SANDPILE MODEL AND TUTTE POLYNOMIALS

YVAN LE BORGNE

Abstract. A new explicit bijection between spanning trees and recurrent configurations of the sandpile model is given. This mapping is such that the difference between the number of grains on a configuration and the external activity of the associate tree is the number of edges of the graph. It is a bijective proof of a result of Merino-López that express the generating function of recurrent configurations as an evaluation of the Tutte polynomial.

Résumé. On donne une nouvelle bijection entre les arbres couvrants et les configurations récurrentes du modèle du tas de sable de Dhar. Cette application est construite de sorte que la difference entre le nombre de grains de la configuration et l'activité externe de l'arbre associé est le nombre d'arêtes du graphe. Cela donne une preuve bijective d'un résultat de Merino-López identifiant la distribution des configurations récurrentes selon leur nombre de grains à une évaluation partielle du polynôme de Tutte du graphe.

Introduction

The sandpile model was introduced in statistical physics ([1], [7], [6]) and was also considered by combinatorialists as the chip firing game ([4],[3],[8],[2]).

One of the interesting results about this model from a combinatorial point of view is that the Tutte polynomial of the underlying graph enumerates the configurations with respect to their level. This result was conjectured by Biggs and proved by Merino-López. His proof is based on the recursive definition of Tutte polynomials.

Our aim is to give an explicit bijection between spanning trees of the graph and recurrent configurations, which maps a tree with p externally active edges to a configuration of level p, giving a bijective proof of the above result.

Our paper is organized as follows. In section 1, we recall definitions of the sandpile model and the Tutte polynomial of a graph and then the result of Merino-López. In section 2, a mapping from recurrent configurations to spanning trees is defined. We give in section 3 an algorithm that compute a mapping from spanning trees to recurrent configurations. Section 4 is dedicated to the proof that those two mappings are bijective since one is inverse of the other.

1. Preliminaries

We present our notation and some definitions.

Let $G = (V, q, E)$ be a connected rooted pseudo-graph, where V is the set of vertices, $q \in V$ the root and E the multi-set of the edges. The endpoints of an edge (ab) are a and b. For any vertex a, $deg(a)$ denotes its degree. $|A|$ denotes the cardinality of the set (or multi-set) A.

Even if the following definitions and mappings are correct for graphs with loops, we assume that the graph has no loops. Graphs with loops will be considered later (see section 4).

The Sandpile model. A *configuration* u on $G = (V, q, E)$ is a mapping $V \longrightarrow \mathbb{N}$ (where $\mathbb{N}$ denotes the set of non-negative integers). An interpretation is that the vertex $a \in V$ contains $u(a)$ grains of sand. A vertex a is *unstable* in u if $u(a) \geq deg(a)$; then this vertex could be *toppled* giving the configuration v defined by:

$$\begin{cases} v(a) = u(a) - |\{(at) | t \neq a, (at) \in E\}| \\ v(b) = u(b) + |\{(ba) | (ba) \in E\}| \ for \ b \neq a \end{cases}$$

In other words, the vertex u passes one grain of sand along each of its edges to the vertex on the other end of the edge. We will denote by $u \longrightarrow^a v$ the toppling of a. A configuration is *stable* if all vertices different from q are stable. A stable configuration u is *recurrent* if $u(q) = deg(q)$ and there exists an order $a_1 <_u a_2 <_u ... <_u a_{|V|}$ of the vertices such that $u \longrightarrow^{a_1} ... \longrightarrow^{a_i} ... \longrightarrow^{a_{|V|}} u$. This means that if q is toppled then all vertices of V could be toppled getting back the initial configuration. $Rec(G)$ denotes the set of recurrent configurations of G. The *level* of a recurrent configuration u is given by

$$level(u) = (\sum_{i \in V} u_i) - |E|.$$

Let $\mathcal{R}(G, y)$ be the polynomial which codes the distribution of recurrent configurations according to their level:

$$\mathcal{R}(G, y) = \sum_{u \in Rec(G)} y^{level(u)}.$$

The Tutte polynomial. Given a graph $G = (V, E)$ and a order $<_e$ over E, Tutte ([9]) introduced the two-variable polynomial $\mathcal{T}(G, x, y)$ defined by:

$$\mathcal{T}(G, x, y) = \sum_{T \in \Sigma(G)} x^{i(T)} y^{j(T)}$$

where $\Sigma(G)$ is the set of spanning trees of G and $i(T)$ (respectively $j(T)$) is the internal (respectively external) activity of T.

