Parking functions and recurrent configurations in the sandpile model.

Robert Cori
Labri, Université Bordeaux 1

Sandpiles in K_{n+1}

- **Configuration** A sequence of non negative integers $u = (u_1, u_2, \ldots, u_n)$
- **Toppling** Occurs if some u_i is not less than n denoted by: $u \to u'$

$$\begin{cases} u'_i = u_i - n \\ u'_j = u_j + 1 \quad \text{if } j \neq i \end{cases}$$

Sandpiles in K_{n+1}

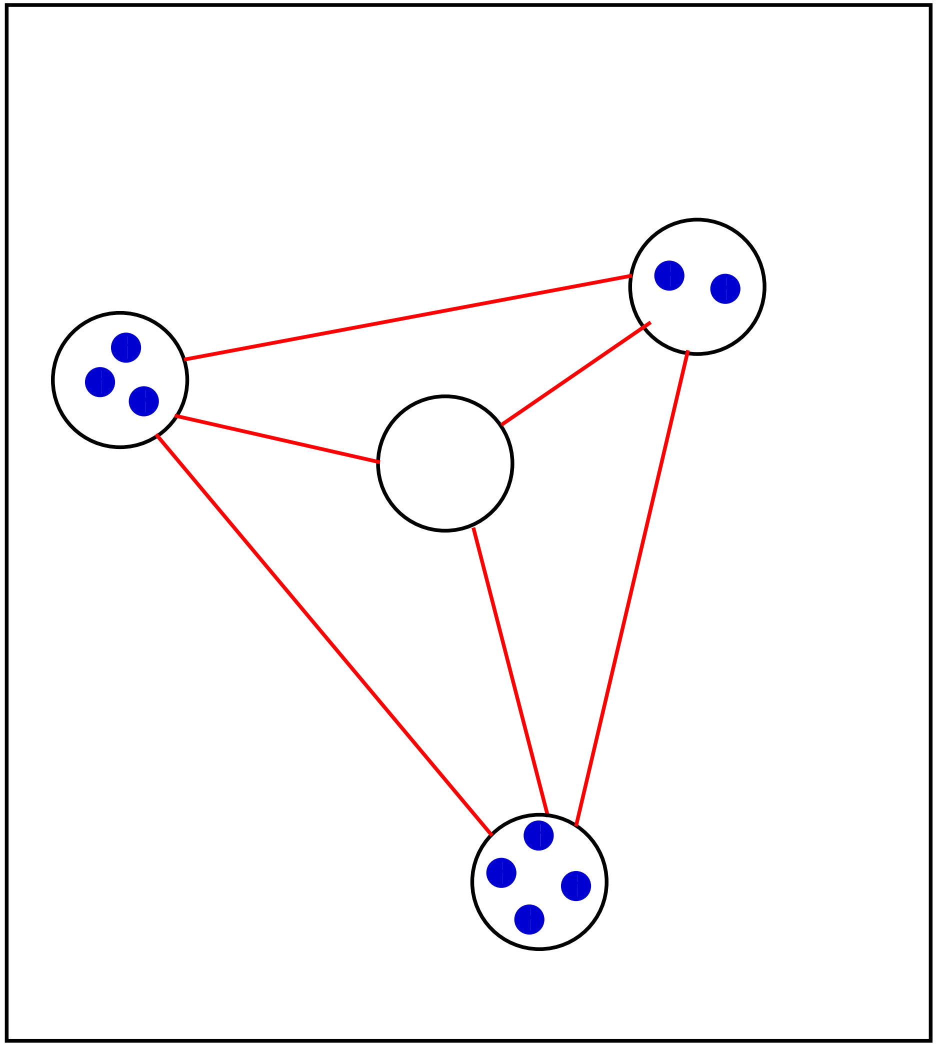
- **Stable configuration** If no toppling is possible, i. e. $\forall i, \quad u_i < n$
- A sequence of topplings is denoted by:

