Physics and phase transitions in parallel computational complexity

Jon Machta

University of Massachusetts Amherst

and

Santa Fe Institute

Physics of Algorithms
August 31, 2009

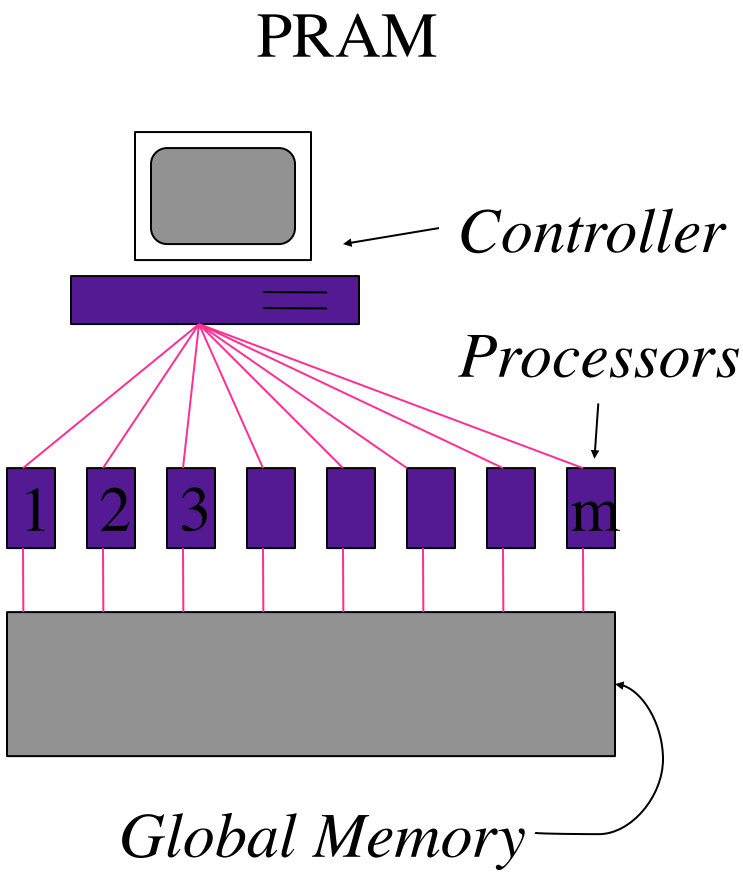
Collaborators

- Ray Greenlaw, *Armstrong Atlantic University*
- Cris Moore, *University of New Mexico, SFI*
- Stephan Mertens, *Otto-von-Guericke University Magdeburg, SFI*
- Students:
  - Ken Moriarty
  - Xuenan Li
  - Ben Machta
  - Dan Tillberg

Outline

- Parallel computing and computational complexity
- Parallel complexity of models in statistical physical
- Random circuit value problem: complexity of solving and sampling

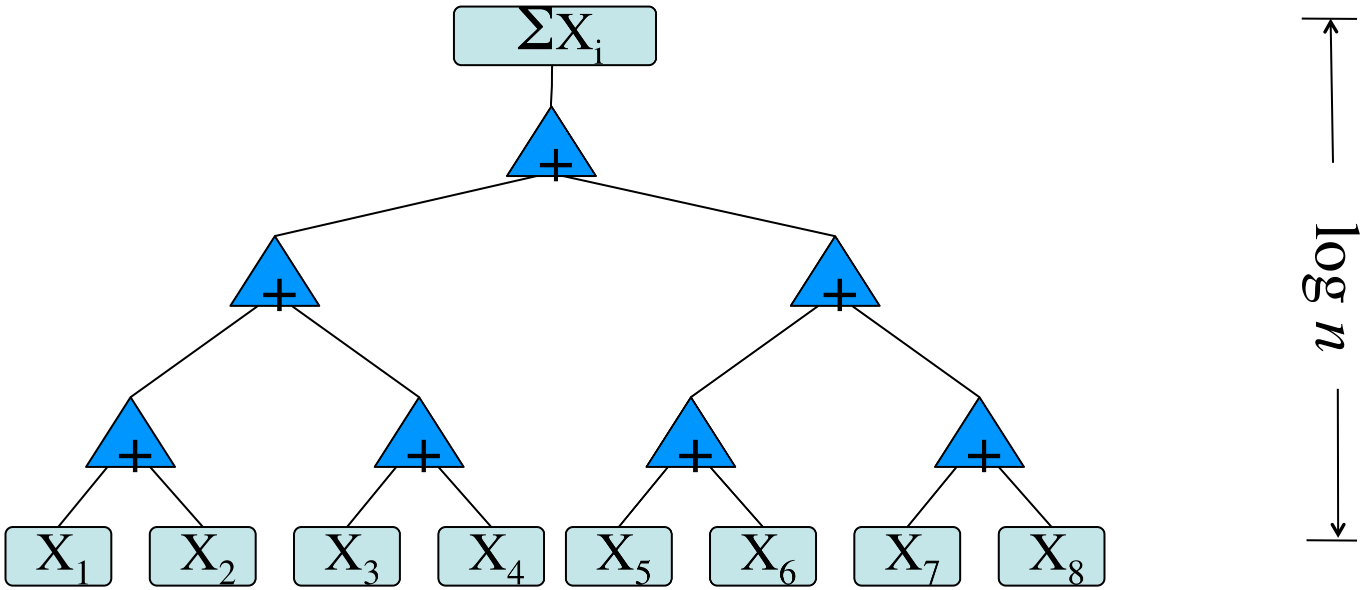
Parallel Random Access Machine



- Each processor runs the same program but has a distinct label
- Each processor communicates with any memory cell in a single time step.
- Primary resources:
  - *Parallel time*
  - *Number of processors*

Parallel Computing

Adding n numbers can be carried out in $O(\log n)$ steps using $O(n)$ processors.



Connected components of a graph can be found in $O(\log^2 n)$ steps using n^2 processors.

Complexity Classes and P-completeness

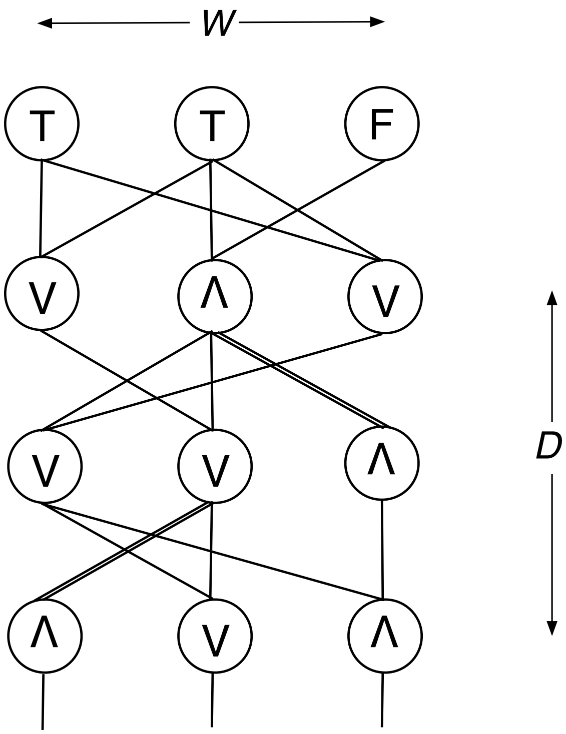
•**P** is the class of *feasible* problems: solvable with polynomial work.