An edge $e \in T$ is *internally active* in T if it is the lowest edge, according to $<_e$, in the unique cocyle contained in $(E \backslash T) \cup \{e\}$ (A *cocyle* is a minimal disconnecting subset of E). The *internal activity* of the spanning tree T is the number of edges internally active in T.

An edge $e \in E \backslash T$ is *externally active* in T if it is the lowest edge, according to $<_e$, in the unique cycle of $(V, T \cup \{e\})$. The *external activity* of the spanning tree T is the number of edges externally active in T.

A first non-trivial observation is that this polynomial does not depend on the arbitrary order $<_e$ even if it is the case for the activities of a given spanning tree.

Merino-López result. Answering to a conjecture of Biggs, Merino-López [8] proved that:

Theorem 1. *For a graph $G = (V, E)$, a vertex $q \in V$ and an order $<_e$ on E,*

$$\mathcal{R}(G, y) = \mathcal{T}(G, 1, y).$$

This means that there are as many recurrent configurations of level i as spanning trees of external activity i.

His proof is based on a recursive characterization of the Tutte polynomial, which do not gives a one-to-one correspondence between trees and configurations. Note that Majumdar and Dhar ([6]) already gave a bijection between spanning trees and recurrent configurations. The aim of this note is to build a sharper bijection mapping recurrent configurations of level i and spanning trees of external activity i.

2. From spanning trees to recurrent configurations

We first define a mapping $\mathcal{U}$ from $\Sigma(G)$ to $Rec(G)$.

Construction of $\mathcal{U}$. Let T be a spanning tree of $G = (V, q, E)$.

The key point in the construction is an order $<_T$ over the vertices of G deduced from $<_e$ and T. We define $<_T$ through the sequence $(X_k)_{1\leq k\leq |V|}$ such that $X_k \subseteq V$ consists of the k lowest vertices according to $<_T$:

$$\begin{cases} X_1 = \{q\} \\ X_{k+1} = X_k \cup \{x_{k+1}\} \textit{ where the largest edge in } T, \textit{ according to } <_e, \\ \quad \textit{with endpoints in } X_k \textit{ and } V\backslash X_k \textit{ is } (zx_{k+1}) \ (z \in X_k) \end{cases}$$

Using this order we give two rules to put grains on vertices. To keep in mind which rule we apply, we consider *black* and *white* grains.

Black grain rule:: For any edge $(ab) \in E$, add a *black grain* on a if and only if $a \leq_T b$.

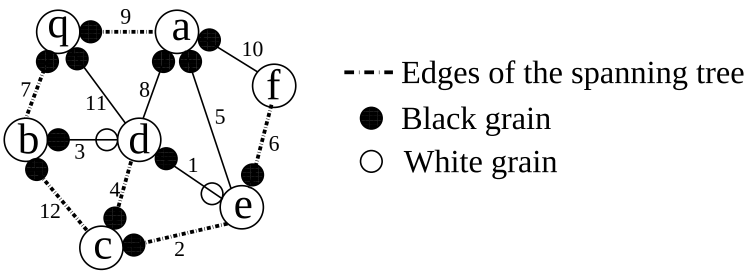
White grain rule:: For any externally active edge $(ab) \in E$, add a *white grain* on b such that $a \leq_T b$.

$\mathcal{U}(T)$ is the configuration made of the black and white grains. More precisely, for any vertex a:

$$\mathcal{U}(T)(a) = |\{(at)|a \leq_T t, (at) \in E\}| + |\{(at)|a \geq_T t, (at) \textit{ externally active}\}|.$$

Example. Here is an example of this construction: $G = (V, q, E)$ where $V = \{q, a, b, c, d, e, f\}$ and $E = \{(de), (ce), (bd), (cd), (ae), (ef), (bq), (ad), (aq), (af), (dq), (bc)\}$ (edges are given in increase order according to $<_e$).

The tree T is given by $\{(qa), (qb), (bc), (cd), (ce)\}$.