$$u \xrightarrow{*} v$$

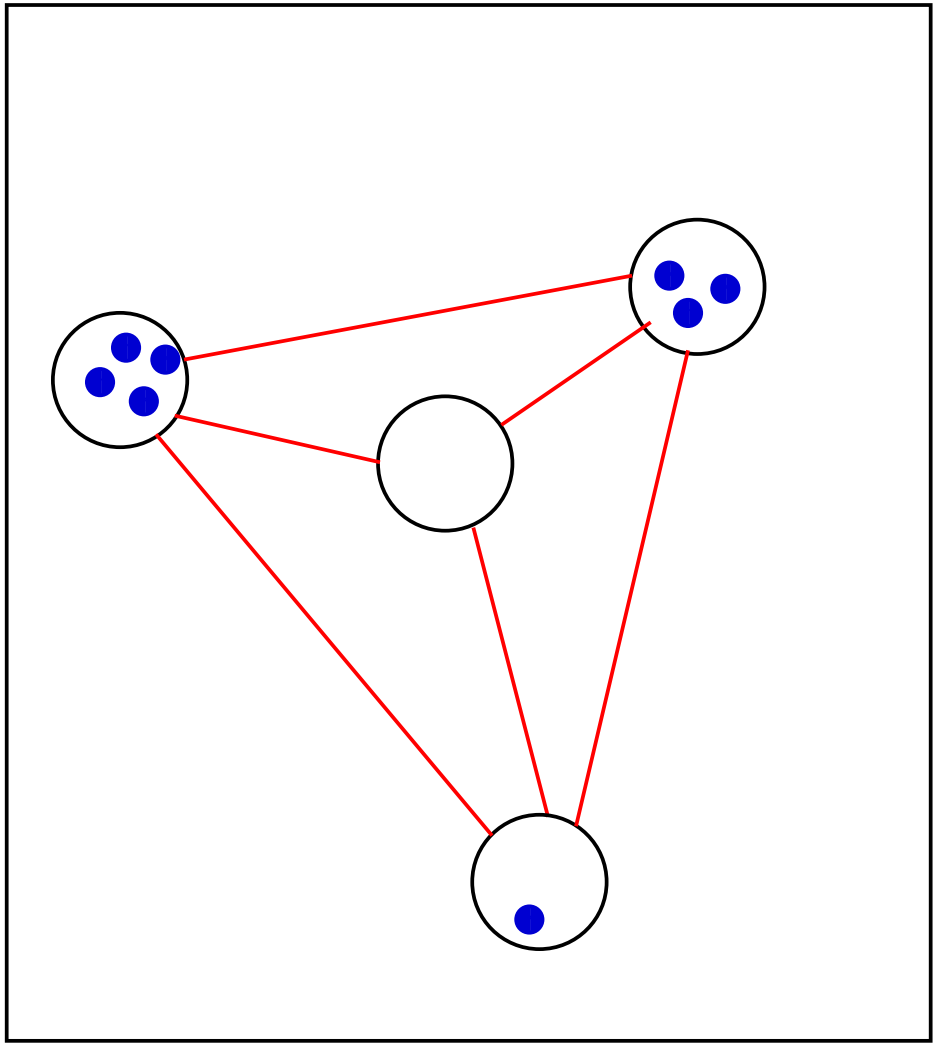
- **Example:**

$$(3,2,4) \rightarrow (4,3,1) \rightarrow (5,0,2) \rightarrow (2,1,3) \rightarrow (3,2,0) \rightarrow (0,3,1) \rightarrow (1,0,2)$$

