

On computation of Baker and Norine's rank on complete graphs

Robert Cori *

LaBRI
Université de Bordeaux
Talence, France
cori@labri.fr

Yvan Le Borgne †

CNRS, LaBRI
Université de Bordeaux
Talence, France
borgne@labri.fr

Submitted: May 5, 2014; Accepted: Feb 3, 2016; Published: Feb 19, 2016
Mathematics Subject Classifications: 05C88, 05C89

Abstract

The paper by M. Baker and S. Norine in 2007 introduced a new parameter on configurations of graphs and gave a new result in the theory of graphs which has an algebraic geometry flavor. This result was called Riemann-Roch formula for graphs since it defines a combinatorial version of divisors and their ranks in terms of configurations on graphs. The so called chip firing game on graphs and the sandpile model in physics play a central role in this theory.

In this paper we present an algorithm for the determination of the rank of configurations for the complete graph K_n . This algorithm has linear arithmetic complexity. The analysis of number of iterations in a less optimized version of this algorithm leads to an apparently new parameter which we call the prerank. This parameter and the parameter div provide an alternative description to some well known q, t -Catalan numbers. Restricted to a natural subset of configurations, the two natural statistics degree and rank lead to a distribution which is described by a generating function which, up to a change of variables and a rescaling, is a symmetric fraction involving two copies of Carlitz q -analogue of the Catalan numbers.

In annex, we give an alternative presentation of the theorem of Baker and Norine in purely combinatorial terms.

1 Introduction

On a graph, a configuration is a map from its vertices to the integers of \mathbb{Z} . Motivated by the generalization for graphs of a Riemann-Roch theorem, Baker and Norine [6] defined

*The first author acknowledges the support of ERC under the agreement “ERC StG 208471 - ExploreMap”

†Part of this work was done in PIMS, Simon Fraser University, Burnaby, Canada

for any configuration a parameter they called rank. This parameter is an integer greater or equal to -1 . The abelian sandpile model [4, 17, 15] provides a framework for the original definition of the rank which seems to involve non-trivial computations. At submission date of this work, the complexity of the computation of the rank for general graphs was unknown. Just after, Kiss and Tóthmérész announced on arXiv [25] their proof for general graphs that computation is NP-hard.

The complete graph K_n is the non-oriented graph with n vertices and exactly one edge between any pair of distinct vertices. We present here an algorithm which computes the rank of a configuration in the particular case of complete graphs. We prove that our most efficient version has a linear arithmetic complexity where the size of the problem is the number of vertices of the graph. The design of this algorithm has enumerative byproducts in the context of the rank parameter and also some intensively studied q, t -Catalan numbers [23]. The results in this second context strengthen the connection with the abelian sandpile model as in recent work involving one of the authors [19, 2].

Let $G = (V_G, E_G)$ be an non-oriented connected simple graph without loops and with vertex set V_G and edge set E_G . Baker's and, Norine's theorem [6] states that for any configuration $u = (u_i)_{i \in V_G} \in \mathbb{Z}^{|V_G|}$ of G ,

$$\text{rank}(u) - \text{rank}(K_G - u) = \text{degree}(u) + |E_G| - |V_G|$$

where the cardinality of a set E is denoted by $|E|$; the definition of the parameter rank is still postponed to the preliminaries in Section 2; the configurations are seen as vectors allowing additions and subtractions; the degree of the configuration u is $\text{degree}(u) := \sum_{i \in V_G} u_i$; the configuration $K_G := (k_i)_{i \in V_G}$ is defined by $k_i := d_i - 2$ where d_i is the degree of the vertex i which is its number of incident edges. In this context, a configuration is called a divisor but we preferred to use the abelian sandpile model terminology. This paper focus on the particular case where G is a complete graph K_n : we use the integers $1, 2, \dots, n$ to label the vertices V_{K_n} of the complete graph K_n ; the configuration $K_{K_n} = (k_i)_{i=1 \dots n}$ satisfies $k_i = n - 3$ for all i .

The two next sections are dedicated to the design of our main algorithm which computes the rank of a configuration on K_n . Section 2 focus on correctness of a first version of the algorithm. We use the numerous symmetries of the complete graphs to obtain a greedy algorithm computing the rank. The complexity of this version is not clear but the aim of Section 3 is to optimize this algorithm. We select a particular run of this non-deterministic first algorithm to reach the announced complexity while preserving the correctness. The optimization requires to work at the level of orbits of a subgroup of symmetries. The toppling equivalence, an additional equivalent relation on configurations based on the sandpile model framework, shows that one can efficiently reduce the computation to the case of a sorted parking configuration. The configuration u is a sorted parking configuration if $0 \leq u_i < i$ for $i < n$ and $u_1 \leq u_2 \leq \dots \leq u_{n-1}$. Notice that there is no constrain on the value of u_n . If we had used the divisor terminology, we may called this configuration a sorted n -reduced divisor. The optimized algorithm computes the rank via the following rather explicit expression for the rank of a sorted parking configuration

u in K_n :

$$\text{rank}(u) = \left(\sum_{i=1}^{n-1} \max(0, q - i + u_i + \chi(i \leq r)) \right) - 1.$$

where $u_n + 1 = q(n - 1) + r$ is the euclidean division defining the quotient q and the non-negative remainder r , and $\chi(P)$ is 1 if the proposition P is true and 0 otherwise.

The second optimized algorithm still manipulates the configurations as vectors. Yet the proof that it corresponds to a run of the first algorithm deeply relies on another coding of some configurations. These configurations on K_n may be encoded by a Dyck word of size $n - 1$ and one or two integers depending on its additional properties. A word v on the two-letter alphabet $\{a, b\}$ is a Dyck word of size n if $|v|_a = |v|_b = n$ and for any prefix p of v , $|p|_a \geq |p|_b$ where $|v|_c$ denotes the number of occurrences of letter c in the word v . This data is embedded in a cut skew cylinder to make use of a cyclic symmetry. The skew cylinder of circumference n is an already known slight variation of the usual $\mathbb{Z} \times \mathbb{Z}/(n - 1)\mathbb{Z}$ cylinder, for example already presented by Kramers and Wannier [22] in 1941. The main interest of this skew is that it merges the usually “parallel” finite cycles interpreting $\mathbb{Z}/(n - 1)\mathbb{Z}$ into a single infinite spiral traversal isomorphic to \mathbb{Z} . A cut of this skew cylinder is a self-avoiding directed cycle made up of north and east steps disconnecting the cylinder into two components. A classical cyclic lemma due to Dvoretzky and Motzkin [20] implies that any cut may be defined by the choice of a Dyck word of size $n - 1$ and a starting vertex. One interest of this change of encoding is that computation on configurations becomes simpler updates of the Dyck word or the integers. After all, our optimized algorithm on the configuration $u = (u_i)_i$ may be interpreted as a spiral traversal of $u_n + 1$ cells on a related cut skew cylinder counting the occurrences of cells in one of the two components defined by the cut.