The order $<_T$ is :

$$q <_T a <_T b <_T c <_T d <_T e <_T f.$$

Since only (bd) and (de) are externally active we have $\begin{cases} \mathcal{U}(T)(q) = 3 + 0 = 3 \\ \mathcal{U}(T)(a) = 3 + 0 = 3 \\ \mathcal{U}(T)(b) = 2 + 0 = 2 \\ \mathcal{U}(T)(c) = 2 + 0 = 2 \\ \mathcal{U}(T)(d) = 1 + 1 = 2 \\ \mathcal{U}(T)(e) = 1 + 1 = 2 \\ \mathcal{U}(T)(f) = 0 + 0 = 0. \end{cases}$

Properties of $\mathcal{U}(T)$. We first prove that $\mathcal{U}(T)$ is a stable configuration; then we will show that it is a recurrent one whose level is the external activity of T.

Lemma 1. *For any $T \in \Sigma(G)$, $\mathcal{U}(T)$ is a* stable *configuration.*

Proof. Any edge incident to vertex a adds at most one grain (black or white) to a. Moreover if $a \neq q$, consider the last edge (ba) of the path from q to a in T. This edge adds no grain to a since by definition of $<_T$, $b <_T a$ (no black grain) and an edge of T can not be externally active (no white grain). Hence any vertex has a number of grains strictly less than its degree. $\square$

If we only take into account the black grains of the configuration $\mathcal{U}(T)$, we have the configuration $\mathcal{B}(T)$ defined by

$$\mathcal{B}(T)(a) = |\{(ab) \in E | a <_T b\}| \ \forall a \in V.$$

Lemma 2. *For any $T \in \Sigma(G)$, the configuration $\mathcal{B}(T)$ is recurrent.*

Proof. For any a, $\mathcal{B}(T)(a) \leq \mathcal{U}(T)(a)$; so $\mathcal{B}(T)$ is stable.

Also q is minimal in $<_T$; so for each incident edge (qt), $q <_T t$. The black grain rule gives $\mathcal{B}(T)(q) = \mathcal{U}(T)(q) = deg(q)$.

To check that the configuration is recurrent we prove that we can topple the vertices in the order given by $<_T$. Before the toppling of vertex a, by assumption all vertices b, $b <_T a$, have been toppled adding one grain per edge (ba). Those $|\{(ba) \in E | b <_T a\}|$ grains are added to the initial $|\{(ba) \in E | a <_T b\}|$ black grains. There are thus exactly $deg(a)$ grains on a which is unstable and can be toppled. So

$$\mathcal{B}(T) \longrightarrow^{x_1 = q} \ldots \longrightarrow^{x_{|V|}} \mathcal{B}(T).$$

Thus $\mathcal{B}(T)$ is recurrent.

□

Corollary 2. *For any $T \in \Sigma(G)$, $\mathcal{U}(T)$ is a recurrent configuration and $level(\mathcal{U}(T)) = j(T)$.*

Proof. Since $\mathcal{B}(T) \leq \mathcal{U}(T)$ and $\mathcal{U}(T)$ is stable, $\mathcal{U}(T)$ is recurrent.

Finally, white grains are in bijection with active edges and black grains with all the edges, hence $level(\mathcal{U}(T)) = j(T)$. □

Then $\mathcal{U}$ is an application from $\Sigma(G)$ to $Rec(G)$ which identifies the external activity of the tree T and the level of recurrent configuration $\mathcal{U}(T)$. To prove that it is a bijective mapping we construct in the next section its inverse which we denote by $\mathcal{T}$.

3. From recurrent configurations to spanning trees

Algorithm for $\mathcal{T}$. To define $\mathcal{T}$, we describe an algorithm which given a recurrent configuration u builds a spanning tree $\mathcal{T}(u)$. This tree is the last of an increasing sequence of subtrees Tr of G. The main loop of the algorithm adds successively edges to the current subtree and R collect all the edges which have been treated. In the description below, X consists of the ends points of edges in the subtree Tr.

Input: The recurrent configuration u.
Initialization: $X = \{q\}$, $R = \emptyset$, $Tr = \emptyset$.
Main loop:
while $(X \subsetneq V)$ **do**
  $(ab) = max_{<_e}\{(st) | s \in X, t \notin X, (st) \notin R\}$, $(a \in X, b \notin X)$
  $R = R \cup \{(ab)\}$
  if $|\{(bt) | (bt) \notin R\}| = u(b)$ **then**
    $X = X \cup \{t\}$
    $Tr = Tr \cup \{(ab)\}$
  end if
end while
Output: The spanning tree Tr.