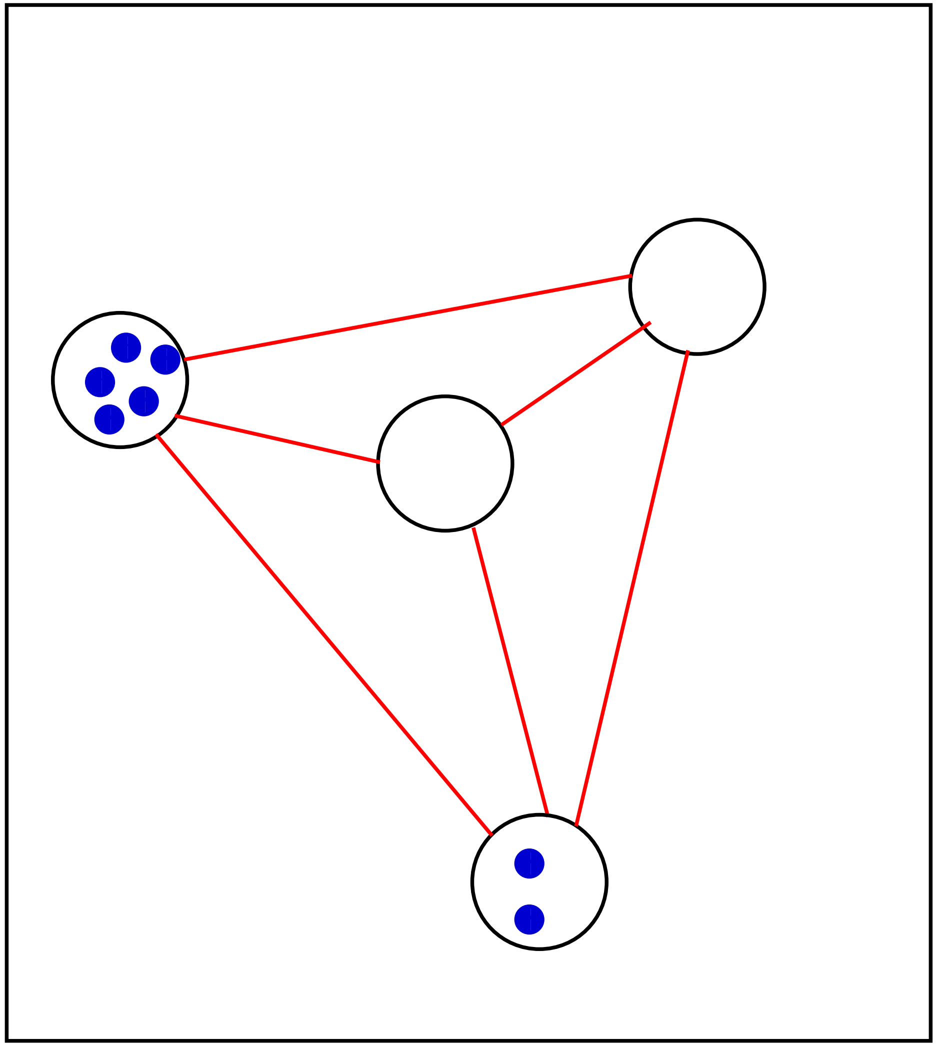
Example



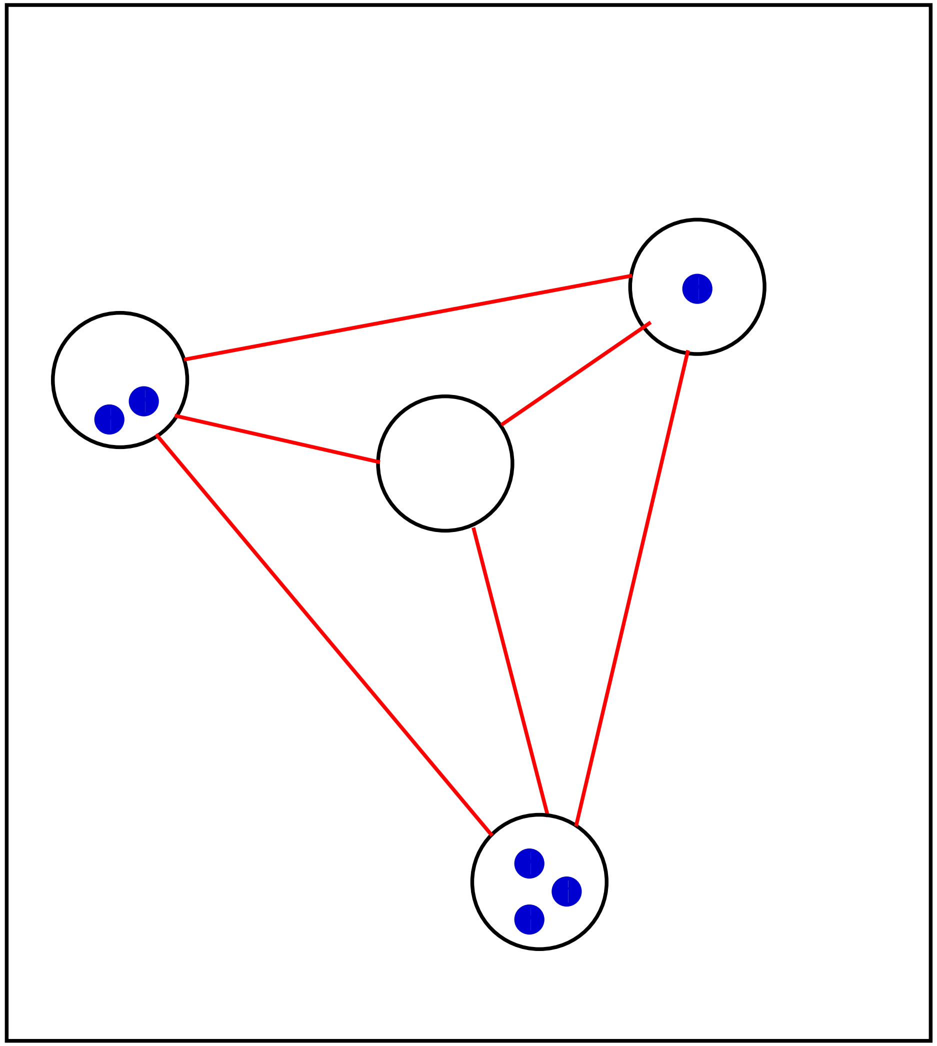
Example



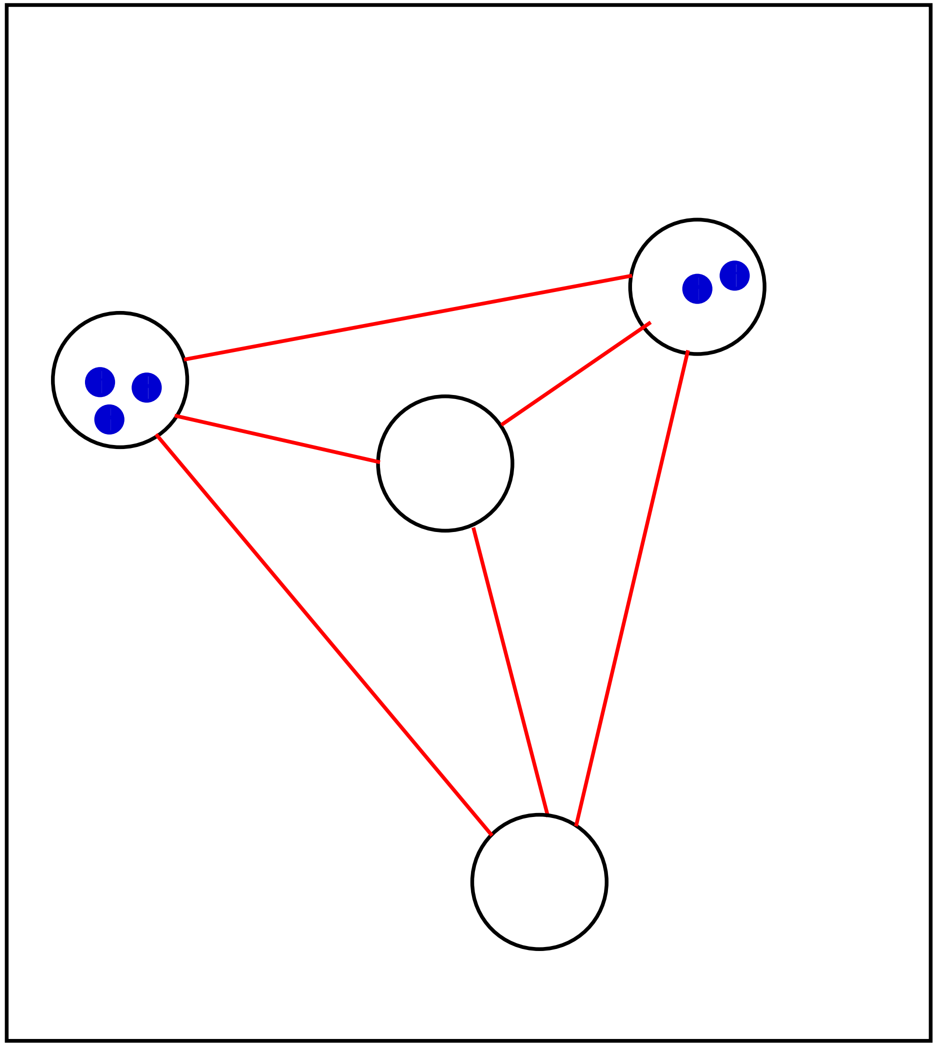
Example



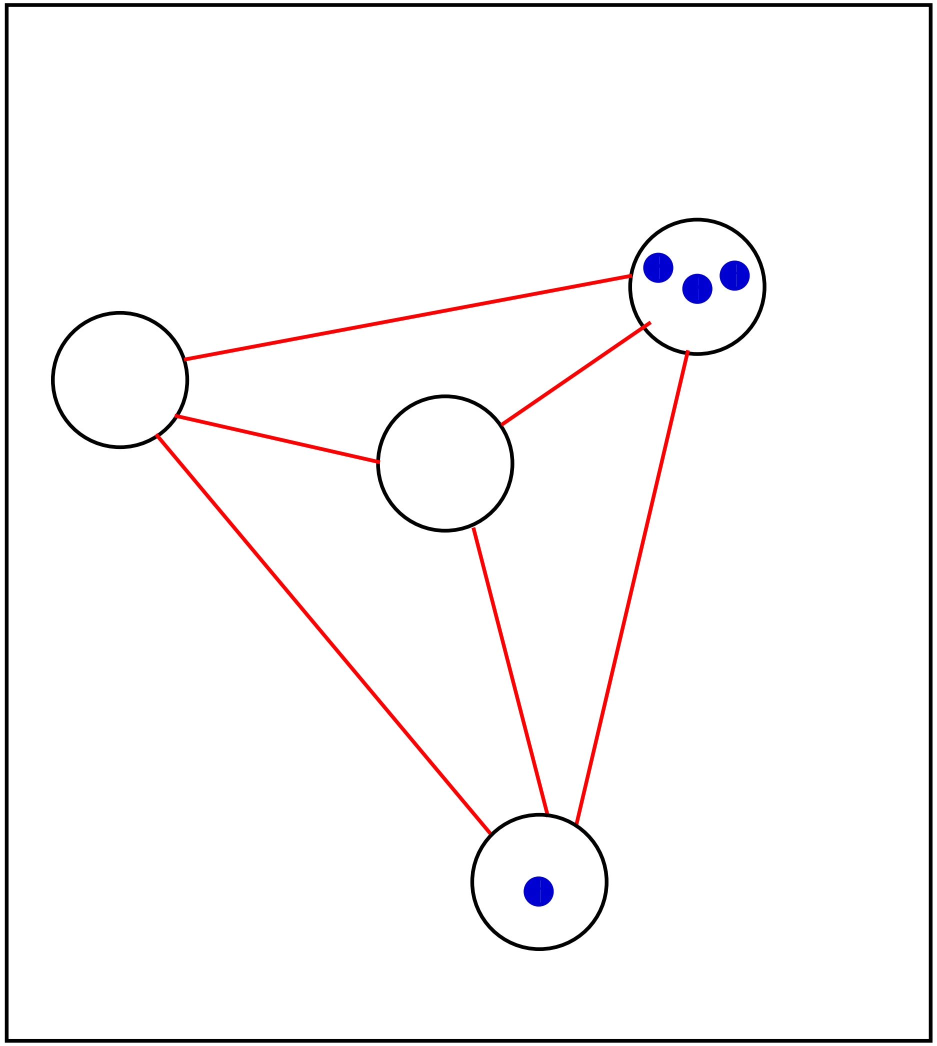
Example



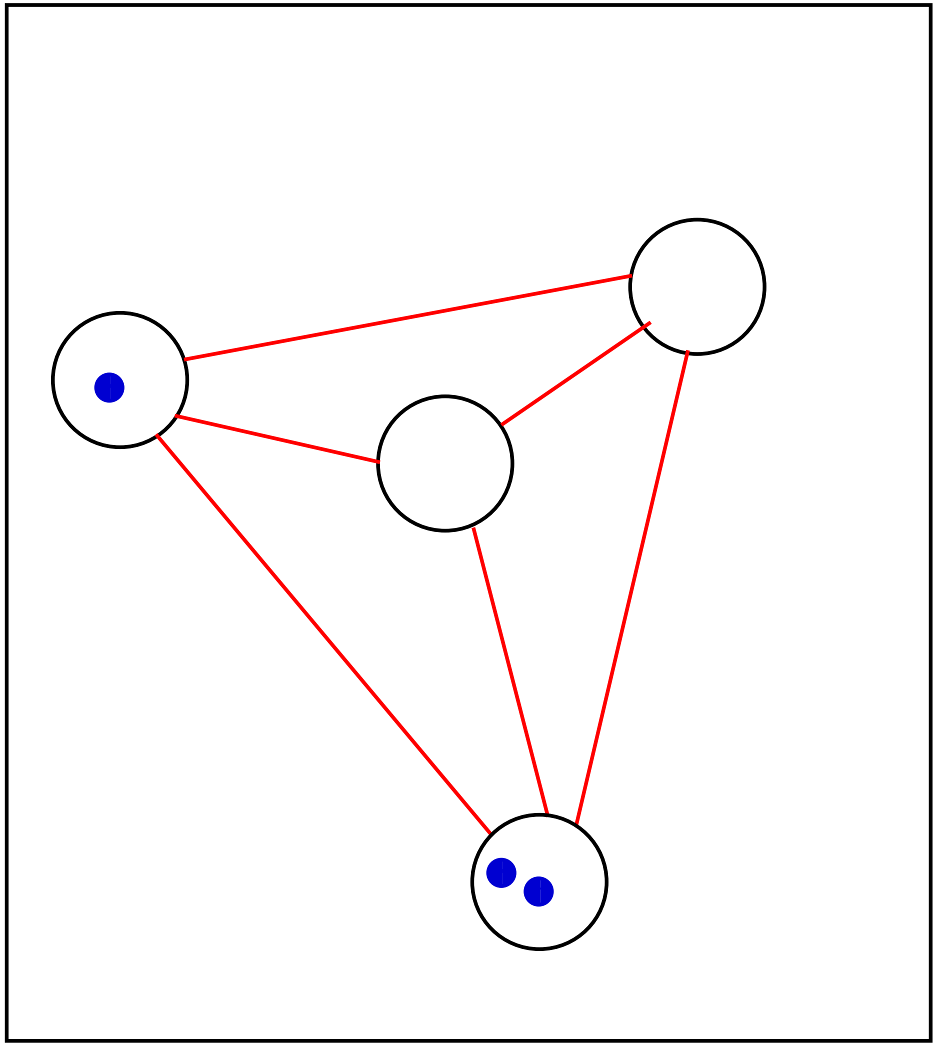
Example



Example



Example



Simple facts

Remarks

- The result does not depend on the order in which topplings are performed
- Proof: A toppling of site i consists in the addition of

$$\Delta_i = (1, 1, \ldots, -n, 1, \ldots 1)$$

Moreover addition is commutative, and if a toppling is also possible at site $j \neq i$ the addition of Δ_i will not modify this fact.