In Section 4, we describe as follows the generating function

$$K_n(r, d) := \sum_u r^{\text{rank}(u)} d^{\text{degree}(u)} = \frac{d^{\binom{n-1}{2}}}{rd} M_n\left(\frac{1}{d}, rd\right)$$

where u runs over *sorted* parking configurations on K_n and $M_n(x, y)$ is the coefficient of z^{n-1} in

$$\mathbb{M}(x, y; z) := \frac{1 - xy}{(1 - x)(1 - y)} \frac{C(x; xz) + C(y; yz) - C(x; xz)C(y; yz)}{1 - C(x; xz)zC(y; yz)}$$

where $C(q; z)$ is a classical Carlitz q -analogue of Catalan numbers counting the Dyck words according to size via the variable z and the later defined area via the variable q . The formula for $\mathbb{M}(x, y; z)$ comes from the decomposition, on each cut skew cylinder, of the spiral traversal at crossings of the cut, leading to geometric sums of reasons x or y . If the sum in $K_n(r, d)$ were not restricted to sorted parking configurations, we would have reached a two-variable zeta function for graphs introduced recently by Lorenzini [28]. We provide a combinatorial interpretation of the generating function $M_n(x, y)$ in terms of cut skew cylinders. Those cut skew cylinders admits an natural involution ξ which

roughly speaking consists in reversing the spiral traversal. This involution also explains via a superimposition principle why $M_n(x, y) = M_n(y, x)$. In addition, this involution is related to the involution on configurations $u \rightarrow K_{K_n} - u$ appearing in Riemann-Roch theorem for graphs.

In Section 5, we study connections of this work with q, t -Catalan numbers presented for example by Haglund in [23]. These q, t -Catalan numbers $(C_n(q, t))_{n \in \mathbb{N}}$ may be combinatorially described by some pairs of statistics on a generic Dyck word w chosen among the later defined $\text{area}(w)$, $\text{bounce}(w)$ and $\text{dinv}(w)$:

$$C_n(q, t) := \sum_w q^{\text{area}(w)} t^{\text{dinv}(w)} = \sum_w q^{\text{area}(w)} t^{\text{bounce}(w)}$$

where w runs over Dyck words of size n . The symmetry $C_n(q, t) = C_n(t, q)$ is obvious once non-trivially proven equivalent to some alternative definitions. There is still no combinatorial explanation of this symmetry [24]. In this context, Haglund designed an involution ζ on Dyck words which satisfies for any Dyck word w :

$$(\text{dinv}(w), \text{area}(w)) = (\text{area}(\zeta(w)), \text{bounce}(\zeta(w)))$$

establishing combinatorially the equivalence of the two presented definitions of $C_n(q, t)$. Our involution ξ on cut skew cylinders may be canonically restricted to Dyck words and this restriction also called ξ . It appears that $\text{dinv}(\xi(w)) = \text{dinv}(w)$. An early optimization of our algorithm computing the rank leads us to the apparently new parameter $\text{prerank}(w) = \text{area}(\xi(w))$. Hence we have

$$C_n(q, t) = \sum_w q^{\text{dinv}(w)} t^{\text{prerank}(w)}$$

where w runs of Dyck words of size n . There is also a relation between the maps ζ and ξ :

$$\zeta \circ \xi = R \circ \zeta$$

where $R(w)$ classically maps the Dyck word w to its reverse word where each occurrence of a is replaced by b and conversely.

2 A first greedy algorithm

In our preliminaries, we recall classical properties of the abelian sandpile model and Baker's and, Norine's definition of the rank parameter. We then prove the correctness of our first version of a greedy algorithm computing the rank for a configuration in a complete graph.

2.1 Preliminaries

Let G be a non-oriented connected graph without loops and where multiple edges are allowed. We denote by $V = \{1, 2, \dots, n\}$ where $n = |V|$ the vertex set of G and E its edge

set. The number of edges of E between the vertices i and j is denoted by $e_{i,j}$. Hence the *degree* d_i of the vertex $i \in V$ is $\sum_{j \in V} e_{i,j}$. A *configuration* $u = (u_i)_{i \in V}$ is a vector of \mathbb{Z}^n : the possibly negative number of grains u_i is placed on vertex i in configuration u . Since a configuration u is a vector we can add or subtract configurations: the configuration $w = u - v$ is defined by $w_i = u_i - v_i$ for any vertex i . The *degree of the configuration* u is $\text{degree}(u) := \sum_{i \in V} u_i$. In the abelian sandpile model, the *toppling of the vertex* i in configuration u consists in sending from the vertex i one unit along each incident edge to the opposite endpoint of the edge. These amounts of one unit are also called *grains*. This leads to the configuration $u' = u - \Delta^{(i)}$ where $\Delta_i^{(i)} := d_i$ and $\Delta_j^{(i)} := -e_{i,j}$ otherwise. This toppling is *legal* if $u_i \geq d_i$ ensuring that $u'_i \geq 0$ and $u'_j \geq u_j$ for any $j \neq i$. In this paper non legal topplings are sometimes allowed. The toppling of vertex i preserves the degree of any configuration since $\text{degree}(\Delta^{(i)}) = 0$. Two configurations u and v are *toppling equivalent* if they differ by a finite combination of topplings:

$$u \equiv_{\Delta} v := \exists (a_i)_{i \in V} \in \mathbb{N}^n, u = v - \sum_{i \in V} a_i \Delta^{(i)}$$

The symmetry of this equivalence relation comes from the relation $\sum_{i \in V} \Delta^{(i)} = 0$ which allows to use $-\Delta^{(i)} = \sum_{j \neq i} \Delta^{(j)}$ to change signs in the combination.

