Observations in the Sandpile Model

Thomas Worsch

June 14, 2006

The Sandpile Model

Recurrent configurations

Transient configurations

Definition of the CA

- CA on a 2-dim. rectangle with 8 possible states for each cell: $\{0, 1, 2, 3, 4, 5, 6, 7\}$
- Idea: each cell contains a number of "grains of sand" (or "chips")
- Neighborhood: von Neumann, with radius 1
- Local rule:
  - If a cell has at least 4 grains
  - it moves 1 to each of its neighbors (toppling, firing).
  - If a cell at the border fires, 1 or 2 grains are lost.
- stable configuration: all states ≤ 3

Fact:

- Each unstable configuration leads to a stable one after a finite number of steps. Call this a relaxation.

Example

5	4	4	4	4	4	4	5
4	3	3	3	3	3	3	4
4	3	3	3	3	3	3	4
4	3	3	3	3	3	3	4
4	3	3	3	3	3	3	4
4	3	3	3	3	3	3	4
4	3	3	3	3	3	3	4
5	4	4	4	4	4	4	5

Example

3	2	2	2	2	2	2	3
2	5	4	4	4	4	5	2
2	4	3	3	3	3	4	2
2	4	3	3	3	3	4	2
2	4	3	3	3	3	4	2
2	4	3	3	3	3	4	2
2	5	4	4	4	4	5	2
3	2	2	2	2	2	2	3

Example

3	3	3	3	3	3	3	3
3	3	2	2	2	2	3	3
3	2	5	4	4	5	2	3
3	2	4	3	3	4	2	3
3	2	4	3	3	4	2	3
3	2	5	4	4	5	2	3
3	3	2	2	2	2	3	3
3	3	3	3	3	3	3	3

Example

3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	2	2	3	3	3
3	3	2	5	5	2	3	3
3	3	2	5	5	2	3	3
3	3	3	2	2	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3

Example

3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3

start again

Example

3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3

Each cell fired exactly once.

start again

burning algorithm

Question 1

Is this a "simple program"?

Remarks

- It's “simple enough” to be generalized to arbitrary graphs:
  - A node fires if it has at least as many chips as it has links to neighbors.
  - This is called the chip firing game.
- It is a “robust” program:
  - One reaches the same stable configuration, no matter whether synchronous or asynchronous updating is used.
  - (a little bit of care required . . .)

Markov chain

Given a stable configuration

- choose with equal probability one of the cells,
- add one grain of sand to it, and
- relax to the next stable configuration.

The Sandpile Model

Recurrent configurations

Transient configurations

Transient and recurrent configurations

- recurrent configuration: a stable configuration c such that after having added one grain anywhere in c one can always add more grains such that relaxation leads to c again
- transient configuration: a non-recurrent configuration

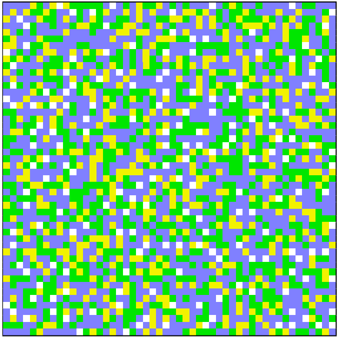
Examples:

- all cells in state 3: recurrent
- all cells in state 0: transient

Some nice results about recurrent configurations

- Linear time check whether a configuration is recurrent: burning algorithm.
- The recurrent configurations form an Abelian group under the operation $\oplus$ of pointwise addition followed by relaxation.
- The number of recurrent configurations equals
  - the number of rooted spanning forests of the graph of nodes and boundary cells
  - the determinant of the Laplacian of that graph
- and more ...

How do recurrent configurations look like?

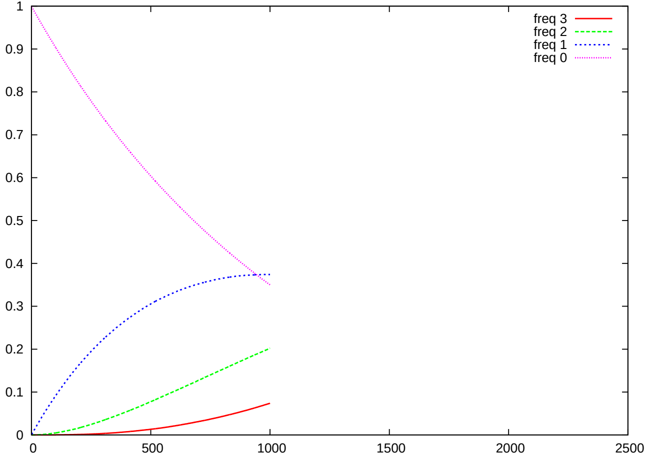


- Does it look random?