Properties of the algorithm. We consider some invariants of the algorithm satisfied at the first step inside the while loop.

Lemma 3. *If u is stable each time the main loop is executed, $x \notin X \Longrightarrow Pot_R(x) = |\{(xt) | (xt) \notin R\}| > u(x)$.*

Proof. Initially $R = \emptyset$ and for any $x \neq q$, $Pot_\emptyset = deg(x) > u(x)$ since u is stable. After an iteration of the loop $Pot_R(x)$ decreases by at most one and this happens when the edge added to R is incident to x. If $Pot_R(x) = u(x)$ then x is added to X, so for all the other vertices $y \notin X$ we have $Pot_R(y) > u(y)$. □

The following lemma shows that Tr is well defined.

Lemma 4. *If u is recurrent, then the algorithm gives a spanning tree of G.*

Proof. Since after each iteration of the loop an edge is added to R, the only possible failure is when $\{(st) | s \in X, t \notin X, (st) \notin R\} = \emptyset$ and $X \neq V$. We prove that this can not be the case when u is recurrent.

Let $w_1, w_2, ..., w_{|V|}$ be an order in which we can topple the vertices in the configuration u. Let $w_j = min\{w_i | w_i \notin X\}$. By our assumption and lemma 3 we have $u(w_j) < Pot_R(w_j)$. If we assume that all the vertices of X are toppled then we have added $deg(w_j) - Pot_R(w_j)$ grains, which is an upper bound on the number of grains added to a after the toppling of $w_1, ...w_{j-1}$. So there are at most

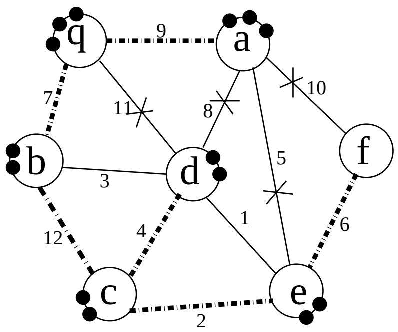
$$u(w_j) + (deg(w_j) - Pot_R(w_j)) < Pot(w_j) + (deg(w_j) - Pot(w_j)) = deg(w_j)$$

grains on w_j at this step. This means that it is stable, in contradiction with the choice of $(w_i)_{1 \leq i \leq |V|}$. Hence the algorithm ends with $X = V$ if the configuration is recurrent.

Tr together with its set of incident vertices is a connected subgraph of G which includes all its vertices and has $|V| - 1$ edges; so it is a spanning tree.

□

Example. We apply the previous algorithm on the recurrent configuration obtained in the example illustrating the definition of $\mathcal{U}$.



Here is a partial description of the execution:

$X = \{q\}, R = \emptyset, Tr = \emptyset;$
$(dq) \to R;$
$(aq) \to R, (aq) \to Tr, a \to X;$
$(af) \to R;$
$(ad) \to R;$
$(bq) \to R; (bq) \to Tr, b \to X;$
$(bc) \to R; (bc) \to Tr, c \to X;$
$(ae) \to R;$
$(cd) \to R; (cd) \to Tr, d \to X;$
$(ce) \to R; (ce) \to Tr; e \to X;$
$(ef) \to R; (ef) \to Tr; f \to X;$

In this case we have $\mathcal{T}(\mathcal{U}(T)) = T$ and moreover the externally active edges in T are exactly those which are not added to R by the algorithm. The next section will prove this fact.

4. $\mathcal{T} = \mathcal{U}^{-1}$

Relations between the two definitions. The two following lemmas link the constructions of $\mathcal{U}$ and $\mathcal{T}$.