•**NC** is the class of problems efficiently solved in parallel (polylog time and polynomial work, **NC** ⊆ **P**).

•Are there feasible problems that cannot be solved efficiently in parallel (**P**≠**NC**)?

•**P**-complete problems are the hardest problems in **P** to solve in parallel. It is believed they are *inherently sequential:* not solvable in polylog time.

•The Circuit Value Problem is **P**-complete.



Complexity Classes and P-completeness

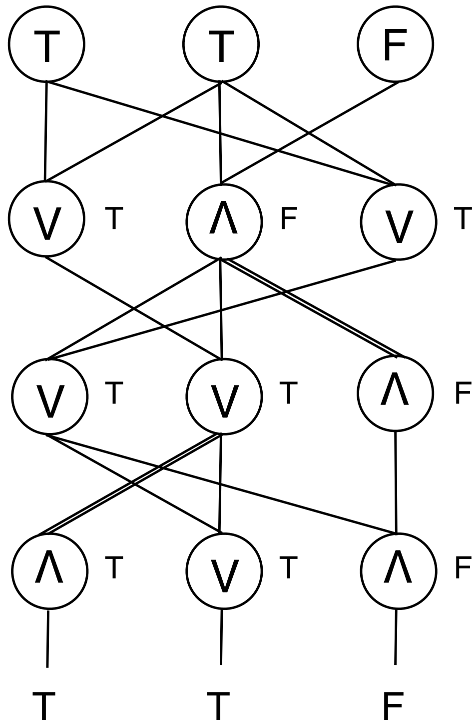
•**P** is the class of *feasible* problems: solvable with polynomial work.

•**NC** is the class of problems efficiently solved in parallel (polylog time and polynomial work, **NC** $\subseteq$ **P**).

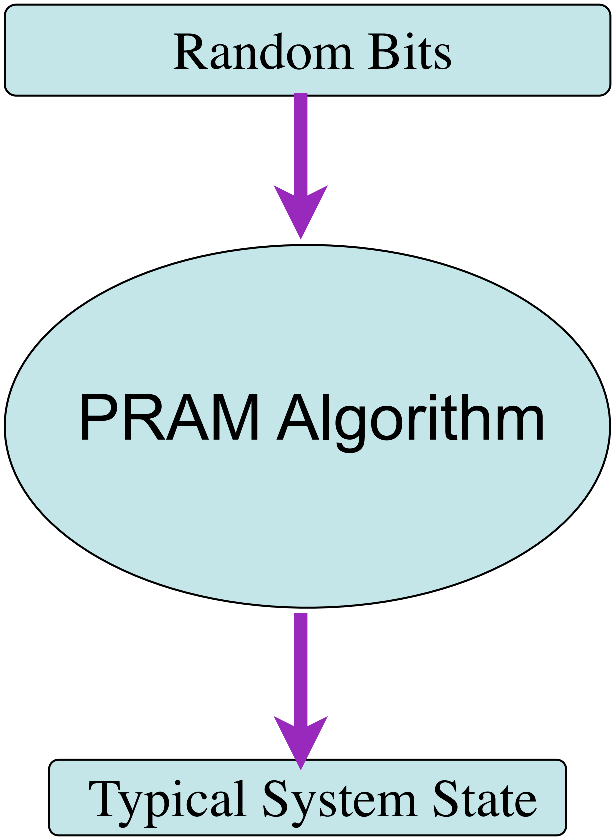
•Are there feasible problems that cannot be solved efficiently in parallel (**P**≠**NC**)?

•**P**-complete problems are the hardest problems in **P** to solve in parallel. It is believed they are *inherently sequential:* not solvable in polylog time.

•The Circuit Value Problem is **P**-complete.



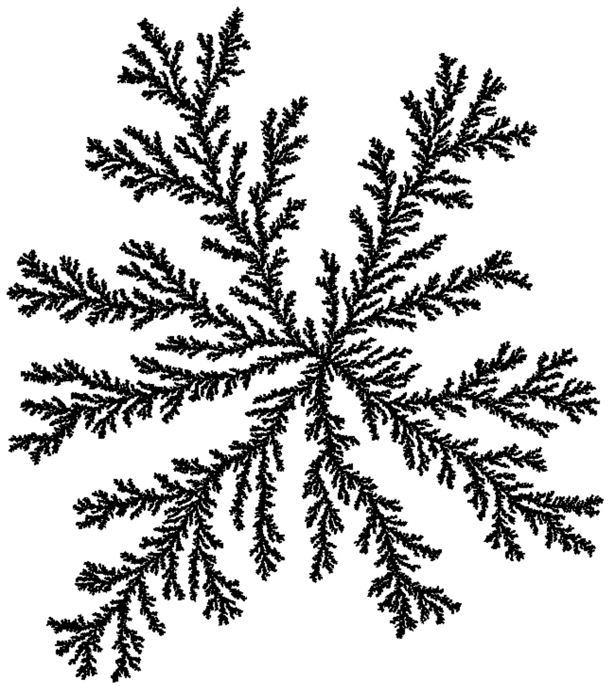
Sampling Complexity



- Models and algorithms in statistical physics convert random bits into typical system states.

Diffusion Limited Aggregation

Witten and Sander, *PRL* 47, 1400 (1981)



•Particles added *one at a time* with sticking probabilities given by the solution of Laplace's equation.

•Self-organized fractal object

d_f=1.715… (2D)