We mainly summarize with our notation the definition of the rank presented by Baker and Norine in [6]. A configuration u is *positive*, also denoted $u \geq 0$, if $u_i \geq 0$ for any vertex i . A configuration v is *effective* if v is toppling equivalent to a positive configuration: $\exists u, v \equiv_{\Delta} u$, and $u \geq 0$. The *rank* of the configuration u is defined by the following optimization:

$$\text{rank}(u) := -1 + \min_f \text{degree}(f)$$

where f runs over all positive configurations such that $u - f$ is non-effective. An optimal choice f for the computation of $\text{rank}(u)$ is called a *proof* for the rank of u . The set of such proofs is denoted by $\text{Proofs}(u)$.

The test that a configuration u is effective is a first difficulty in this description of the rank since naively it may require to run over the infinite set of all configurations toppling equivalent to u . Hopefully, one can limit the test to a canonical configuration in the class of toppling equivalence: the n -parking configuration of u denoted $\text{park}(u)$ which will be defined below.¹

Lemma 1 ([6], see also Proposition 37 in annex). *We have u effective if and only if $\text{park}(u)$ is positive.*

The definition of these n -parking configurations requires additional notions. By convention the vertex n is distinguished and called the *sink*. A vertex i in configuration u is *unstable* if $u_i \geq d_i$ where d_i is the degree of the vertex i in G . A configuration u is *stable outside the sink* if any vertex $i \neq n$ is stable. A configuration u is *positive outside the sink* if for any vertex $i \neq n$, $u_i \geq 0$. This is also denoted by $u \geq_{\neq n} 0$. A subset of vertices

¹It corresponds to the notion of n -reduced divisor.

A is *set-unstable* in a configuration u if A is non-empty, $n \notin A$ and, in the configuration $u' = u - \sum_{a \in A} \Delta^{(a)}$, we have $u'_a \geq 0$ for any $a \in A$. In other words, A is set-unstable if the toppling of all the vertices of A leads to a configuration whose restriction to A is positive. If A is a singleton $\{i\}$, A is set-unstable if and only if i is unstable and hence the toppling of i is legal. A configuration u is *set-stable* if there is no subset of vertices set-unstable in u . A configuration u is *n -parking*, also shorten in *parking*, if u is stable and positive outside the sink and set-stable. We refer to the literature [15] for a proof that any configuration u is toppling equivalent to exactly one n -parking configuration. This property implies that the parking configuration $\text{park}(u)$ toppling equivalent to the configuration u is well-defined. There exists many algorithms to compute the map $u \rightarrow \text{park}(u)$ on any graph. A recent one is due to Baker and Shokrieh [7]. We will use the existence of such an algorithm in this section but in Section 3 we will design a most efficient one in the case of the complete graphs.

The *stabilization process* of the configuration u consists in toppling one unstable vertex distinct from the sink n while there exists at least one. A key classical property of the abelian sandpile model [15] is that this stabilization process terminates on a configuration denoted $\text{stabilize}(u)$ which is stable outside the sink and do not depend on the hence unspecified order in which the unstable vertices are toppled.

2.2 A first greedy algorithm computing the rank on complete graphs

From now on we assume that G is a complete graph K_n with $n \geq 2$. Let $\epsilon^{(i)}$ be the configuration with $\epsilon_i^{(i)} = 1$ and for $j \neq i$, $\epsilon_j^{(i)} = 0$. We consider the following greedy algorithm where the variable g is a configuration on K_n :

First greedy algorithm for the rank:

Input: A configuration u on K_n

$u \leftarrow \text{park}(u)$; $\text{rank} \leftarrow -1$; $g \leftarrow 0$;

While $u_n \geq 0$ do let $i \neq n$ such that $u_i = 0$ in $u \leftarrow \text{park}(u - \epsilon^{(i)})$; $g \leftarrow g + \epsilon^{(i)}$; od;

Output: rank ($= \text{rank}(u)$) and g ($\in \text{Proofs}(u)$).

Proposition 2 (Correctness of the rank algorithm). *Given any configuration u on K_n , the greedy naive algorithm for the rank terminates and returns $\text{rank}(u)$ in the variable rank and the configuration variable $g \in \text{Proofs}(u)$ is a proof of this rank.*

The postponed proof of this proposition requires two main proofs: one for the existence of some vertex $i \neq n$ such that $u_i = 0$ in any parking configuration and another one that g is a proof for $\text{rank}(u)$. The second proof relies on the combination of the two following lemmas: the first is related on symmetries of the complete graph K_n while the other, more general, is probably folklore.

Lemma 3 (Greedy choice for rank's proof on K_n). *Let u be a positive configuration on the complete graph K_n such that there exists a vertex i for which $u_i = 0$. Then there exists a proof $g \in \text{Proofs}(u)$ for the rank of u such that $g_i > 0$.*

Lemma 4 (Subproof for the rank). *Let u be a configuration on any graph G and i a vertex of G . Assume there is a proof g for the rank of u such that $g_i > 0$. Then any proof f' for the rank of $u' = u - \epsilon^{(i)}$ leads to a proof $f = f' + \epsilon^{(i)}$ of the rank of u .*

The possibility in our algorithm of a greedy choice for finding a proof for the rank comes from the numerous symmetries of the complete graph. Let σ be a generic permutation of the set S_n of permutations on elements of $\{1, 2, \dots, n\}$. The action of σ on a configuration $u = (u_i)_{i=1..n}$ is defined by $\sigma.u := (u_{\sigma(i)})_{i=1..n}$. We gather in the following lemma properties related to symmetries used in this section but also later:

Lemma 5 (Symetries on K_n and the rank). *Let u and v be any configuration on K_n and σ any permutation of S_n . We have :*

1. u positive $\iff \sigma.u$ positive,
2. $u \equiv_{\Delta} v \iff \sigma.u \equiv_{\Delta} \sigma.v$,
3. u effective $\iff \sigma.u$ effective,
4. $g \in \text{Proofs}(u) \iff \sigma.g \in \text{Proofs}(\sigma.u)$,
5. $\text{rank}(\sigma.u) = \text{rank}(u)$,
6. If $\sigma(n) = n$ (or with a slight abuse of notation $\sigma \in S_{n-1} \subset S_n$) then:
 - (a) u positive outside the sink $\iff \sigma.u$ positive outside the sink,
 - (b) u set-stable $\iff \sigma.u$ set-stable,
 - (c) u parking $\iff \sigma.u$ parking.
 - (d) $\text{park}(\sigma.u) = \text{park}(u)$

Now we listed the required lemmas to prove the correctness of the algorithm, we use another schedule to prove them.