Lemma 5. *In the algorithm for $\mathcal{T}$, vertices are added to X in increasing order according to $<_{\mathcal{T}(u)}$.*

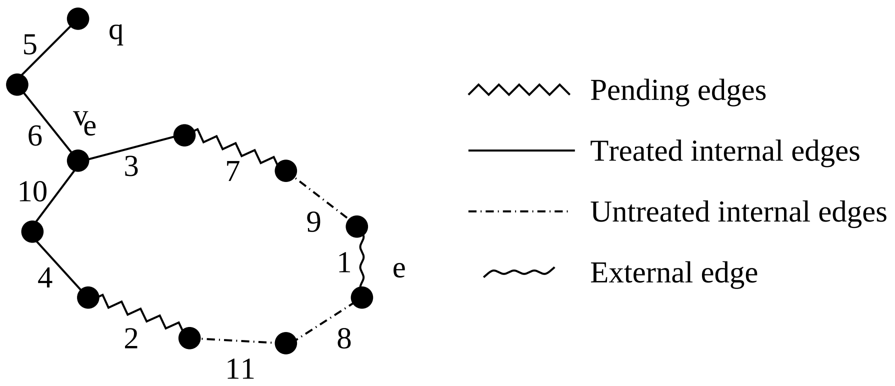
Proof. Initially $X = \{q\}$ and $q = min_{<_{\mathcal{T}(u)}}\{a \in V\}$. Then, since we consider edges in decreasing order with respect to $<_e$, we add successively to X the vertex which is an endpoint of the maximal edge, so we follow $<_{\mathcal{T}(u)}$. □

Lemma 6. *For any $u \in Rec(G)$, let R^u be the set of edges treated by the algorithm in the computation of $\mathcal{T}(u)$. Then*

$$e \notin R^u \iff e \textit{ is externally active in } \mathcal{T}(u).$$

The proof of this lemma requires some definitions. Given a spanning tree Tr and an external edge $e = (ab) \in E\backslash Tr$, $C_{e,Tr}$ denotes the edges of the unique cycle of $(V, Tr \cup \{e\})$, $V_{e,Tr}$ the vertices on this cycle and v_e is the first vertex added to X during the computation of Tr. There are two directions to walk through the cycle from v_e to v_e, we arbitrary call one the *clockwise walk* and the other the *counterclockwise walk*. At any step of the computation, described by (X, R), the *clockwise pending edge* (respectively *counterclockwise pending edge*) is the first edge $f = (cd)$ such that $c \in X$ and $d \notin X$ along the clockwise walk (respectively counterclockwise walk).

Example. This example may help to follow definitions and proofs.



At this step of the algorithm, the edges $5, 6, 10, 4, 3$ have been treated in this order and $7, 9, 2, 11, 8$ will follow. The external edge 1 will not be treated and is externally active.

Lemma 7. *When an edge of $C_{e,Tr}$ is added to R it is a pending edge.*

Proof. Futhermore, we show that "$(V \cap X, C_{e,Tr} \cap R)$ is connex" is a loop invariant.

$K_{e,Tr}(X, R)$ denotes the quantity $|V_{e,Tr} \cap X| - |C_{e,Tr} \cap (R \cap Tr)|$.

At (X^0, R^0) such that $V_{e,Tr} \cap X^0 = \{v_e\}$ and $C_{e,Tr} \cap (R^0 \cap Tr) = \emptyset$ we have $K_{e,Tr}(X^0, R^0) = 1$. At the end of the computation $((X^f, R^f))$ all the edges of $C_{e,Tr}$ different of e are in $R \cap Tr$ and $V_{e,Tr} \subseteq X$ then again $K_{e,Tr}(X^f, R^f) = 1$.

We discusse the modification of $K_{e,Tr}$ depending of the type of the added edge f:

1. $f \notin C_{e,Tr}$ and has no endpoints in $V_{e,Tr}\backslash X$. $K_{e,Tr}$ keeps the same value. $(V \cap X, C_{e,Tr} \cap R)$ remains connex if it was the case.
2. $f \notin C_{e,Tr}$ and has one endpoint in $V_{e,Tr}\backslash X$. $K_{e,Tr}$ decreases by 1.
3. $f \in C_{e,Tr}$ then it has one endpoint in $V_{e,Tr}\backslash X$. So $K_{e,Tr}$ keeps the same value. Moreover $(V \cap X, C_{e,Tr} \cap R)$ remains connex if it was the case.