- After a certain number of topplings the configuration reached is stable

Markov chain

Operation A_i**:** Let u be a stable configuration, add 1 to u_i, then perform topplings until a stable configuration is reached.

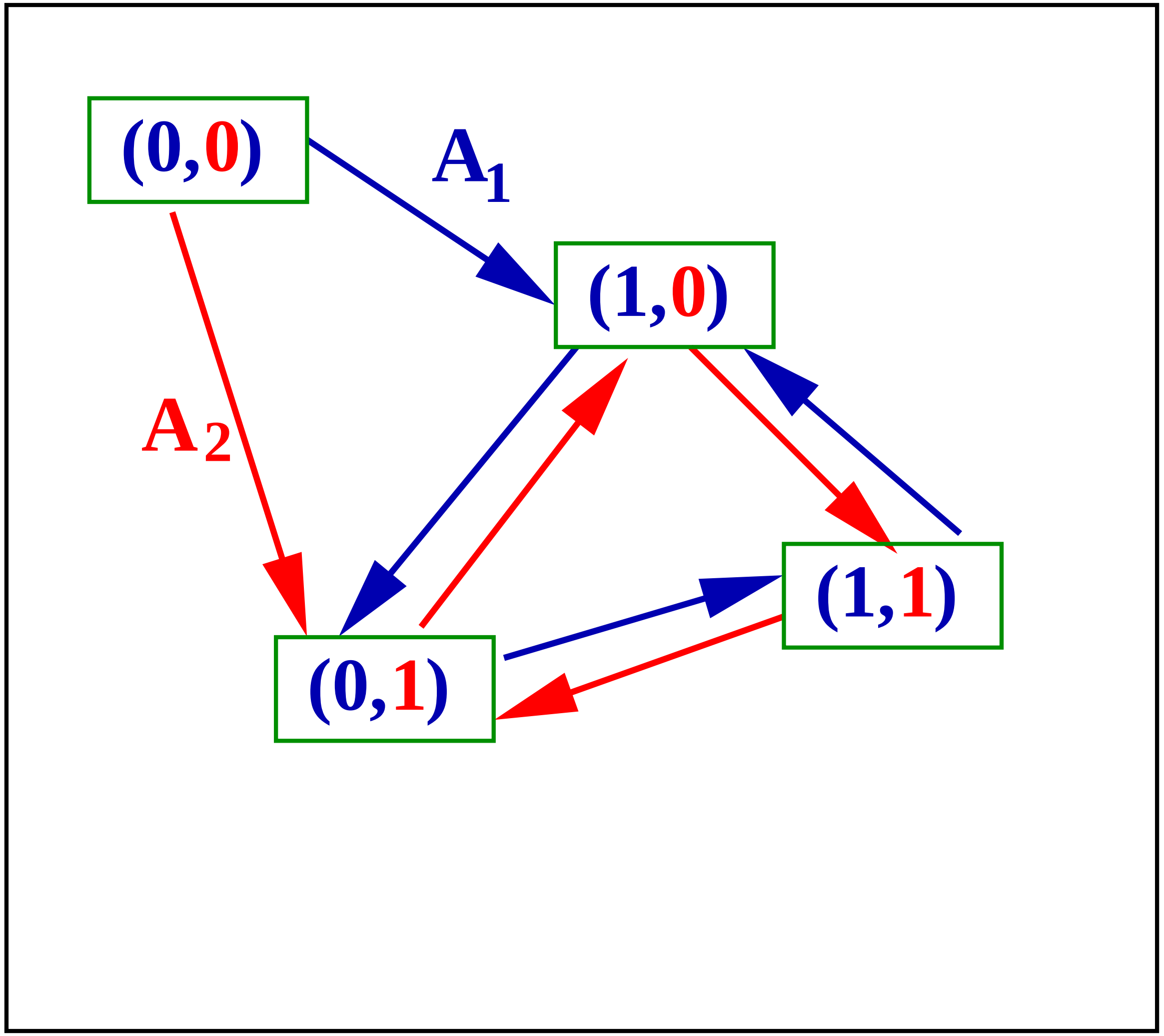
Example:

$$A_3(2,1,2) = (1,0,2)$$

$$(2,1,3) \to (3,2,0) \to (0,3,1) \to (1,0,2)$$

Behaviour of the Markov chain: choose i at random, then perform A_i

Markov Chain



Recurrent configurations

- A configuration u is *recurrent* if there is a (non empty!) sequence of operations A_i leading from u to itself.
- The recurrent configurations can all be reached one from the other
- The number of recurrent configurations is:

$$(n+1)^{n-1}$$

Dhar's algorithm

- A configuration u is recurrent if and only if the configuration v such that $\forall i, \quad v_i = u_i + 1$ satisfies:

$$v \quad \xrightarrow{*} u$$

Parking functions

A *parking* function is a sequence of non negative integers $u = u_1, u_2, \ldots, u_n$, such that there exists a permutation $a = a_1, a_2, \ldots, a_n$ satisfying :

$$\forall i, \quad u_i < a_i$$

For example, $3, 0, 1, 3, 1$ is parking function , use the permutation $4, 1, 3, 5, 2$; but $1, 4, 2, 0, 4$ is not.