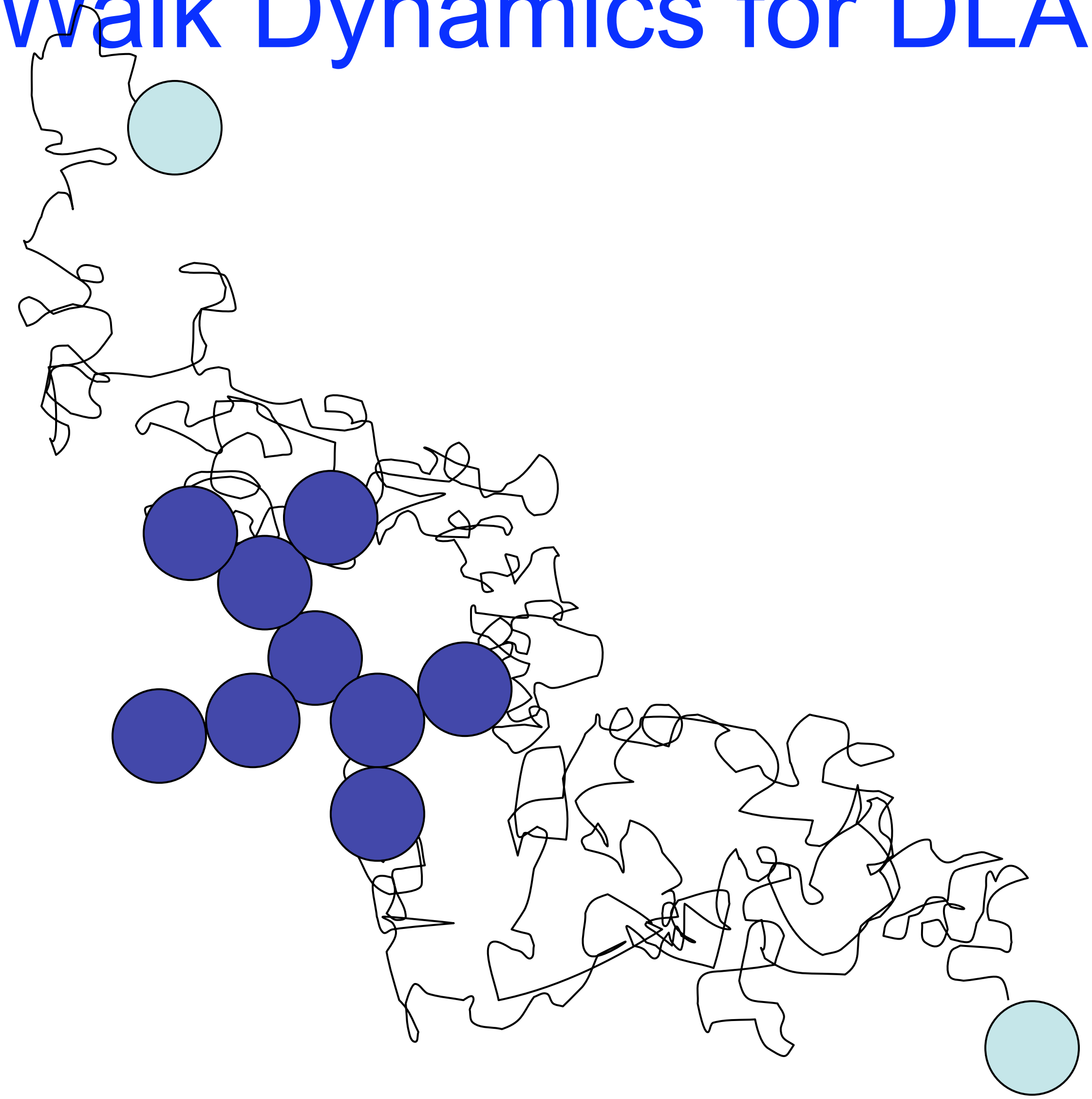
•Physical systems:

