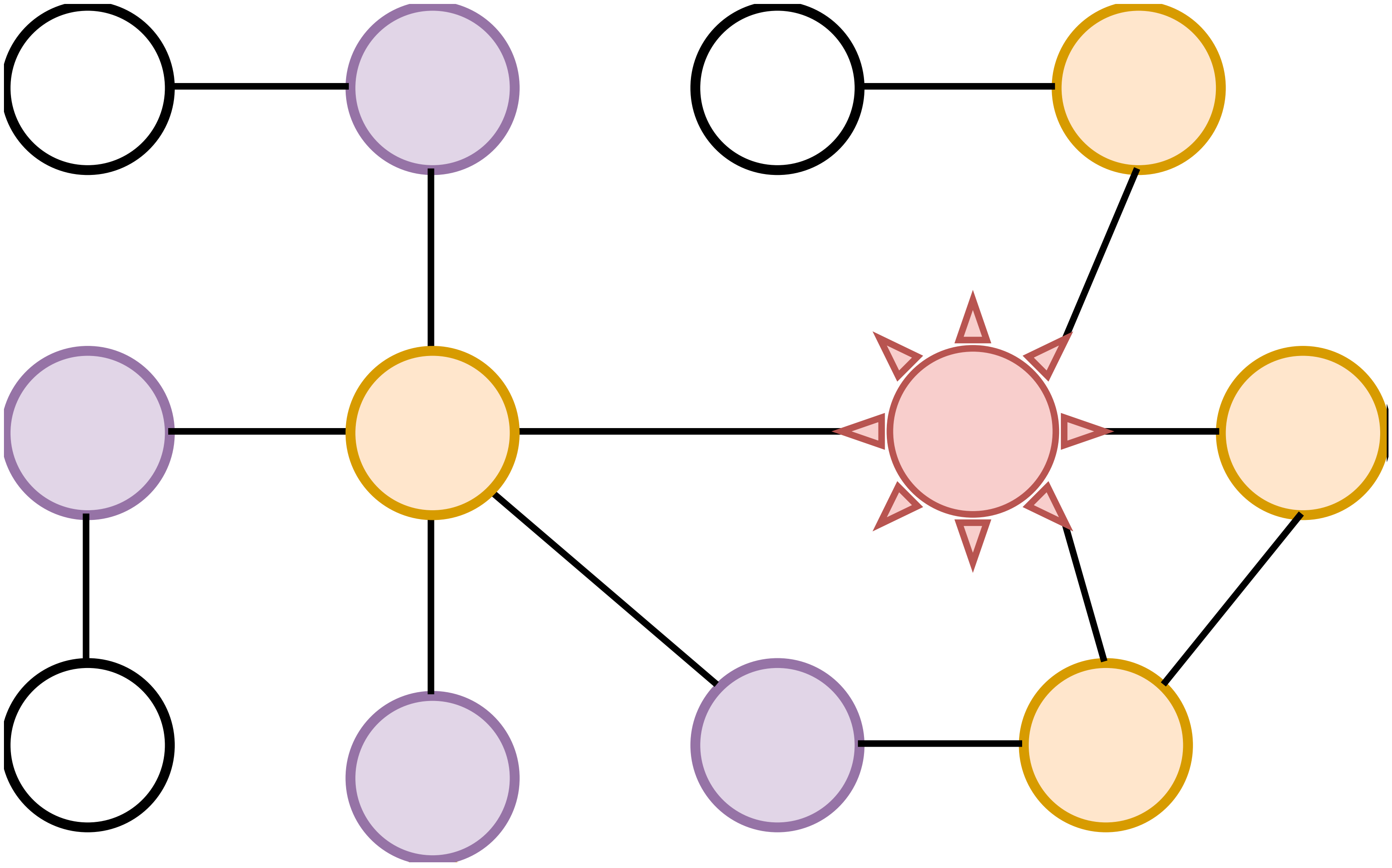
- so far we only ported the C to CUDA C
- only active nodes and their neighbors can change their state
- keep track of these nodes and calculate new states only for these nodes