Proof. (of Lemma 5: Symmetries on K_n and the rank)

(1) is straightforward. We remark that σ acts also nicely on the configurations $\Delta^{(i)}$, hence $u = v - \sum_i a_i \Delta^{(i)} \iff \sigma.u = \sigma.v - \sum_i a_i \Delta^{(\sigma(i))}$. This remark leads to (2); (1) and (2) imply (3).

The following paragraph proves (4). If $g \in \text{Proofs}(u)$ then $u - g$ is non-effective and for any positive configuration f such that $\text{degree}(f) < \text{degree}(g)$, $u - f$ is effective. From (3) we deduce that $\sigma.u - \sigma.g$ is non-effective and for any positive configuration f such that $\text{degree}(f) < \text{degree}(g)$, $\sigma.u - \sigma.f$ is effective. Since $\{f | f \geq 0, \text{degree}(f) < \text{degree}(g)\} = \{\sigma.f | \sigma.f \geq 0, \text{degree}(\sigma.f) < \text{degree}(\sigma.g)\}$ we deduce that $\sigma.g \in \text{Proofs}(\sigma.u)$. Conversely, for the opposite implication of (4) we use similarly σ^{-1} , the inverse permutation of σ .

(4) implies (5). The proof of (6.(a)) is similar to the proof (1) and the proof of (6.(b)) to the proof of (2): the single additional remark is that the sink vertex n is simultaneously excluded from the set-unstable set of vertices and a fix-point of $\sigma \in S_{n-1} \subset S_n$. (6.(a))

and (6.(b)) imply (6.(c)). We have $\text{park}(u) = u - \sum_i a_i \Delta^{(i)}$ hence like in the proof of (2), $\sigma.\text{park}(u) = \sigma.u - \sum_i a_i \Delta^{(\sigma(i))}$ where $\sigma.\text{park}(u)$ is parking according to (6.(c)) and toppling equivalent to $\sigma.u$ hence $\text{park}(\sigma.u) = \sigma.\text{park}(u)$. \square

Proof. (of Lemma 3 Greedy choice for rank's proof on K_n)

Let $f \in \text{Proofs}(u)$. If $f_i > 0$ then $g = f$ concludes the proof. In the remaining case, we have $f_i = 0$. Since f is a proof, $u - f$ is non-effective hence there exists a vertex j such that $u_j - f_j < 0$. We have $j \neq i$ since $u_i - f_i = 0 - 0$. We will show that $g = f + (f_j - u_j)(\epsilon^{(i)} - \epsilon^{(j)})$ is the expected proof for the rank of u . By definition, $g_i = f_j - u_j > 0$ and more generally g is positive and $\text{degree}(g) = \text{degree}(f)$. Hence it simply remains to show that $u - g$ is non-effective. We consider $u' = u - f + (f_j - u_j)\epsilon^{(j)}$. Let $\sigma = (ij)$ the transposition of S_n which exchanges i and j : $\sigma(i) = j$, $\sigma(j) = i$ and otherwise $\sigma(k) = k$. By definition $u'_j = u_j - f_j + (f_j - u_j) = 0$ and $u'_i = 0 - 0 = 0$, so $\sigma.u' = u'$. Hence:

$$\sigma.(u - g) = \sigma.(u - f - (f_j - u_j)(\epsilon^{(i)} - \epsilon^{(j)})) = \sigma.(u' - (f_j - u_j)\epsilon^{(i)}) = u' - (f_j - u_j)\epsilon^{(j)} = u - f.$$

Using $\sigma.(u - g) = u - f$ and (3) in lemma 5 we deduce as expected that like $u - f$, $u - g$ is non-effective. \square

Proof. (of Lemma 4 Subproof for the rank) Let g a proof of u such that $g_i > 0$ and f' a proof of $u' = u - \epsilon^{(i)}$. On one hand, $u' - f' = u - (f' + \epsilon^{(i)})$ is non-effective so by definition of g , $\text{degree}(f') + 1 \geq \text{degree}(g)$. On the other hand, $u' - \epsilon^{(i)} = u - g$ is non-effective and $g - \epsilon^{(i)}$ is positive so by definition of f' $\text{deg}(g) - 1 \geq \text{deg}(f')$. Joining these hands, $\text{degree}(f') = \text{degree}(g) - 1$. Since $f = f' + \epsilon^{(i)}$ is of degree $\text{degree}(g)$ and $u - f = u' - f'$ is non effective, f is a proof for the rank of u . \square

Proof. (of Proposition 2 Correctness of the rank algorithm)

First, if u is a parking configuration on K_n there exists a vertex i distinct from the sink such that $u_i = 0$. Indeed we obtain a contradiction if for any $i \neq n$, $u_i \neq 0$. Since u is positive outside of the sink, it means that $u_i \geq 1$. Hence $A = \{1, 2, \dots, n - 1\} = V - \{n\}$ is a set-unstable set. This is in contradiction with the assumption that u is parking.