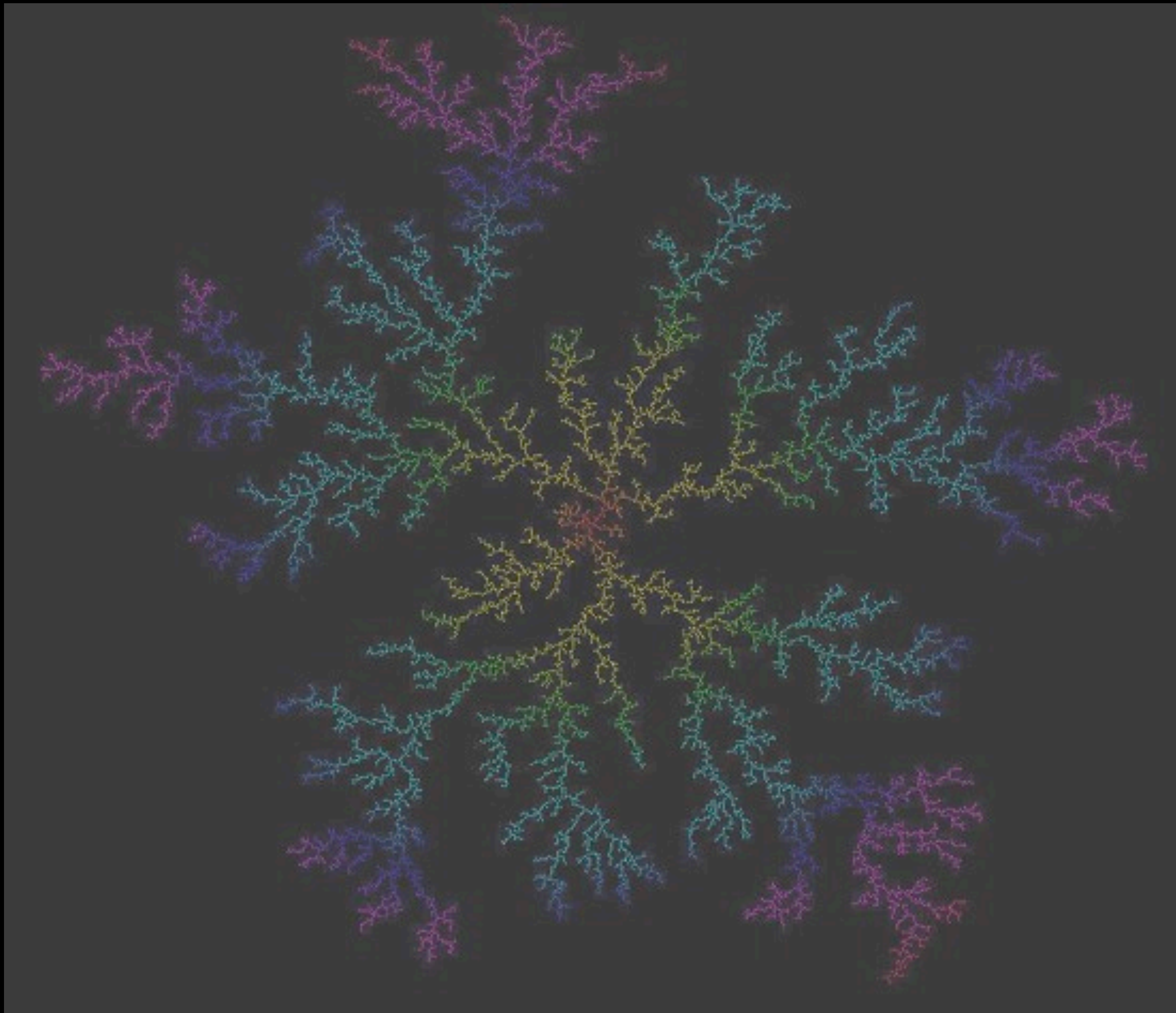
Fluid flow in porous media

Electrodeposition

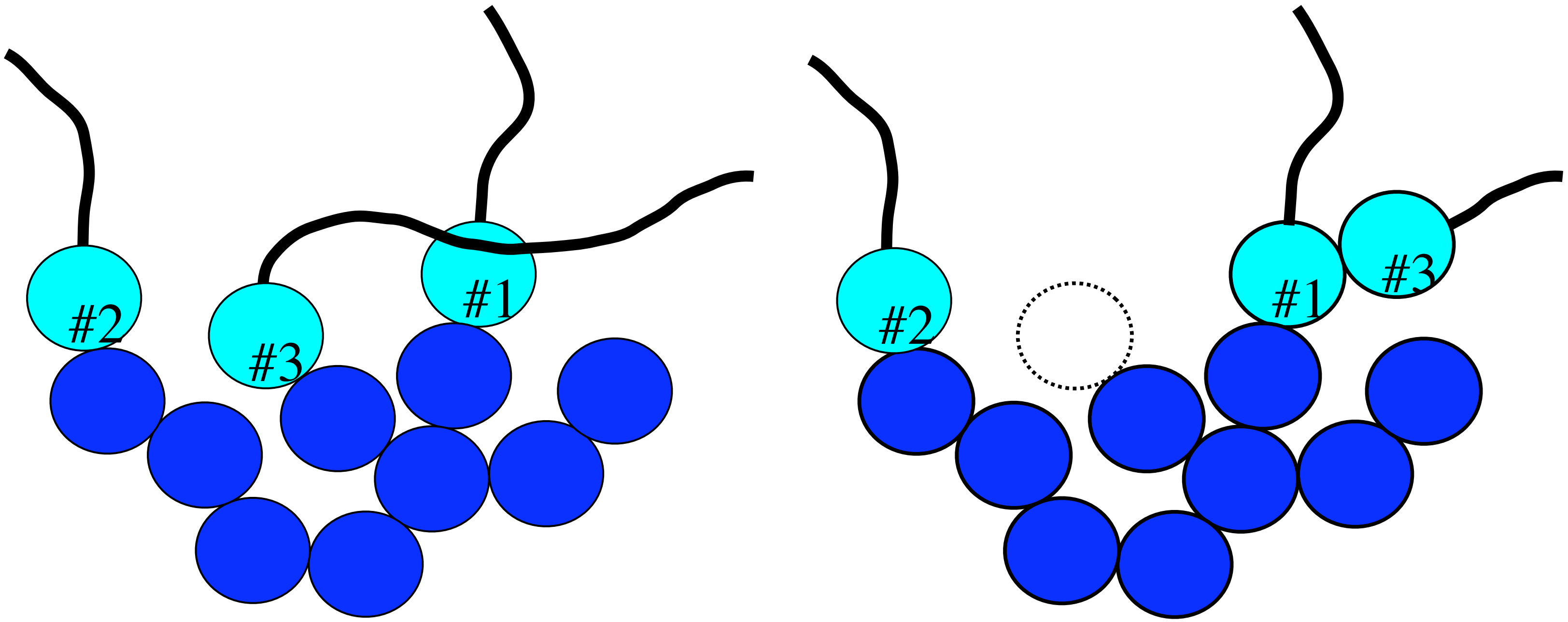
Bacterial colonies

Random Walk Dynamics for DLA





The Problem with Parallelizing DLA



Parallel dynamics ignores *interference* between 1 and 3

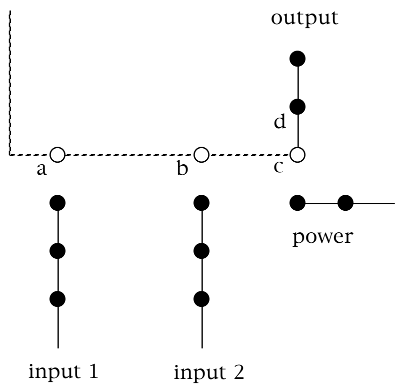
Sequential dynamics

Complexity of DLA

Theorem: Determining the shape of an aggregate from the random walks of the constituent particles is a **P**-hard problem.

Proof idea: Reduce the Circuit Value Problem to DLA dynamics.

Gadet for NOR gate



Caveats:

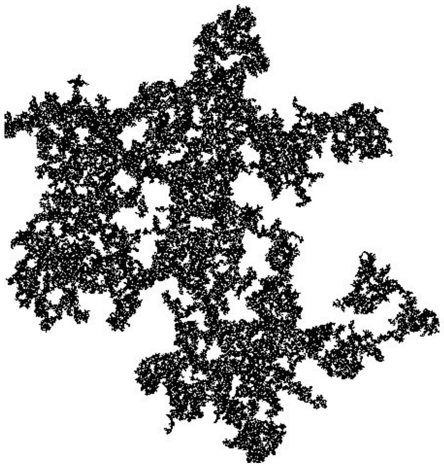
1. **P≠NC** not proven
2. Average case may be easier than worst case
3. Alternative dynamics may be faster than random walk dynamics for sampling DLA

Sequential models with polylog parallel complexity

- Eden growth
- Invasion percolation
- Scale free networks
- Ballistic deposition
- Bak-Sneppen model
- Internal DLA

Eden growth

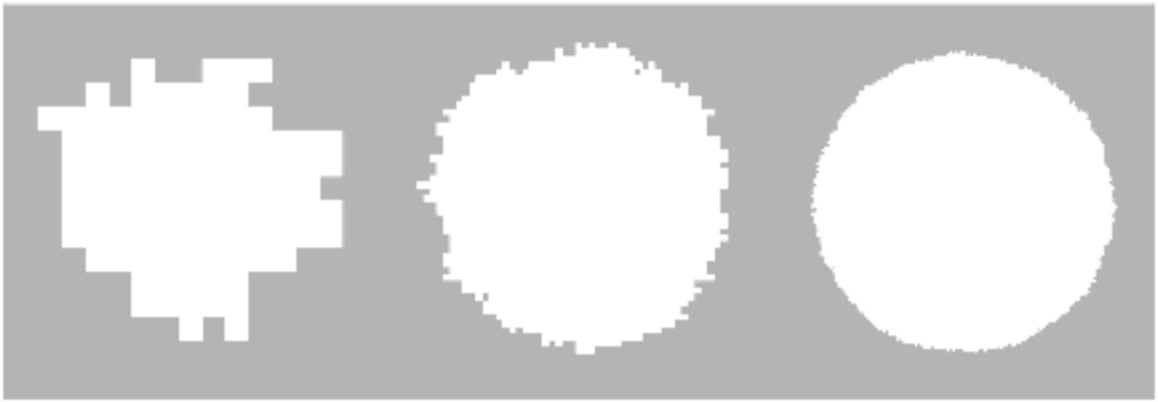
Scale free network



Invasion percolation

Internal DLA

Particles start at the origin, random walk and stick where they first leaves the cluster.