Bijection between parking functions and recurrent configurations

Proposition The configuration

$$(u_1, \ldots, u_i, \ldots, u_n)$$

is recurrent if and only if

$$(n-1-u_1, \ldots, n-1-u_i, \ldots, n-1-u_n)$$

is a parking function

Proof: Use Dhar's criteria.

Consequence The number of parking functions of length n is:

$$(n+1)^{n-1}$$

General graphs

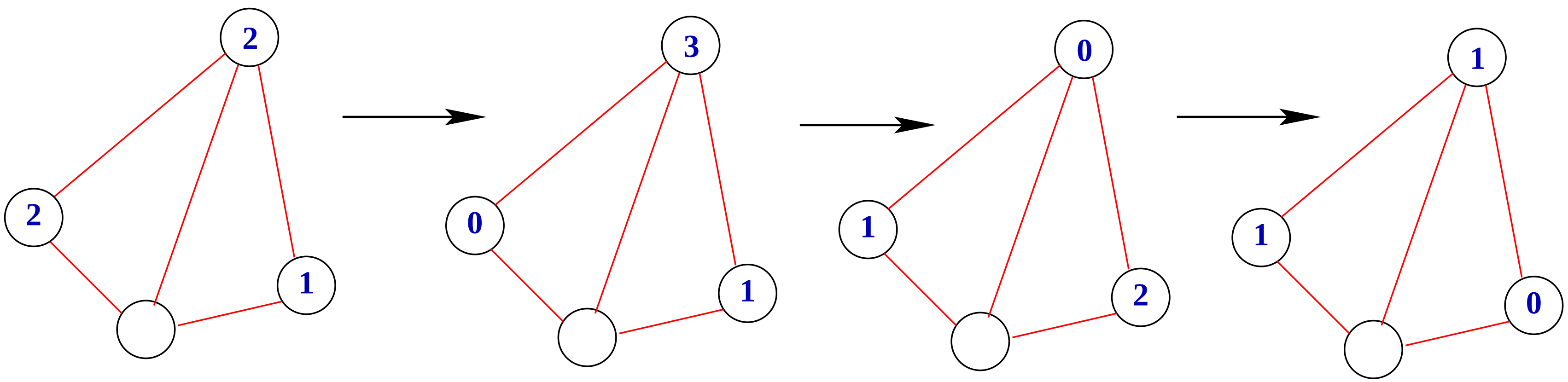
- Consider a non oriented connected graph $G = (X, E)$ with $n + 1$ vertices one of them being choosed as a *sink* the others being denoted $x_1, x_2, \ldots, x_n$.
- A configuration is a sequence of non negative integers $u_1, u_1, \ldots u_n$ the number u_i is considered as a number of chips (or of grains of sand) in vertex x_i.
- A toppling at vertex x_i can occur if u_i is not less than the degree d_i of this vertex, in that case we write $u \to u'$

$$\begin{cases} u'_i = u_i - d_i \\ u'_j = u_j + 1 \quad \text{if } j \text{ is a neighbour of } i \end{cases}$$

Stable configurations

- A configuration is stable if $u_i < d_i$ for all i
- From any configuration a stable one is reached after a finite number of topplings.
- The stable configuration attained does not depend on the order in which topplings are performed