We conclude the proof of correctness by an induction on the rank of the given configuration u . If $\text{rank}(u) = -1$, the configuration u is non-effective so, according to Lemma 1, $\text{park}(u)$ has a negative value on the sink then the algorithm skips the while-loop and returns $\text{rank} = -1$ and $g = 0$ as expected. Let $m \geq 0$. We assume that the algorithm is correct on any given configuration of rank at most $m - 1$. Let u be a configuration of rank m , $v = \text{park}(u)$ and i the first chosen vertex by the algorithm such that $v_i = 0$. Lemma 3 on the greedy choice for the rank implies the existence of a proof g for the rank of u such that $g_i > 0$ and so $g - \epsilon^{(i)}$ is positive. Since $v - g = (v - \epsilon^{(i)}) - (g - \epsilon^{(i)})$ is non-effective, the rank of $v - \epsilon^{(i)}$ is at most $m - 1$. Our inductive assumption implies that the algorithm applied to $v - \epsilon^{(i)}$ leads to a proof f of its rank. Applying Lemma 4 about subproofs for the rank, the configuration $f + \epsilon^{(i)}$ is a proof for the rank of the given u . This is also the value in the variable g at the end of the run given the configuration u . \square

3 An algorithm of linear arithmetic complexity

In this section we optimize the correct greedy naive algorithm. Subsection 3.1 explains how to restrict quickly the algorithms to the manipulation of configurations soon defined and called compact and sorted. Subsection 3.2 motivates this restriction by a bijection between these compact and sorted configurations and pointed cut skew cylinders. This second class of combinatorial objects may be encoded by a Dyck word and one up to three integers. In the algorithms computing the rank or the parking configurations, atomic updates or tests on configurations admits more regular interpretations in terms of pointed cut skew cylinders. Subsection 3.3 translates a possible run of the algorithm for the rank into a simple partial spiral traversal in the cut skew cylinder. The analysis of this traversal leads to an explicit formula for the rank of a sorted and parking configuration based on a single euclidean division. Subsection 3.4 describes an algorithm adapted to the complete graph to find the parking configuration toppling equivalent to any given configuration. We conclude this section by discussing the details of a possible implementation in linear arithmetic complexity of an algorithm computing the rank of any configuration on K_n .

3.1 Restriction to compact and sorted configurations

The distinguished sink n on K_n implies that we frequently manipulate configurations where the sink is the only particular case. We often abbreviate a configuration $u = (u_1, \dots, u_n)$ into $(u_i; u_n)$ which means implicitly that the variable i , part of the notation, runs from 1 to $n - 1$. Our next definition illustrates this notation.

The *compacted* configuration u^c of u is the configuration

$$\text{compact}(u) := u^c := ((u_i - u_n) \pmod n; \text{degree}(u) - \sum_{i=1}^{n-1} u_i^c).$$

In other words, for $i = 1 \dots n - 1$, $u_i^c := (u_i - u_n) \pmod n$ and for $i = n$, $u_n^c := \text{degree}(u) - \sum_{i=1}^{n-1} u_i^c$.

Lemma 6. (*Compact configuration on K_n*) For any configuration u on K_n , we have $\text{compact}(u) \equiv_{\Delta} u$.

Proof. On one hand, we topple u_n times each vertex different from the sink in the configuration u to obtain

$$u \equiv_{\Delta} (u_i; u_n) - \sum_{j=1}^{n-1} u_n \Delta^{(j)} = (u_i - u_n; nu_n).$$

On the other hand, by definition of $u^c := \text{compact}(u)$ there exists $(k_1, \dots, k_{n-1}) \in \mathbb{Z}^n$ such that for $i \neq n$, $u_i - u_n = u_i^c + k_i n$. Joining the hands, $u \equiv_{\Delta} (u_i^c + k_i n; nu_n)$. Then we consider

$$(u_i^c + k_i n; nu_n) - \sum_{j \neq n} k_j (\Delta^{(j)} - \Delta^{(n)}) = (u_i^c; nu_n - n \sum_{j \neq n} k_j) = (u_i^c; \text{degree}(u) - \sum_{j \neq n} u_j^c)$$

where $k_i(\Delta^{(i)} - \Delta^{(n)})$ cancels $k_i n$ in $u_i^c + k_i n$ and $\text{degree}(u)$ is introduced using that topplings preserve the degree of configurations. The rightmost term is the preceding sequence of equalities is $\text{compact}(u)$ which is hence proved to be toppling equivalent to u . \square

This definition of $\text{compact}(u)$ particular to the complete graphs is inspired by a work on general graphs of Dhar, Ruelle and, Verma [18]. One find similar ideas also in [7]. We observe that $\text{compact}(u)$ can be quickly computed, it is positive and stable outside of the sink but not necessarily equal to $\text{stabilize}(u)$.

A configuration u on K_n is *compact* if

$$\max_{i \neq n, j \neq n} |u_i - u_j| \leq n.$$

It is straightforward that the configuration $\text{compact}(u)$ is compact. Moreover, most of the configurations appearing in the optimized algorithms are compacts.

Another aspect of the optimization consists in working at the level of orbits under the action of $S_{n-1} \subset S_n$ permuting the vertices $1, 2, \dots, n-1$ distinct from the sink. The representative of the orbit of u will be the defined below $\text{sort}(u)$ sorted configuration. A configuration u is *sorted* if u_1, \dots, u_{n-1} is weakly increasing: for $i = 2 \dots n-1$, $u_{i-1} \leq u_i$. Notice that there is no constraint on u_n . The configuration $\text{sort}(u)$ is obtained by sorting in weakly increasing order of the first $n-1$ entries of the configuration u .

We also combine the action of S_{n-1} and the toppling equivalence. Two configurations u and v on K_n are *toppling and permuting equivalent*, denoted $u \equiv_{\Delta, S_{n-1}} v$ if there exists a permutation $\sigma \in S_{n-1} \subset S_n$ such that u and $\sigma.v$ are toppling equivalent.

Lemma 7. (*compatibilities for toppling and permuting equivalence*)

- (1). *The relation $\equiv_{\Delta, S_{n-1}}$ is an equivalence relation.*
- (2). *Moreover, for any configuration u , there exists a unique sorted and parking configuration $v := \text{sort}(\text{park}(u))$ toppling and permuting equivalent to u .*

Proof. (1). The relation is symmetric mainly according to the second of the following equivalences:

$$\begin{aligned} u \equiv_{\Delta, S_{n-1}} v &\iff \exists \sigma \in S_{n-1}, \exists (a_i)_i \in \mathbb{Z}, \sigma.u = v - \sum_{i=1}^n a_i \Delta^{(i)} \\ &\iff \exists \sigma \in S_{n-1}, \exists (a_i)_i \in \mathbb{Z}, \sigma^{-1}.v = u - \sum_{i=1}^n (-a_{\sigma^{-1}(i)}) \Delta^{(\sigma^{-1}(i))} \\ &\iff v \equiv_{\Delta, S_{n-1}} u \end{aligned}$$

The transitivity of the relation comes from the composition of permutations. Let $\sigma.u = v - \sum_{i=1}^n a_i \Delta^{(i)}$ and $\tau.v = w - \sum_{i=1}^n b_i \Delta^{(i)}$ then

$$(\tau \circ \sigma).u = w - \sum_{i=1}^n (b_i - a_{\tau^{-1}(i)}) \Delta^{(i)}.$$

(2). The existence of a sorted parking configuration toppling and permuting equivalent to u is straightforward since by definition $\text{sort}(\text{park}(u))$ is such a configuration.