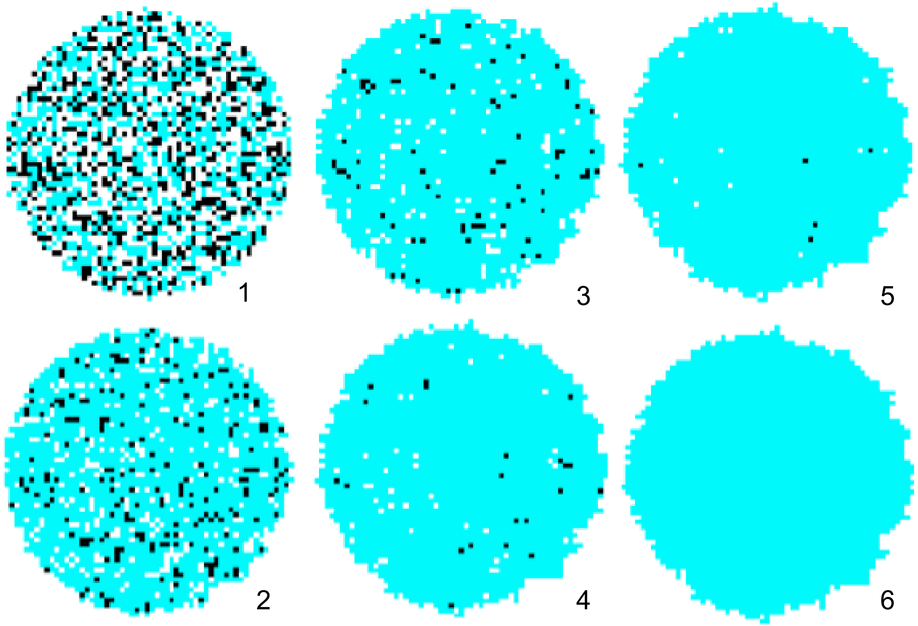
- Shape approaches a circle with logarithmic fluctuations.
- P-completeness proof fails. (However, IDLA is CC-complete)

Parallel Algorithm for IDLA

C. Moore and JM, *J. Stat. Phys.* **99**, 661 (2000)

1. Start with seed particle at the origin and N walk trajectories
2. Place particles at expected positions along their trajectories.
3. Iteratively move particles until holes and multiple occupancies are eliminated

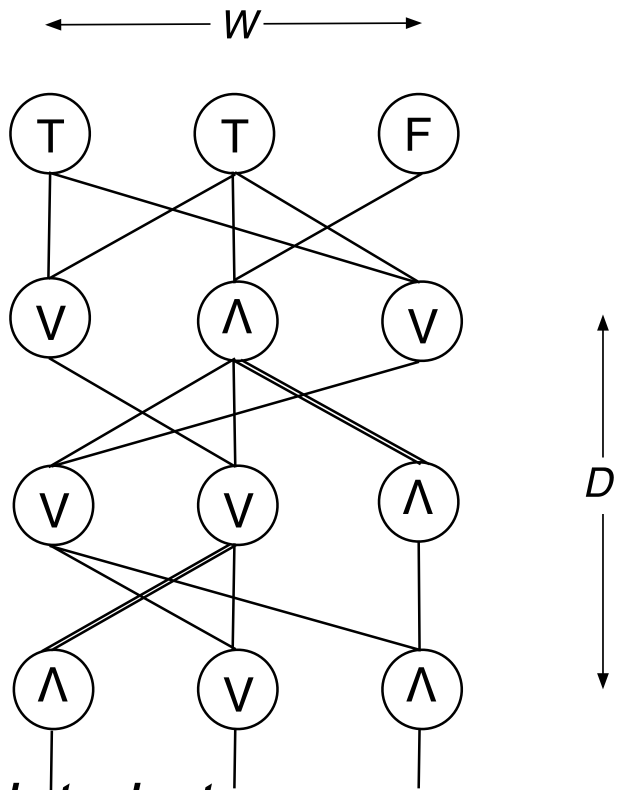
Average parallel time polylogarthmic or possibly a small power in N.



Cluster of 2500 particles made in 6 parallel steps.

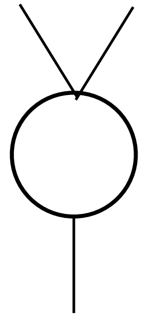
Random Monotone CVP

- Circuit arranged in levels with *W* gates on a level and *D* levels.
- τ_0 =fraction of TRUE inputs.
- *p* =fraction of OR gates.
- Gates at level *n+1* randomly take *k* inputs from gates at level *n* (with replacement).



Monotone CVP is P-complete but how hard is it on average to evaluate the circuit in parallel?

Recursion relations, *k*=2

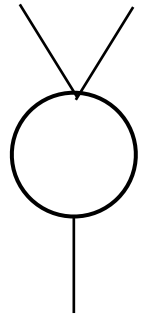


- Let τ_n be the expected fraction of gates evaluating to TRUE at level *n*.

$$\tau_{n+1} = p(1-(1-\tau_n)^2) + (1-p)\tau_n^2$$

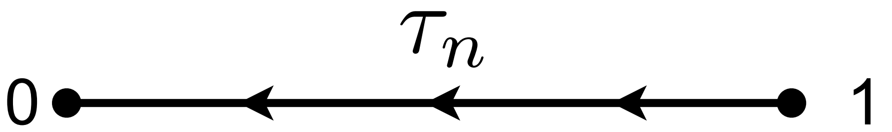
Absorbing fixed points at $\tau = 0$ and $\tau = 1$.

Recursion relations, *k*=2

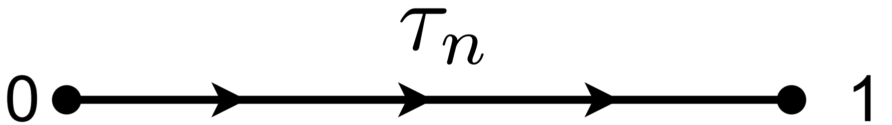


- Let τ_n be the expected fraction of gates evaluating to TRUE at level *n*.