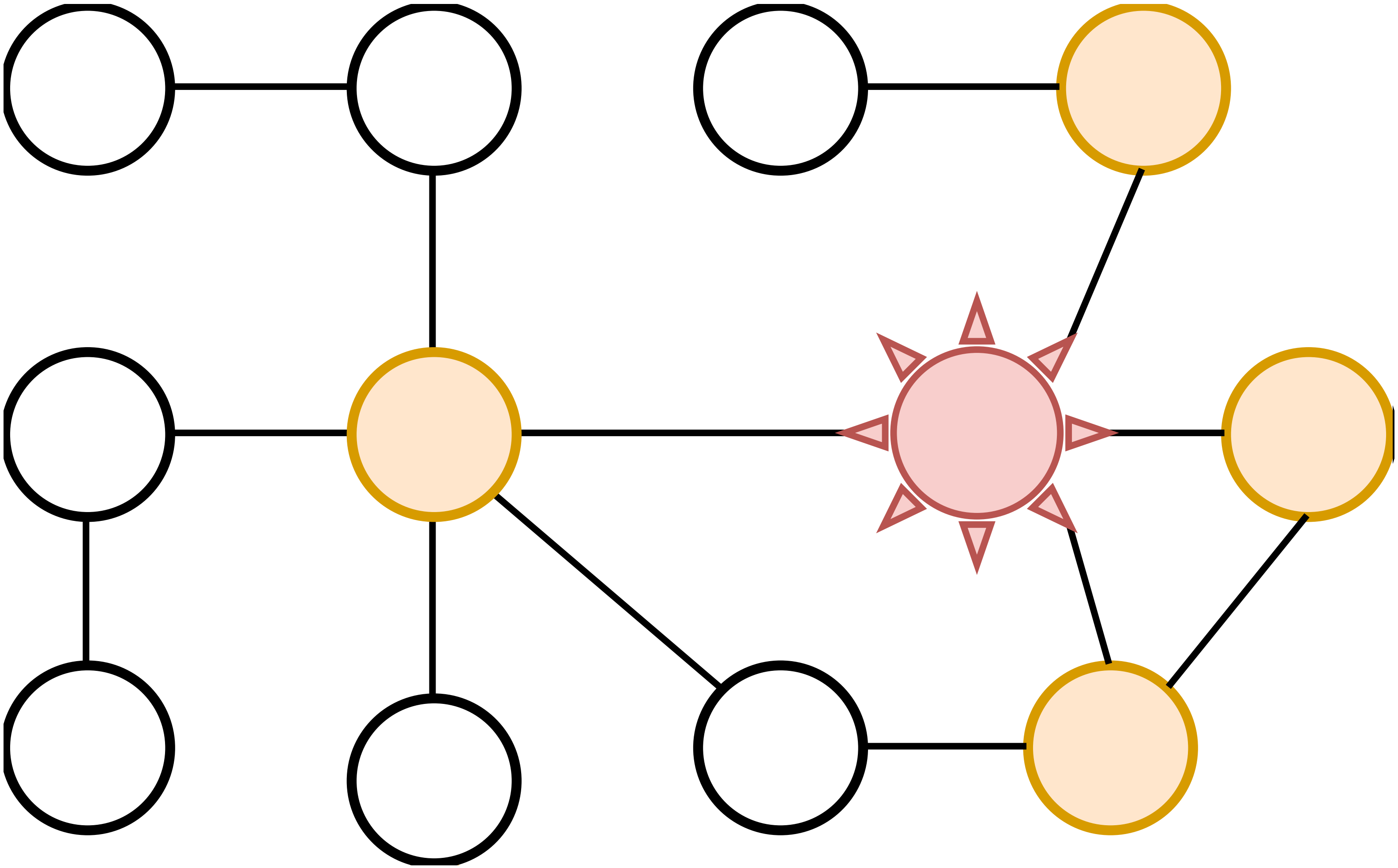












Plans

New algorithm?

- only active nodes and their neighbors can change their state
- keep track of these nodes and calculate new states only for these nodes
- complicates program, need to implement all supporting data structures and algorithm
- could improve performance significantly for cases where active node count is low (switch between algorithms depending on active node count???)
- same algorithm written for CPU might perform better?

Run simulations on more networks

- Automate downloading from Open Connectome Project and transform the network data files
- Run simulations automatically on downloaded data sets
- Analyse results, look for patterns, find connection between network properties and simulation results

Let me know if you'd like to try this program for your spreading simulations.

Questions?

Any advice?

Books, articles

Hagmann P. PhD thesis. STI School of Engineering, University of Lausanne; 2005. From diffusion MRI to brain connectomics

Sporns O, Tononi G, Kotter R. The Human Connectome: A Structural Description of the Human Brain. PLoS Comput Biol. 2005;1(4):e42. doi: 10.1371/journal.pcbi.0010042

Lichtman JW, Sanes JR. Ome sweet ome: what can the genome tell us about the connectome? Curr Opin Neurobiol. 2009;18(3):346–353. doi: 10.1016/j.conb.2008.08.010

Beggs, John M., and Nicholas Timme. "Being critical of criticality in the brain." Frontiers in physiology 3 (2012): 163.

Wolfram, Stephen. "Universality and complexity in cellular automata." Physica D: Nonlinear Phenomena 10.1-2 (1984): 1-35.

Vogelstein JT. Q&A: What is the Open Connectome Project? Neural Systems & Circuits. 2011;1:16. doi:10.1186/2042-1001-1-16.

NeuroData Mission statement

Images

C Elegans http://www.socmucimm.org/overview-model-organism-c-elegans/

MRI http://www.extremetech.com/wp-content/uploads/2013/10/scan.jpg

SBFSEM http://brainmaps.org/index.php?p=speciesdata&species=mus-musculus