Our proof of the uniqueness considers two configurations u and v toppling and permuting equivalent. By assumption, there is a permutation $\sigma \in S_{n-1}$ such that $\sigma.u$ is toppling equivalent to v . The uniqueness of the parking configuration in a toppling equivalent class implies that $\text{park}(\sigma.u) = \text{park}(v)$. Using the identity 6.(d) of Lemma 5 on symmetries of K_n , we deduce that $\sigma.\text{park}(u) = \text{park}(v)$ hence $\text{sort}(\text{park}(u)) = \text{sort}(\text{park}(v))$ as expected. \square

3.2 From sorted and compact configurations to pointed cut skew cylinders

The parking configurations on the complete graph K_n admits a more explicit description which explains the adjective parking. A *parking function* of size n is a map f from $\{1, 2, \dots, n\}$ to \mathbb{Z} such that for $i = 1 \dots n-1$, $f(i) \geq 0$ and for $j = 1 \dots n$, $|\{i | f(i) \geq j\}| \leq n - j$. It was already noticed in [13] that a configuration u on K_n is parking if and only if f defined by $(f(i))_{i=1 \dots n-1} := (u_i)_{i=1 \dots n-1}$ is a parking function of size $n-1$. We already observed that u is parking if and only if $\sigma.u$ where $\sigma \in S_{n-1} \subset S_n$, in particular $\text{sort}(u)$, is parking. Sorting simplifies the checking: a sorted configuration u is parking if and only if for $i = 1 \dots n-1$, $0 \leq u_i < i$.

Until now, we manipulated the configurations as vectors in \mathbb{Z}^n . We introduce an alternative description of sorted and compact configurations on K_n which will simplify our analysis. This description requires some (painful but then rewarding) preliminaries.

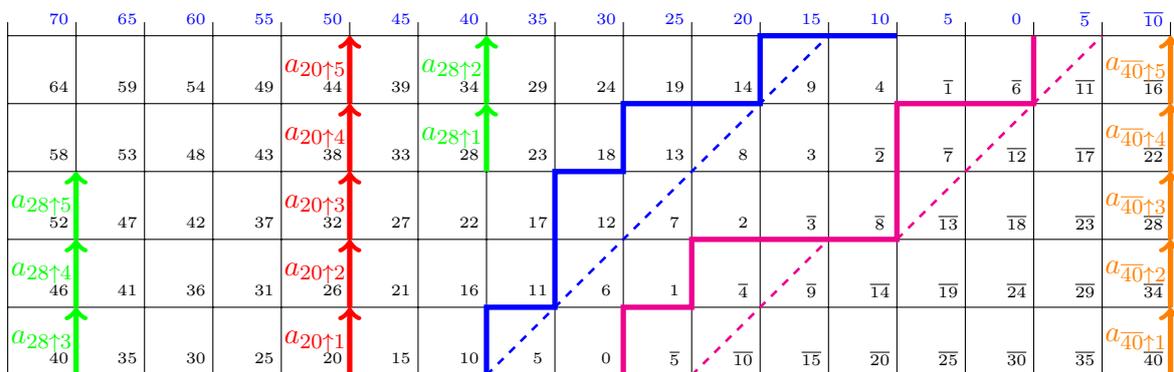


Figure 1: Skew cylinder of circumference 6, two of its cuts and three oriented segments.

The *skew cylinder* of circumference n is a kind of skewed two dimensional grid embedded in a cylinder. A first precise definition considers the quotient of the usual grid of \mathbb{Z}^2 where each vertex of integer coordinates (x, y) admits the four neighbours $\{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\}$ by the equivalent relation $(x, y) \equiv (x', y')$ if there exists $k \in \mathbb{Z}$ such that $(x', y') = (x + kn, y + k(n-1))$. A possible notation for this is $\mathbb{Z}^2 / (n, n-1)\mathbb{Z}$. We fix our notation on a second precise and more explicit definition of skew cylinder which has the drawback to partially hide the invariance by translation. This

definition is illustrated by the example in Figure 1. The *strip* of $n - 1$ rows is the subgraph of the usual two dimensional grid \mathbb{Z}^2 induced by the vertices $\{(x, y) | x \in \mathbb{Z}, 0 \leq y \leq n - 1\}$. The two-dimensional of *cell* coordinates $(x, y) \in \mathbb{Z}^2$ is the unit square which corners are $\{(x, y), (x - 1, y), (x - 1, y + 1), (x, y + 1)\}$. In other words the two-dimensional coordinates of a cell are the coordinates of its south-east corner. For a cell c we denote by $(x(c), y(c))$ its two-dimensional coordinates. We will use the *spiral coordinate* as an alternative coordinate for the cells of the strip. The cell of two dimensional coordinate $(0, 0)$ is labeled by the spiral coordinate $0 \in \mathbb{Z}$, then we label the cells, incrementing the spiral coordinate, from south-west to north-east in the cells along the diagonal in north-east direction, then the next diagonal is the next on the west. This *spiral traversal* labels all the cells of spiral coordinate $k \in \mathbb{N}$. In Figure 1, the spiral coordinate of each cell is indicated by the small number written in the south-west corner. Negative coordinate $-x$ is written \bar{x} for graphical purpose. This number is blue for partial cells in the top row and means that these are repetitions of the cells on the bottom row. In the opposite direction and starting from the cell $(0, 0)$, the reversed spiral traversal labels all the cells with a non-positive spiral coordinate. Hence, this spiral traversal defines a bijection between the cells of the strip of $n - 1$ rows and \mathbb{Z} . We frequently use the cell (of spiral coordinate) $k \in \mathbb{Z}$ and the vertex (of spiral coordinate) k which is the south-east corner of this cell k . A more operational description is that the cell of two dimensional coordinates (x, y) in the strip has spiral coordinate $ny - (n - 1)x$ and the cell of spiral coordinate $k \in \mathbb{Z}$ has two dimensional coordinates $(r - q, r)$ where $k = q(n - 1) + r$ is the euclidean division of k by $n - 1$. We turn this strip into the skew cylinder of circumference n by an identification of the vertices of coordinates $(x, 0)$ and $(x + n, n - 1)$ which implies that the south side of the cell $(x, 0)$ is also the north side of the cell $(x + n - 1, n - 2)$.