Example



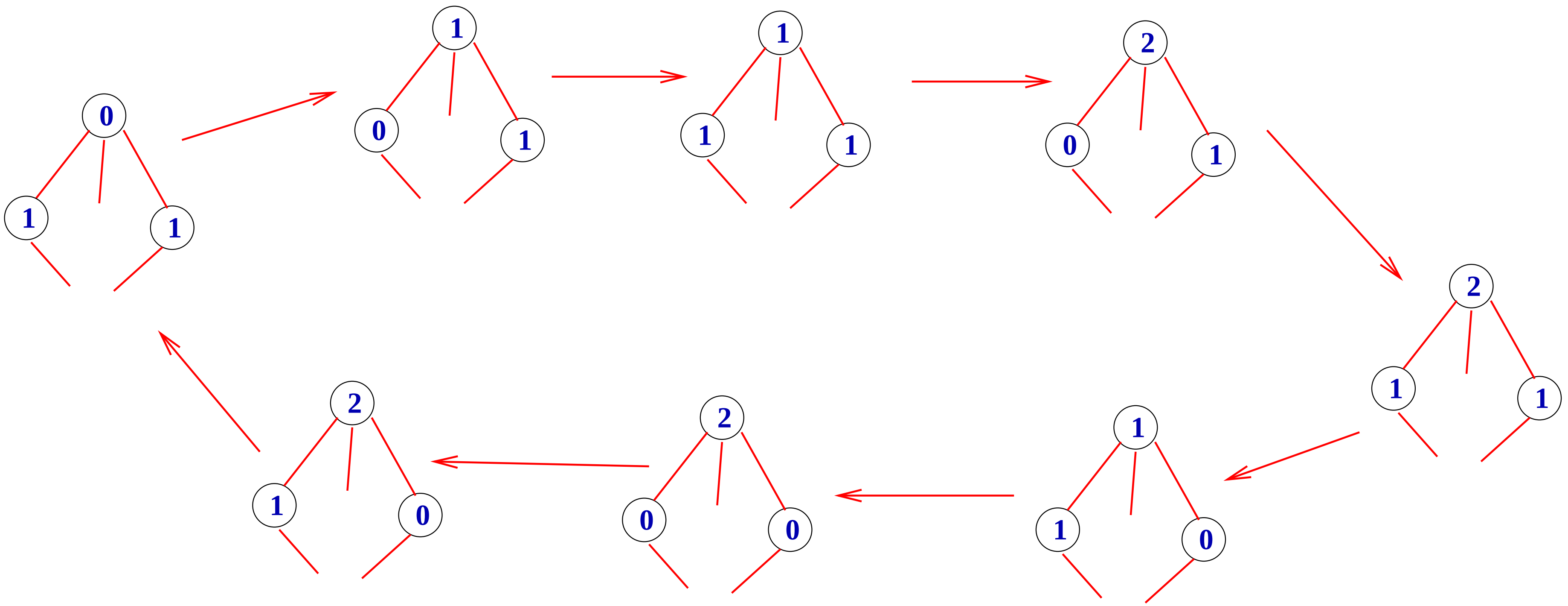
Recurrent configurations

- A Markov chain on stable configurations can be defined as well. The operation A_i consists in adding a grain of sand in vertex x_i then topple until a stable configuration is reached
- A configuration u is *recurrent* if there is a (non empty!) sequence of operations A_i leading from u to itself.
- **Theorem** The number of recurrent configurations is equal to the number of spanning trees of the graph.
- **Dhar's algorithm**

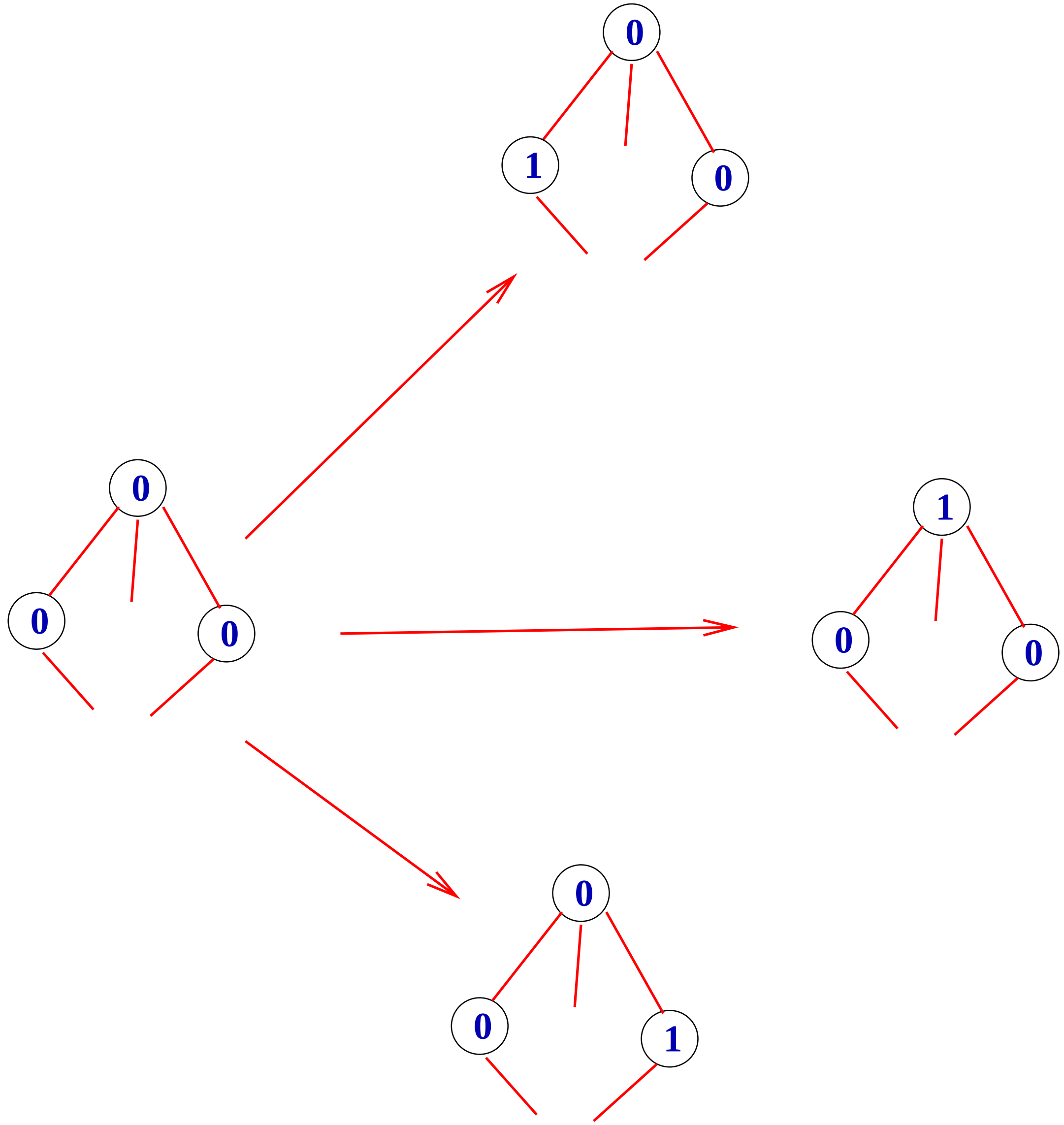
  A configuration u is recurrent if and only if the configuration v such that if i is a neighbour of the sink then $v_i = u_i + 1$, else $v_i = u_i$ satisfies:

$$v \xrightarrow{*} u$$

Recurrent configurations



Transient configurations



Some linear algebra

- A configuration on $G = (X, E)$ is a vector u :
- A toppling consists in substracting the vector Δ_i such that $\Delta_{i,i} = d_i$, $\Delta_{i,j}$is equal to the number of edges joining x_i and x_j
- Two configurations are equivalent if one can be obtained from the other by adding a linear combination of Δ_i
- This defines an equivalence relation and we have:

  Theorem Any class contains exactly one recurrent configuration.