$$\tau_{n+1} = p(1-(1-\tau_n)^2) + (1-p)\tau_n^2$$



$p < 1/2$ *mainly AND*



$p > 1/2$ *mainly OR*

Linearize around fixed points

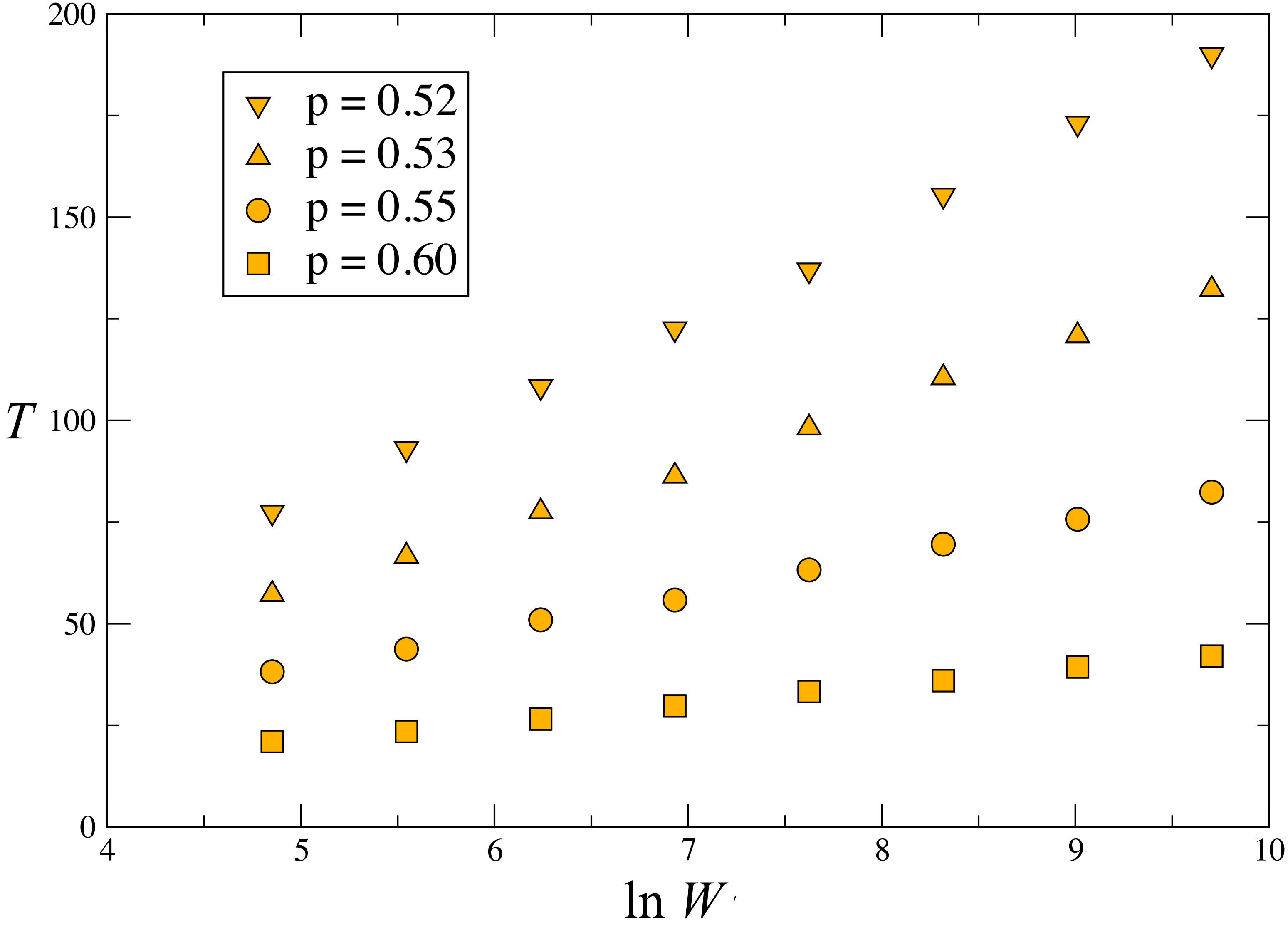
Near the $\tau = 0$ fixed point for $p<1/2$ the linearized recursion relations are:

$$\tau_{n+1} = 2p\tau_n + \mathcal{O}(\tau_n^2)$$

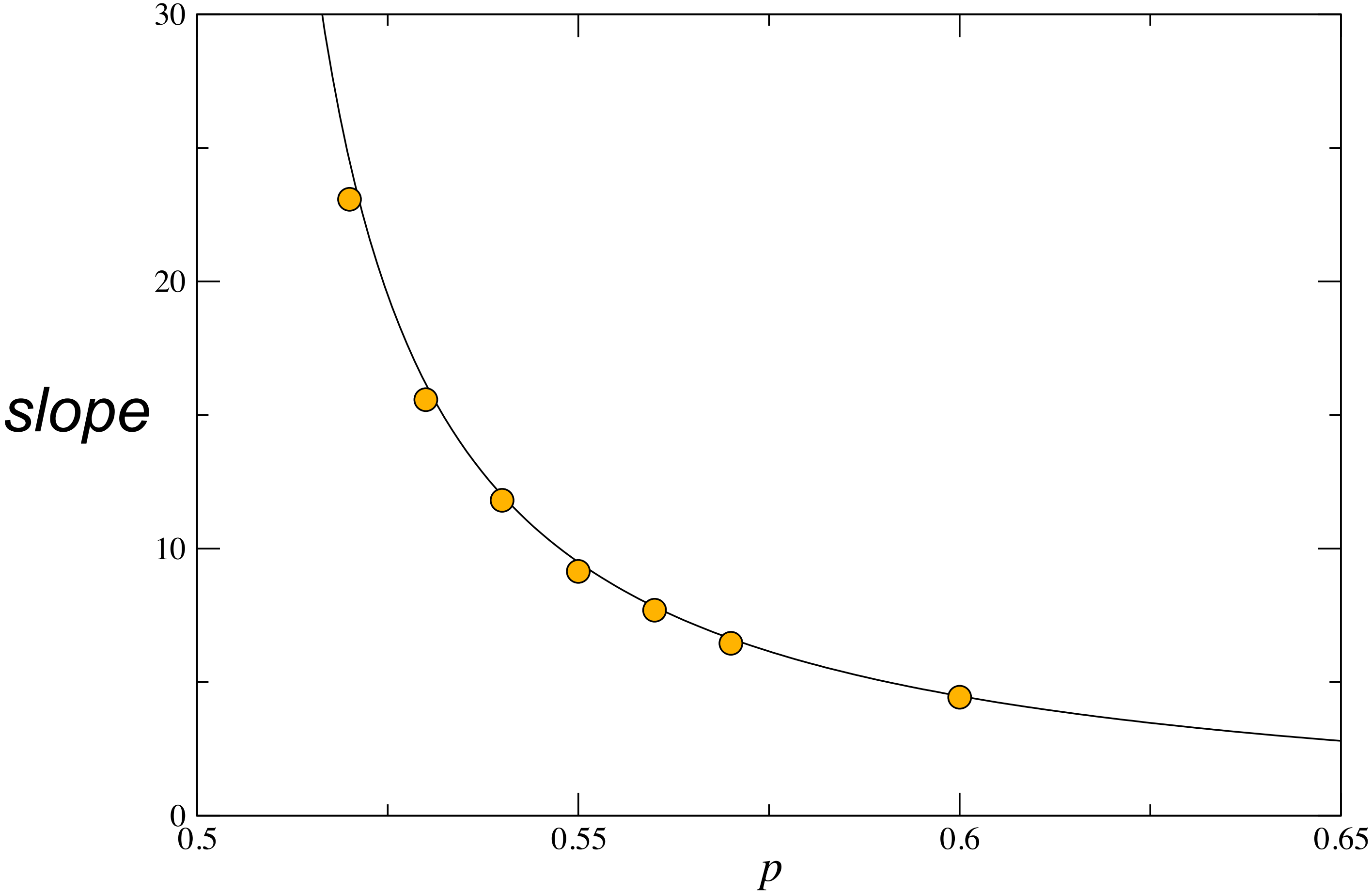
Let T be the time to saturation to all FALSE,

$$\tau_T \approx 1/W$$

$$\boxed{T \sim \frac{\ln W}{-\ln(2p)}}$$



Time to saturation T as a function of circuit width W for various fractions p of OR gates.



Slope of the logarithmic scaling of the saturation time vs. p.
The solid line is the prediction, $-1/\ln(2(1-p))$.