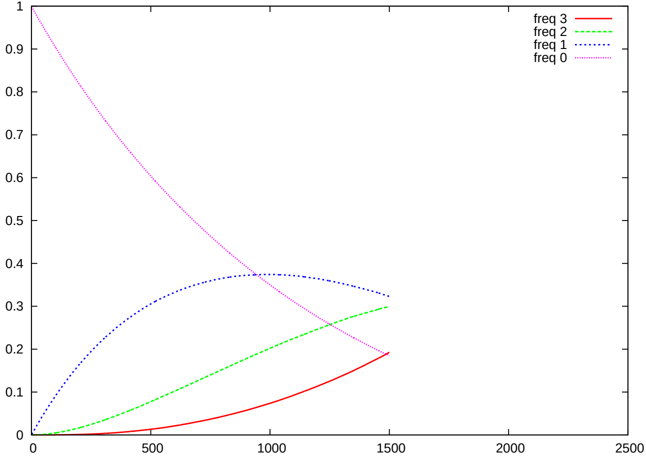
Question 2

What is “random”?

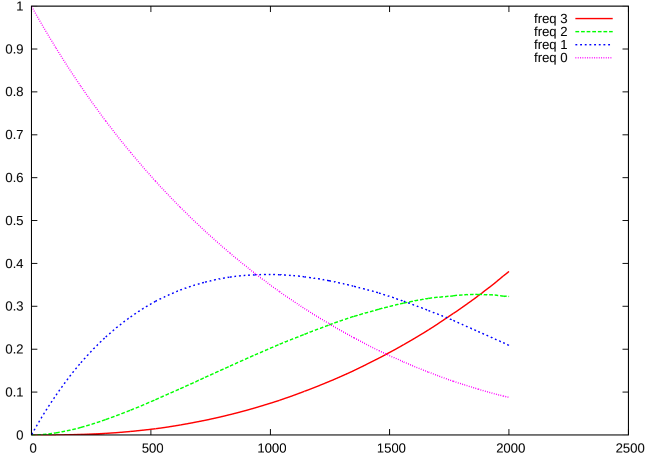
From zero to recurrent



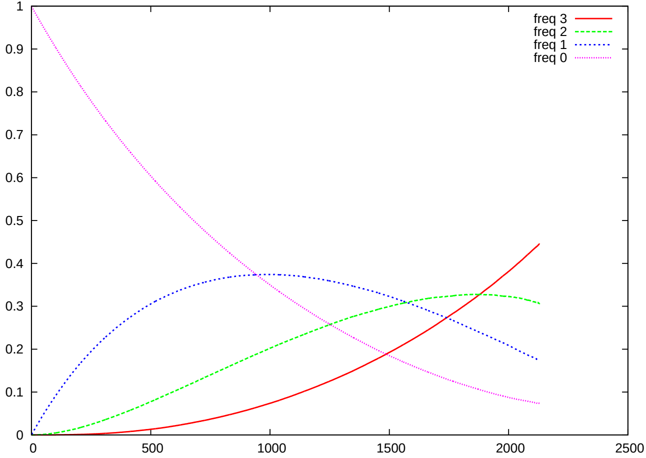
From zero to recurrent



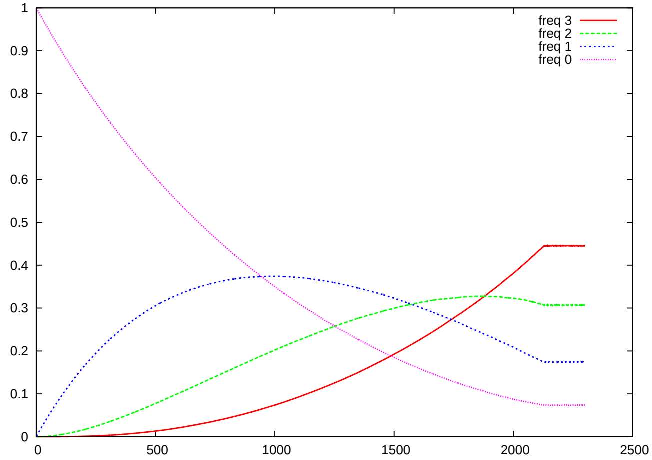
From zero to recurrent



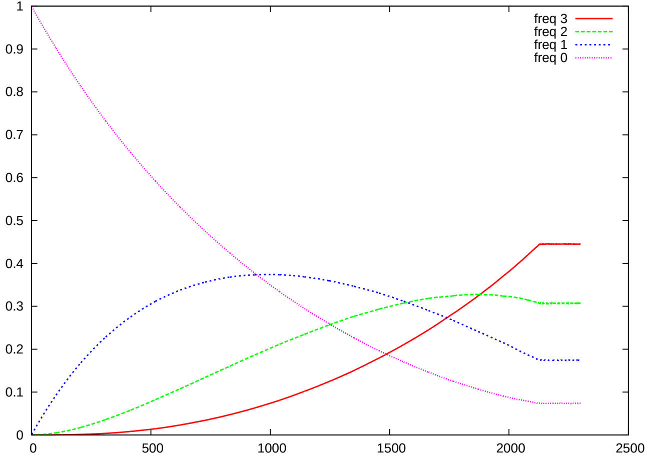
From zero to recurrent



From zero to recurrent

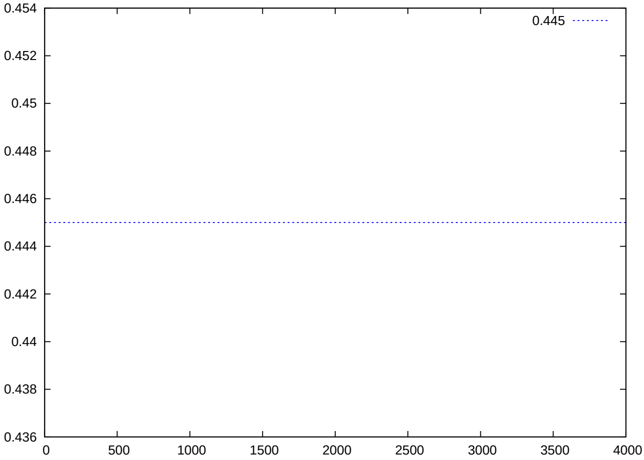


From zero to recurrent

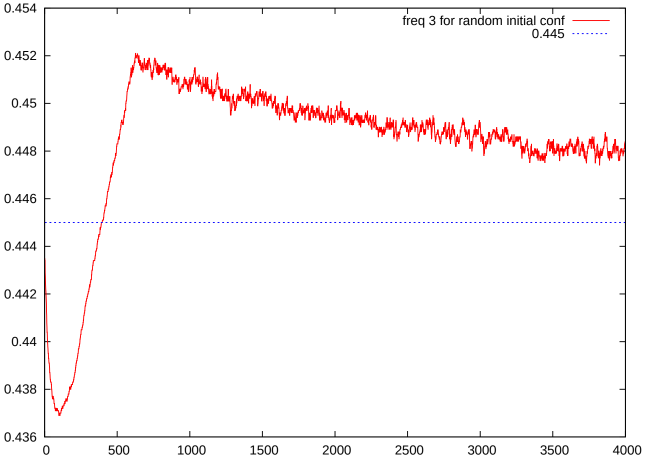


In the end "usually":	0	1	2	3
	7.4%	17.4%	30.7%	44.5%

From random to recurrent



From random to recurrent



The Sandpile Model

Recurrent configurations

Transient configurations

Less is known about the transient configurations.

- Partial orders?
  (such that adding sand always leads to confs. "upward")
- Can one measure a "distance from the set of recurrent configurations"?
  (such that adding sand always decreases the "distance")

skip partial orders

Partial orders on (transient) configurations

- reachability $c \leq d \iff \exists\, e : c \oplus e = d$
- Def.: $\mathit{diff}(c, d) = \mathbf{3} - (c \oplus (\mathbf{3} - d))$
- Fact: $c \leq d \iff c \oplus \mathit{diff}(c, d) = d$
- $c \sqsubseteq d \iff \mathit{diff}(d, c) = \mathbf{0}$

"Distance from recurrent configurations"

Consider 3-dim. case with 32 2-dim. layers:

- Problem instance: a transient configuration for that case
- Question: What is the minimum number of grains you have to add in order to reach a recurrent configuration?
- Theorem (M. Schulz, 2006):
  This problem is NP-complete.
- Proof: by reduction from 3SAT.