In a word w , we denote by $|w|_x$ the number of occurrences of the letter x . An *almost balanced word* of size n is a word on the alphabet $\{a, b\}$ such that $|w|_a = n - 1$ and $|w|_b = n$. Let D_n denotes the set of almost balanced words of size n .

Given a vertex $k \in \mathbb{Z}$ in the skew cylinder of circumference n and a word w on the two-letter alphabet $\{a, b\}$, the *embedding of the word w at vertex k* is the *path* starting from vertex k and such that an occurrence of letter a in w corresponds to a unit north step and an occurrence of letter b to a unit east step. The *cell of a north step* in such a path is the cell whose east side is the north step.

We first observe that the embedding at vertex $m \in \mathbb{Z}$ of an almost balanced word $w \in D_n$ in the skew cylinder of circumference n also ends at vertex m . This defines a (self-avoiding) loop that we call a *cut*. The resulting *cut skew cylinder* is denoted by $\text{cylinder}[m \circlearrowleft w]$. In Figure 1, the blue path is a cut defining $\text{cylinder}[10 \circlearrowleft abaababbabb]$ and the magenta path is a cut defining $\text{cylinder}[2 \circlearrowleft bbaabbaabab] = \text{cylinder}[-3 \circlearrowleft baabbaababb]$.

Definition 8. A cut skew cylinder $\text{cylinder}[m \circlearrowleft w]$ and two additional integers $k \in \mathbb{Z}$ and $s \in \mathbb{Z}$ allow the definition of a configuration on K_n

$$\text{scompact}[k \Rightarrow m \circlearrowleft w; s] := (u_i; u_n) = (x(a_i) - x(a_{k \uparrow i}); s)$$

where the $a_{k \uparrow i}$ for $i = 1 \dots n - 1$ are the $n - 1$ north steps of the *oriented segment* which

is the embedding of the word a^{n-1} at vertex k , and a_i is the north step of the cut on cylinder $[m \circlearrowleft w]$ which crosses the same row as the north step $a_{k \uparrow i}$ and $x(\cdot)$ is the first coordinate of the two-dimensional coordinates of the starting vertex of a step.

In Figure 1, there are three oriented segments: the red one starts from 20, the orange one from -40 and the green one from 28. The last one is drawn into two green lines since it visits the two copies of the vertex 40. The red oriented segment, the blue cut and the integer 39 define the configuration

$$\begin{aligned} \text{scompact}[20 \Rightarrow 10 \circlearrowleft abaababbabb; 39] &= (2, 3, 3, 4, 6; 39) \\ &= (x(10) - x(20), x(11) - x(26), x(17) - x(32), x(18) - x(38), x(14) - x(44); 39). \end{aligned}$$

The properties of the description $\text{scompact}[k \Rightarrow m \circlearrowleft w; s]$ require additional preliminary definitions.

Two cells are connected in a cut skew cylinder if they share a common side which is not a step of the cut. In the cut skew cylinder $\text{cylinder}[m \circlearrowleft w]$ there are two connected components of cells: the *left component* $L[m \circlearrowleft w]$, which contains the cell m , and the other called the *right component* $R[m \circlearrowleft w]$. A cell is *left* if it belongs to $L[m \circlearrowleft w]$ and *right* otherwise.

A *Dyck word* of size n is a word w on the two-letters alphabet $A = \{a, b\}$ where for any prefix p of w , $|p|_a \geq |p|_b$ and in addition $|w|_a = |w|_b = n$. Any factorization $w = w'w''$ defines a cyclic conjugate $v = w''w'$ of the word w . The classical cyclic lemma due to Dvoretzky and Motzkin [20] states that among the $2n - 1$ distinct cyclic conjugates of an almost balanced word $w \in D_n$ exactly one may be written vb where v is a Dyck word of size $n - 1$. We recover this lemma in our setting. In Figure 1, a dashed line indicates the Dyck factor $abaababbab$ respectively $aabbaababb$ in the cyclic conjugate for the blue, respectively magenta, cut. A first indication is that pairs (w, s) where w is a Dyck word of size $n - 1$ and $s \in \mathbb{Z}$ are in classical bijection with sorted parking configurations on K_n as follows.

$$(u_i; u_n) = \left(\sum_{j=1}^i k_j; s \right) \leftrightarrow (w, s) = (ab^{u_2-u_1} ab^{u_3-u_2} \dots ab^{u_{n-1}-u_{n-2}} ab^{n-1-u_{n-1}}, u_n)$$

where² $w = b^{k_1} ab^{k_2} ab^{k_3} \dots ab^{k_n}$. This bijection translates the fact that in a sorted parking configuration u we have $0 \leq u_i < i$ for $i = 1 \dots n - 1$ into the fact that w is a Dyck word. Another classical notion for a non-empty Dyck word w is its non-ambiguous *decomposition at first return* $w = afbg$ where f and g are both possibly empty Dyck words.

We define the map T on sorted configurations on K_n :

$$T.u := \text{sort}(u - \Delta^{(n-1)})$$

and the map E on sorted *parking* configurations on K_n which corresponds to the subtraction of one unit on a vertex with value 0 in the configuration:

$$E.u := u - \epsilon^{(1)}.$$

²Remark that $k_1 = 0$ and $u_1 = 0$

Proposition 9. (From sorted compact configurations to pointed cut skew cylinders)

1. A configuration u on K_n is sorted and compact if and only if there exists a word $w \in D_n$ and three integers $(k, m, s) \in \mathbb{Z}^3$ such that $u = \text{scompact}[k \Rightarrow m \circlearrowleft w; s]$.