Since $K_{e,Tr}$ could only decrease and is equal at the beginning and the end, it remains constant. So (2) is impossible. Therefore $(V \cap X, C_{e,Tr} \cap R)$ remains connex hence the only edges $f = (cd)$ of $C_{e,Tr}$ such that $c \in X$ and $d \notin X$ are the pending ones. □

Now we can define the *comparing edge* of an edge f of $C_{e,Tr}$ as the pending edge in the opposite direction when f is added to R. For any edge $e \in E$, let X_e (respectively R_e) be the set of vertices covered (respectively treated edges) when e is considered in the main loop.

Lemma 8. *If $f \in R$ then its comparing edge is lower, according to $<_e$, than any edge of $(R_f \cup \{f\}) \cap C_{e,Tr}$.*

Proof. Initially $R_f \cap C_{e,Tr} = \emptyset$, since the comparing edge g has endpoints in X_f and $V \backslash X_f$ and f is maximal among those edges, $g <_e f$. Then at this step the lemma is satisfied.

Now assume β is the comparing edge of the last edge of $C_{e,Tr}$ added to R and α the new pending edge. (possibly $\alpha = \beta$) The next edge of $C_{e,Tr}$ added to R is one of this two edges. Moreover it is the minimum according to $<_e$. In any case we have $\beta = min_{<_e}(R \cap C_{e,Tr})$. If α is added to R, $\alpha = max_{<_e}\{(st) \in E | s \in X, t \notin X\}$, and $\beta \in \{(st) \in E | s \in X, t \notin X\}$ then $\beta \leq_e \alpha$. So β is lower than any edge of $(R \cup \{\alpha\}) \cap C_{e,Tr}$. If β is added to R, for the same reasons, $\alpha \leq_e \beta$ and by transitivity α is lower than any edge of $R \cap C_{e,Tr}$.

□

We deduce lemma 6 from lemma 8.

Proof. (lemma 6)

If $e \notin R$, then it was the comparing edge of the last edge of $C_{e,Tr}$ added to R. Then e is lower than any edge of $C_{e,Tr} \backslash \{e\}$ so it is externally active.

If $e \in R$, it can not be the last edge of $C_{e,Tr}$ added to R else it would have both endpoints in R. So e is different from its comparing edge which is in $C_{e,Tr}$ and lower than e. Hence e can not be externally active. □

Proof of $\mathcal{T} = \mathcal{U}^{-1}$ for graphs with no loops.

Theorem 3. *For any graph $G = (V, q, E)$ with no loops, any order $<_e$ and any recurrent configuration u*

$$\mathcal{U} \circ \mathcal{T}(u) = u$$

Proof. Since $\mathcal{U} \circ \mathcal{T}(u)$ and u are recurrent, $u(q) = deg(q) = \mathcal{U} \circ \mathcal{T}(u)(q)$. So the relation holds on vertex q.

Let $t \in V \backslash \{q\}$. During the computation of $\mathcal{T}(u)$, t is added to X with the edge $f = (xt)$, $x \in X_f$.

$u(t) =^{(1)} Pot_{R_f}(t) - 1$
$=^{(2)} |\{e \in E | e = (tz), e \notin R_f, e \neq f, z \in V\}|$
$=^{(3)} |\{e \in E | e = (tz), e \notin R_f, e \neq f, z \in X_f\}| + |\{e \in E | e = (tz), e \notin R_f, z \notin X_f\}|$
$=^{(4)} |\{e \in E | e = (tz), e \notin R_f, e \neq f, z <_{\mathcal{T}(u)} t\}| + |\{e \in E | e = (tz), t <_{\mathcal{T}(u)} z\}|$
$=^{(5)} |\{e \in E | e = (tz), e \notin R^u, z <_{\mathcal{T}(u)} t\}| + |\{e \in E | e = (tz), t <_{\mathcal{T}(u)} z\}|$
$=^{(6)} |\{e \in E | e = (tz), e \text{ externally active in } \mathcal{T}(u), z <_{\mathcal{T}(u)} t\}| + |\{e \in E | e = (tz), t <_{\mathcal{T}(u)} z\}|$
$=^{(7)} \mathcal{U} \circ \mathcal{T}(u)(t)$.

(1) and (2) come from the definition of $\mathcal{T}$, (3) is a splitting of the multi-set. (4) comes from lemma 5. Since t is covered by f, any edge $e = (zt)$, $z \in X_f$ would have both endpoints in X until the end of the algorithm. So if e is not treated before f, it never will be the case. This gives (5), (6) follows from lemma 6 and (7) from the definition of $\mathcal{U}$.