Critical point at $p=1/2$

The number of gates, X_n evaluating to TRUE at level n obeys a stochastic recursion relation,

$$X_{n+1} = \mathcal{B}(W, X_n/W)$$

Here $\mathcal{B}(n,p)$ is a binomial random variable.

After taking the continuum limit, one obtains a diffusion process with absorbing endpoints and a diffusion coefficient that vanishes at the endpoints.

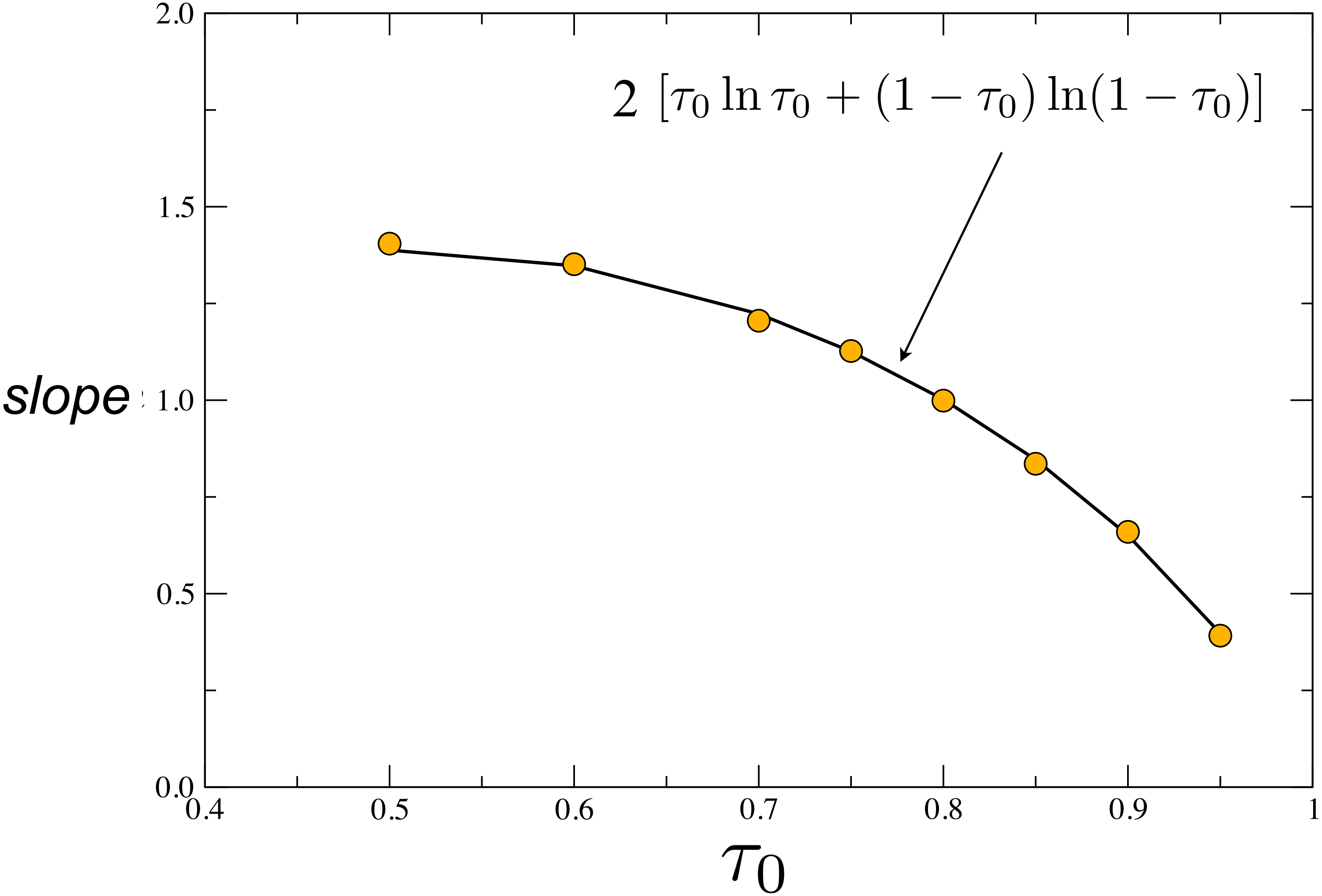
Critical Saturation Time

Using known results for mean first passage times with the spatially non-uniform diffusion coefficient

$$D(x) = \frac{x}{2}(1 - \frac{x}{W})$$

we obtain a linear saturation time:

$$T = -2\underline{\underline{W}} \left[\tau_0 \ln \tau_0 + (1 - \tau_0) \ln(1 - \tau_0)\right]$$



Slope of the linear scaling of the saturation time vs. W.

Summary for two input gates

- For $p \neq 1/2$

  $$T \sim \ln W$$

  Circuit evaluation easy

- For $p = 1/2$

  $$T \sim W$$

  Circuit evaluation hard

$k > 2$

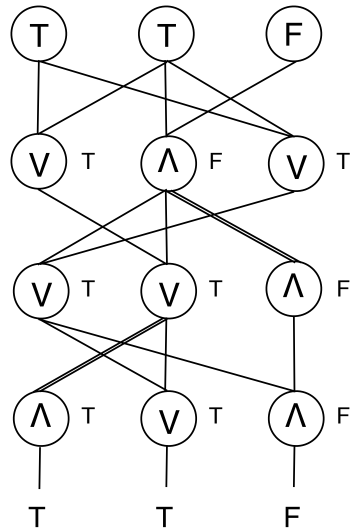
- For $p<1/k$ or $p>1-1/k$ have $T \sim \ln W$ ➜ *Fast circuit evaluation*.

- For $1/k < p < 1-1/k$ have non-trivial fixed point:

$$0 < \tau^* < 1$$

Circuit does not saturate to a single value except via a large deviation ➜ *Slow circuit evaluation*.

Generating Circuit+Solution Pairs



- Q: How difficult is it to simultaneously generate an instance of random monotone CVP together with its evaluation?

- A: For any values of the parameters, a random instance chosen from the correct distribution *and* its evaluation can be generated in polylog parallel time on a PRAM.

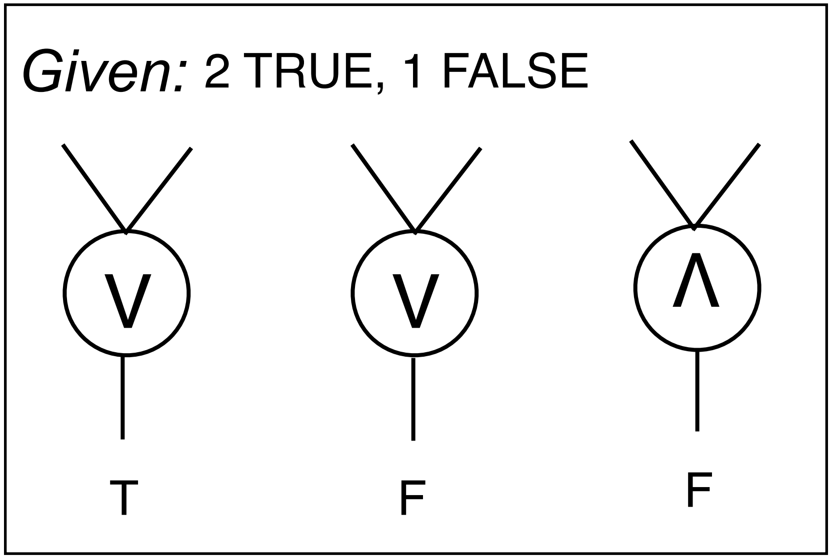
Fast Parallel Sampling of Circuit +Evaluation Pairs

- Idea: In parallel generate an instance of each level--gates and their inputs and outputs--then put the levels together into a complete circuit+evaluation.