Laplacian Matrix

- The vectors Δ_i are the lines of a matrix called the Laplacian matrix of the graph (in fact a minor of maximal size)
- The number of classes is the determinant of this minor

The graph K_{n+1}

$$\Delta = \begin{pmatrix} n & -1 & -1 & \cdot & \cdot & -1 \\ -1 & n & -1 & \cdot & \cdot & -1 \\ -1 & -1 & n & \cdot & \cdot & -1 \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ -1 & -1 & -1 & \cdot & \cdot & n \end{pmatrix}$$

Smith Normal Form

Any matrix M with integer coefficients may be decomposed in a product :

$$M = A\,D\,B$$

such that

- A and B are matrices with determinant equal to 1.
- D is a diagonal matrix
- Any element on the diagonal D divides the next one

Invariants of the toppling operation

- Use matrix $U = A^{-1}$
- $$D = U \Delta V$$
- Use the lines of U denoted: $\theta_1, \theta_2, \ldots, \theta_n$,
- Two configurations u, v are equivalent if and only if the products $< \theta_i, u >$ and $< \theta_i, v >$ are equal $mod\ d_i$ for any i

Bijections between parking functions and sequences of length $n-1$ composed of integers less than $n+1$

The graph K_{n+1}

$$\Delta = \begin{pmatrix} n & -1 & -1 & \cdot & \cdot & -1 \\ -1 & n & -1 & \cdot & \cdot & -1 \\ -1 & -1 & n & \cdot & \cdot & -1 \\ . & . & . & . & & . \\ . & . & . & & . & . \\ -1 & -1 & -1 & \cdot & \cdot & n \end{pmatrix} \quad D = \begin{pmatrix} n+1 & 0 & \cdot & \cdot & 0 & 0 \\ 0 & n+1 & \cdot & \cdot & 0 & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & & . & n+1 & 0 \\ 0 & 0 & . & . & 0 & 1 \end{pmatrix}$$

Pollak's bijection

$$U = \begin{pmatrix} 1 & 0 & \cdot & \cdot & 0 & -1 \\ \cdot & \cdot & \cdot & \cdot & 0 & -1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & . & . & 1 & -1 \\ 0 & 0 & . & . & 0 & 1 \end{pmatrix} \quad V = \begin{pmatrix} 2 & 1 & 1 & \cdot & \cdot & 1 \\ 1 & 2 & 1 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & . & . & 2 & 1 \\ 1 & 1 & . & . & 1 & 1 \end{pmatrix}$$

Another bijection

$$U = \begin{pmatrix} 2 & 1 & 1 & \cdot & \cdot & 1 \\ 1 & 2 & 1 & \cdot & \cdot & 1 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1 & 1 & . & . & 2 & 1 \\ 1 & 1 & . & . & 1 & 1 \end{pmatrix} \quad V = \begin{pmatrix} 1 & 0 & 0 & \cdot & \cdot & 0 \\ 0 & 1 & 0 & \cdot & \cdot & 0 \\ . & . & . & . & & . \\ . & . & . & & . & . \\ 0 & 0 & . & . & 1 & 0 \\ -1 & -1 & . & . & -1 & 1 \end{pmatrix}$$

Family of bijections

$$(p_1, p_2, \ldots, p_n) \rightarrow (q_1, q_2, \ldots, q_{n-1})$$

$$0 \leq q_i \leq n$$

$$q_i = p_i + \sum_{j=1}^{n} p_j$$

A new bijection for each matrix U such that there exists V satisfying:

$$D \ = U \, \Delta \, V \quad det(U) = det(V) = 1$$

Enumeration of configuration by their weights

Definition The weight $W(u)$ of a configuration u is the sum:

$$\sum_{i=1}^{n} u_i$$

Proposition The weight $W(u)$ of a recurrent configuration u satisfies:

$$m - d_{n+1} \leq W(u) \leq 2m - n - d_{n+1}$$

Proof:

$$W(u) \leq \sum_{i=1}^{n}(d_i - 1) = \sum_{i=1}^{n} d_i - \sum_{i=1}^{n-1} 1 = (2m - d_n) - n$$

For the lower bound label the grains by the edge they follow when a toppling is performed.

Relation with Tutte Polynomials (Biggs-Merino-Lopez-Le Borgne)

Definition The polynomial enumerating the recurrent configurations by their weights:

$$W_G(z) = \sum_{i=m-d_n}^{2m-d_n-n} c_i z^i$$

where c_i is the number of recurrent configurations of weight i.

Theorem The polynomial W_G is a specialisation of the Tutte polynomial T_G, more precisely:

$$W_G(z) = z^{m-d_n} T_G(1, z)$$

Generalized parking functions

- Given a sequence $x = x_1, x_2, \ldots, x_n$
- An x-*parking* function is a sequence of non negative integers $u = u_1, u_2, \ldots, u_n$, such that once sorted as $u' = u'_1, u'_2, \ldots, u'_n$ such that $u'_i \leq u'_{i+1}$ one has for all i :,

$$u'_i < \sum_{j=1}^{i} x_j$$

- Note that the usual parking functions are $(1, 1, 1, \ldots, 1)$-parkings
- Many papers consider $(a, b, b, \ldots, b)$-parking functions which are often called (a, b)-parking functions

Enumeration

The number of (a, b)-parking functiuons of length n est donné par :

$$a(a + bn)^{n-1}$$

Two proofs

1. Sandpile on a complete multi-graph with $n + 1$ vertices $v_0, v_1, \ldots v_n$ where v_0 is joined to all the other ones by a edges and such that any two vertices v_i, v_j, $i, j > 0$, are joined together by b edges.
2. Any sequence $w_1, \ldots w_n$ such that $0 \leq w_i < a + nb$ has exactly a conjuguates which are (a, b)-parking functions.

Other results

1. G-parking functions
2. Which graphs have D with only $D_{1,1} \neq 1$, that is the group is cyclic ?
3. The Tutte enumeration of inversion in trees and the Tutte Polynomial of the complete graphs.