□

Corollary 4. *$\mathcal{U}$ and $\mathcal{T}$ are opposite bijective mappings such that the level of a recurrent configuration is equal to the external activity of the corresponding spanning tree.*

Proof. Since $\mathcal{U} \circ \mathcal{T} = Id_{Rec(G)}$, $\mathcal{U}$ is a surjection and $\mathcal{T}$ an injection. To avoid a similar rather technical proof to show that we have also $\mathcal{T} \circ \mathcal{U}(T) = T$ for any spanning tree T, we can use the fact that there are finitely as many spanning trees than recurrent configurations (Theorem 5). Thus $\mathcal{U}$ and $\mathcal{T}$ are bijections. For self-contain, we give below sketch of a proof of this result (the complete proof is in [6]). From corollary 2 we know that in this one-to-one correspondence the level of the configuration is equal to the external activity of the tree. $\square$

Theorem 5. *(Dhar-Majumdar) For any graph $G = (V, E, q)$, $|Rec(G)| = |\Sigma(G)|$.*

Proof. Let $v \in V$, then $\mathcal{O}_v$ is a total order over all edges incident to v. Given a family $(\mathcal{O}_v)_{v \in V}$ we build an application $\mathbb{U}$ from $Rec(G)$ to $\Sigma(G)$. Let u be a recurrent configuration. The spanning tree $\mathbb{U}(u)$ is defined by the choice at vertex $v \neq q$ of the first edge of the path from v to q in $\mathbb{U}(u)$. This edge is called the *covering edge of* v. We perform one toppling of q, then all the others step by step. At step 0, only q is toppled and it is the unique time. At step $t+1$ all the unstable vertices after step t are toppled. Since u is recurrent any vertex v is toppled one time at step t_v. Any edge e incident to v whose opposite endpoint topples, adds one grain to v. At step $t_v - 2$ there are $a < deg(v)$ grains on v then at least $deg(v) - a$ incident edges add grain to v during step $t_v - 1$, the covering edge of v is the $(deg(v) - a)$-th of them according to $\mathcal{O}_v$.

The *height* $h^T(v)$ of a vertex v in the spanning tree T is the length of the path from v to q. Given a spanning tree T, the vertex v will be toppled at step $h^T(v)$. If $f \in T$ is the covering edge of v then $u(v) = deg(v) - |\{e = (wv) | h^T(w) < h^T(v) - 1\}| - |\{e = (vw) | h^T(w) = h^T(v) - 1, e \leq_{\mathcal{O}_v} f\}|$. $\square$

Graphs with loops. Loops are relevant since they appear in the recursive characterization of Tutte polynomial. Loops are degenerate cases of usual edges so have additional properties. As regards Tutte definitions, a loop can not be in a spanning tree. Moreover, since it is itself a cycle, a loop is always externally active. In the sandpile model, a loop contribute for 2 to the degree of its incident vertex v. When v is toppled, two grains cross themselves along the loop, staying on v.

Since we now consider different graphs, any object O (set, function, order...) defined on the graph H is denoted by O^H if necessary. Given a graph $G = (V, E, q)$ with loops we consider its *loops-erased graph* $\Psi(G) = (V, E^{\Psi(G)} = E \backslash \{e = (vv) \in E\}, q)$. The order $<_e^{\Psi(G)}$ is the restriction of $<_e^G$ to $E^{\Psi(G)}$. The number of loops in G is denoted by l^G. For any vertex i, l_i^G denotes the number of loops of endpoints i. Any recurrent configuration u of G is mapped to the configuration $\Psi^u(u)$ of $\Psi(G)$ defined by :

$$\Psi^u(u)(i) = u(i) - 2l_i^G.$$

Any spanning tree T of G is mapped to the spanning tree $\Psi^T(T) = T$ of $\Psi(G)$.

Lemma 9. *$u \longmapsto \Psi^u(u)$ is a bijection between $Rec(G)$ and $Rec(\Psi(G))$, and $\forall u \in Rec(G)$:*

$$level(\Psi^u(u)) = level(u) - l^G.$$

$T \longmapsto \Psi^T(T)$ is a bijection between $\Sigma(G)$ and $\Sigma(\Psi(G))$, and $\forall T \in \Sigma(G)$:

$$j(\Psi^T(T)) = j(T) - l^G.$$

Proof. Since $V^G = V^{\Psi(G)}$ and $E^{\Psi(G)} \subseteq E^G$ all spanning trees of $\Psi(G)$ are spanning trees of G. Besides any edge of $E^G \backslash E^{\Psi(G)}$ is a loop so there is no other spanning tree. Hence Ψ^T is a bijection.