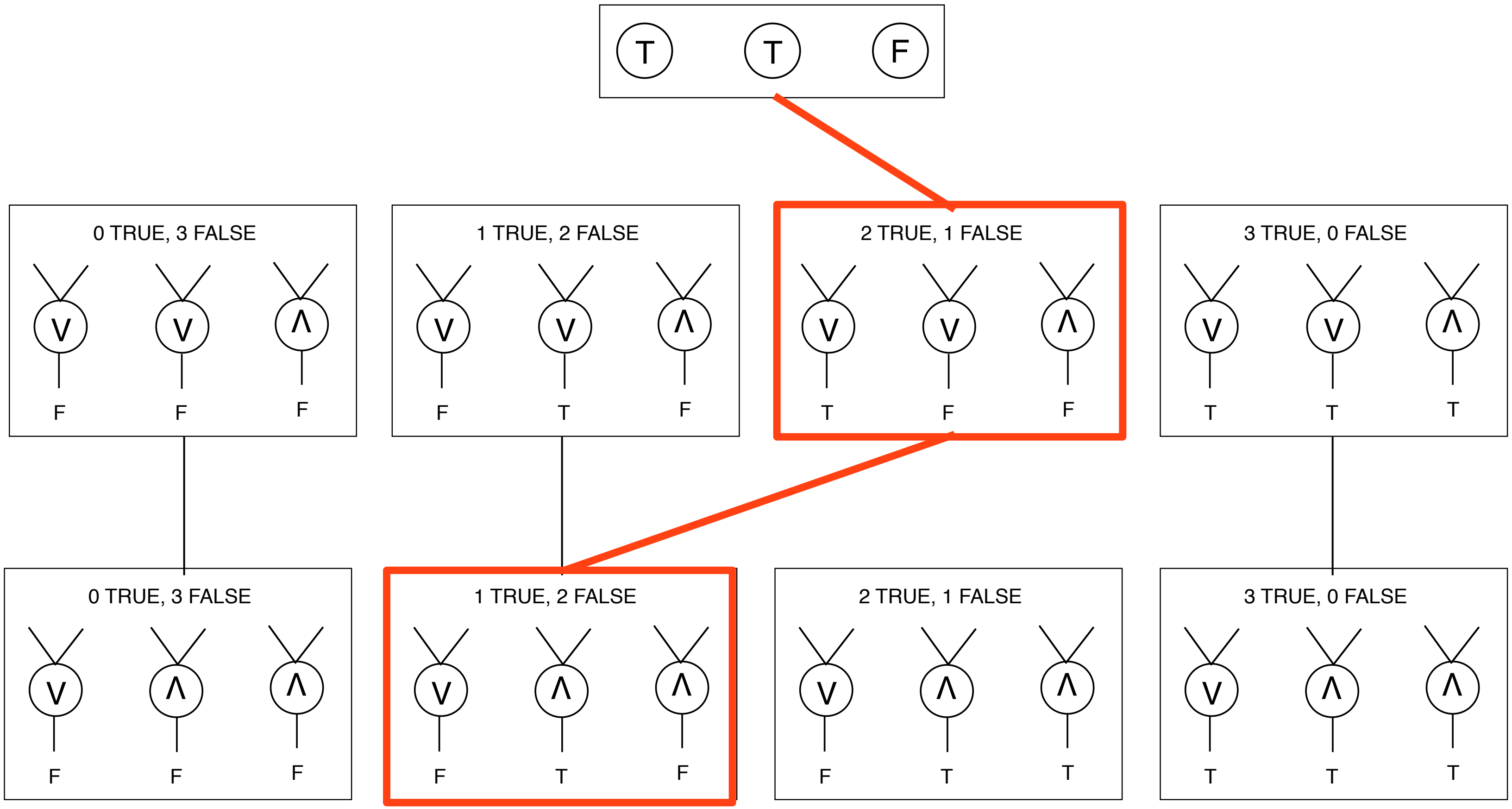
- Difficulty: Inputs to layer $n+1$ are not known until layer n is evaluated.

- Solution: The <u>number</u> of TRUE inputs is all that is required to generate a random level. In parallel construct W+1 instances of each level, one for each number of TRUE inputs.

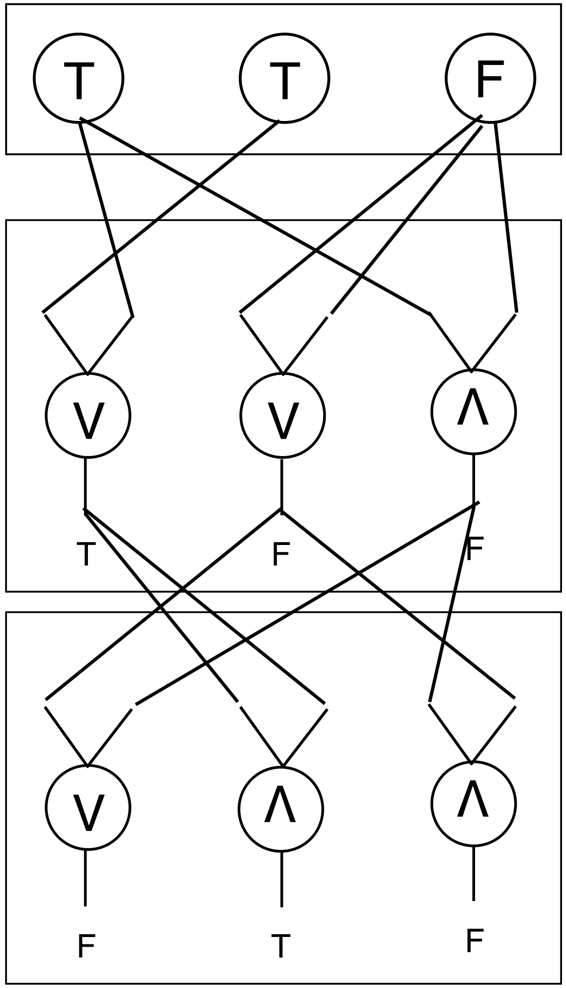
Construct one level



Attaching levels into a circuit+evaluation



Wiring the circuit



Conclusion

- Parallel computational complexity provides a unique perspective on models in statistical physics.
- Simple methods yield interesting results for random ensembles of CVP revealing phase transitions in complexity.
- Although CVP is hard to solve in parallel, it is easy to generate random instances and solutions simultaneously.