Construction 1

One layer with columns for setting each variable:

	...						...							...							...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...			4	5		...				4	5		...				4	5		...			
	...			5	5		...				5	5		...				5	5		...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...			5	5		...				5	5		...				5	5		...			
	...			4	5		...				4	5		...				4	5		...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...			5	5		...				5	5		...				5	5		...			
	...	0	0	5	5		...		0	0	5	5		...		0	0	5	5		...			
	...			5	5		...				5	5		...				5	5		...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...			4	5		...				4	5		...				4	5		...			
	...			5	5		...				5	5		...				5	5		...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...			5	5		...				5	5		...				5	5		...			
	...			4	5		...				4	5		...				4	5		...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...			5			...				5			...				5			...			
	...						...							...							...			

Construction 2

One layer for each clause, e.g. $x_1 \vee \bar{x}_2 \vee x_3$:

		…					…					…					…						
		…		5			…		5			…		5			…						
		…		5			…					…		5			…						
		…		5			…					…		5			…						
		…		5	5		…					…		5	5		…						
		…		4	5		…					…		4	5		…						
		…		5			…					…		5			…						
		…		5			…					…		5			…						
		…		5			…					…		5			…						
		…		5	5		…					…		5	5		…						
		…		4	5		…					…		4	5		…						
		…		5			…					…		5			…						
		…		5			…					…		5			…						
		…		5			…					…		5			…						
		…		5			…					…		5			…						
	5	…	5	5	5	5	…	5	5	5	5	…	5	5	5	5	…	5	5	5	0	0	
		…					…		5			…					…		5	5			
		…					…		5			…					…						
		…					…		5			…					…						
		…					…		5			…					…						
		…					…		4	5		…					…						
		…					…		5	5		…					…						
		…					…		5			…					…						
		…					…		5			…					…						
		…					…		5			…					…						
		…					…		4	5		…					…						
		…					…		5	5		…					…						
		…					…		5			…					…						
		…					…		5			…					…						
		…		5			…		5			…		5			…						
		…					…					…					…						

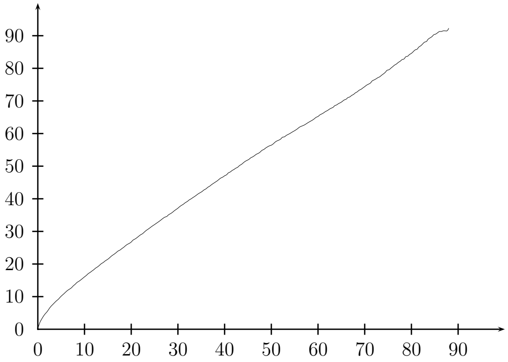
Question 3

Is the problem NP-complete for 2-dimensional CA?

▸ skip another measure

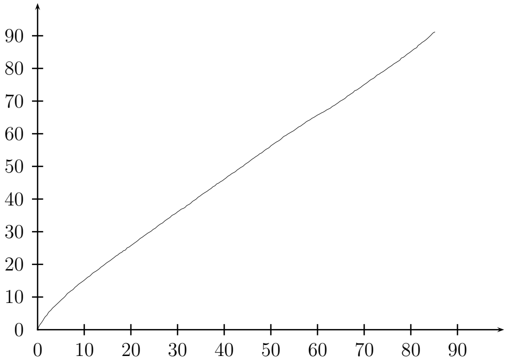
Another measure

- $m(c) = grains(\mathbf{id}) - grains(diff(c, c \oplus \mathbf{id}))$



Another measure

- $m(c) = \mathit{grains}(\mathbf{id}) - \mathit{grains}(\mathit{diff}(c, c \oplus \mathbf{id}))$



Outlook

- M. Schulz is now looking for CA with only 2 states showing “similar” behavior (at least for asynchronous updating).

Thank you very much for your attention.