Any loop is externally active in T^G so Ψ^T remove l^G active edges. No loop can be in the cycle created by an other external edge so the activities of the other edges are not modified. Also $j(T^{\Psi(G)}) = j(T^G)) - l^G$.

The two grains which cross along the loop never leave the vertex, then we can remove the loop and these two grains without disturbing the other topplings. So if u^G is recurrent then $\Psi^u(u^G)$ also. Moreover Ψ^u is injective and from [6] we have $|Rec(G)| = |\Sigma(G)| = |\Sigma(\Psi(G)| = |Rec(\Psi(G))|$. Hence Ψ^u is a bijection.

Finally $level(\Psi^u(u)) = level(u) - \sum_{i\in V} 2l_i^G + |E^G \backslash E^{\Psi(G)}| = level(u) - l^G$. □

Corollary 6. *$(\Psi^T)^{-1} \circ \mathcal{T}^{\Psi(G)} \circ \Psi^u$ and $(\Psi^u)^{-1} \circ \mathcal{U}^{\Psi(G)} \circ \Psi^T$ are inverse bijections between $Rec(G)$ and $\Sigma(G)$ which identify level of configurations and external activity of spanning trees.*

Besides the first bijection is $\mathcal{T}^G$.

Proof. The loop (vv) has one grain from the black rule since $v \leq_T v$, and one grain from the white rule since it is externally active. So any loop adds 2 grains on its incident vertex as $(\Phi^u)^{-1}$.

To express $\mathcal{U}^G$ by an algorithm, we just have to replace $u(b)$ by $u(b) - 2l_b^G$. □

Thus we have the result of Merino-López.

Corollary 7. *For any graph with $G = (V, E)$, any vertex $q \in V$ and any order $<_e$ on E,*

$$\mathcal{R}(G, y) = \mathcal{T}(G, 1, y).$$

Conclusion

Given a graph, we gave a new bijection between its spanning trees and its recurrent configurations in the sandpile model. This mapping shows that, the external activity of the Tutte polynomial context can be translate into the number of grain of a recurrent configuration. It will be relevant to express the internal activity in the sandpile model too, giving with this model a proof that the Tutte polynomial does not depend on the arbitrary order over the edges.

The author wishes to thank Robert Cori for fruitful references and comments.

References

[1] Bak, P., and Tang, C., and Wiesenfeld, K., " Self-organized criticality : An explanation of $1/f$ noise" *Physical Review Letters.* **59(4)**(1987), 381–384.

[2] Biggs N., "The Tutte polynomial as a growth function", *J. Alg. Comb.* **10**, 115–133,1999.

[3] Biggs, N., and Winkler, P., "Chip-firing and the chromatic polynomial" *CDAM Research Report Series.* **LSE-CDAM-97-03**(1997).

[4] Björner, A., and Lovász, I., and Shor, P.W., "Chip-firing game on graphs" *European J. Combin.* **12** 283–291.

[5] Cori, R., and Rossin, D., "On the sandpile group of dual graphs" *European Journal of Combinatorics.* **21(4)**(2000) 447–459.

[6] Dhar, D. and Majumdar, S.N., "Equivalence of the abelian sandpile model and the $q \longrightarrow 0$ limit of the Potts model" *Physica A.* **185**(1992), 129–135.

[7] Dhar, D., and Ruelle, P., and Sen, S., and Verma, D.N., "Algebraic aspects of abelian sandpile model" *J. Phys.*, **A28**(1995) 805–831.

[8] López, C.M., "Chip firing and Tutte polynomial" *Annals of Combinatorics.* **3**(1997), 253–259.

[9] Tutte, W.T., "A contribution to the theory of chromatics polynomials" *Canad. J. Math.* **6**(1953), 80–91.

E-mail address: borgne@labri.u-bordeaux.fr

LaBRI, Université Bordeaux I, 351 Cours de la Libération, 33405 Talence Cedex, FRANCE