2. For any factorization $w = w'w''$ of $w \in D_n$ we have

$$\text{cylinder}[m \circlearrowleft w'w''] = \text{cylinder}[m + n|w'|_a - (n - 1)|w'|_b \circlearrowleft w''w'].$$

3. For any $p \in \mathbb{Z}$, we have

$$\text{scompact}[k \Rightarrow m \circlearrowleft w; s] = \text{scompact}[k + p \Rightarrow m + p \circlearrowleft w; s].$$

4. The map $(v, k, s) \rightarrow \text{scompact}[k \Rightarrow 0 \circlearrowleft vb; s]$ defines a bijection between the triplets made up of a Dyck word v of size $n - 1$ and two integers $(k, s) \in \mathbb{Z}^2$ and the sorted and compact configurations on K_n .

Moreover, the sorted and compact configurations toppling and permuting equivalent to $\text{scompact}[k \Rightarrow 0 \circlearrowleft vb; s]$ are exactly the configurations $(\text{scompact}[k + r \Rightarrow 0 \circlearrowleft vb; s - r])_{r \in \mathbb{Z}}$.

5. The restriction of T to sorted and compact configurations is well-defined and reversible. Moreover we have

$$T.\text{scompact}[k \Rightarrow m \circlearrowleft w; s] = \text{scompact}[k - 1 \Rightarrow m \circlearrowleft w; s + 1]$$

and for any sorted and compact configuration u on K_n we have

$$T^{1-n}.u = u - \Delta^{(n)}.$$

6. The configuration $\text{scompact}[k \Rightarrow m \circlearrowleft w; s]$ is parking if and only if $k = \min L[m \circlearrowleft w]$ where we use the spiral coordinate for cells of $L[m \circlearrowleft w]$. Moreover

$$\begin{aligned} \text{park}(\text{scompact}[k \Rightarrow m \circlearrowleft w; s]) &= T^{k - \min L[m \circlearrowleft w]}.\text{scompact}[k \Rightarrow m \circlearrowleft w; s] \\ &= \text{scompact}[\min L[m \circlearrowleft w] \Rightarrow m \circlearrowleft w; s + k - \min L[m \circlearrowleft w]]. \end{aligned}$$

7. The map $(v, s) \rightarrow \text{scompact}[0 \Rightarrow 0 \circlearrowleft vb; s]$ defines a bijection between the pairs made up of a Dyck word v of size $n - 1$ and one integer $s \in \mathbb{Z}$ and the sorted and parking configurations on K_n .

8. Let $u = \text{scompact}[m \Rightarrow m \circlearrowleft vb; s]$ and $v = afbg$ be the factorization at first return of the Dyck word v and $j = \min(L[m \circlearrowleft vb] - \{m\})$ the second smallest spiral coordinate of a left cell, we have

$$\text{sort}(\text{park}(E.u)) = \text{scompact}[j \Rightarrow j \circlearrowleft gabfb; s - (j - m)]$$

where $gabf$ is a Dyck word and moreover

$$L[j \circlearrowleft gabfb] = L[m \circlearrowleft vb] - \{m\}.$$

Proof. 1. If $u = (u_i; u_n)$ is a sorted and compact configuration, let $m = 0$, $k = (n - 1)u_1$, $s = u_n$ and $w = ab^{u_2 - u_1} ab^{u_3 - u_2} ab^{u_4 - u_3} \dots ab^{u_{n-1} - u_{n-2}} ab^{n - (u_{n-1} - u_1)} \in D_n$. By definition, we have $u = \text{scompact}[k \Rightarrow m \circlearrowleft w; s]$. Conversely, given $w \in D_n$ and $(m, k, s) \in \mathbb{Z}^3$. Let $k = q(n - 1) + r$ be the euclidean division of k by $n - 1$ and factorize w as $w = b^{k_1} ab^{k_2} ab^{k_3} \dots ab^{k_n}$. By definition we have $u = \text{scompact}[k \Rightarrow m \circlearrowleft w; s]$ which is also defined by $u_n = s$ and $u_i = q + \sum_{j=1}^i k_j$. Hence u is sorted and $u_{n-1} - u_1 = \sum_{j=2}^{n-1} k_j \leq n$ so u is also compact.

2. Moving the start of the cut along the first step does not modify the cut skew cylinder. This remark leads to either $\text{cylinder}[m \circlearrowleft aw'] = \text{cylinder}[m + n \circlearrowleft w'a]$ or $\text{cylinder}[m \circlearrowleft bw'] = \text{cylinder}[m - (n - 1) \circlearrowleft w'b]$. Iterations of these two rules leads to the expected claim.

3. Two equal translations of the cut and the oriented segment preserve the pairs $(a_i, a_{k \uparrow i})$ and their differences $x(a_i) - x(a_{k \uparrow i})$ hence proves the claim.

5. (see 4. below) Let u be a sorted and compact configuration. First we compute explicitly

$$\begin{aligned} v := T.u &= \text{sort}((u_1 + 1, u_2 + 1, \dots, u_{n-2} + 1, u_{n-1} - (n - 1); u_n + 1)) \\ &= (u_{n-1} - (n - 1), u_1 + 1, u_2 + 1, \dots, u_{n-2} + 1; u_n + 1) \end{aligned}$$

where sort induces the cyclic permutation τ of the $n - 1$ first entry sending $n - 1$ to 1, since $u_{n-1} - (n - 1) \leq u_1 + 1$ because u is compact. Moreover $v_{n-1} - v_1 = u_{n-2} + 1 - u_{n-1} + (n - 1) \leq n$ so v is also compact. The inverse of the cyclic permutation τ is well defined and also the inverse of the toppling of $n - 1$ which corresponds to the toppling of all vertices distinct from $n - 1$. The reader may then check that the inverse T^{-1} of T is the expected composition of these two inverses, *ie* $T^{-1}.v = u$.

According to 1., let (k, m, w, s) such that $u = \text{scompact}[k \Rightarrow m \circlearrowleft w; s]$. Inspection shows that $\text{scompact}[k - 1 \Rightarrow m \circlearrowleft w; s + 1] = v = T.u$.

During the computation of $T^{1-n} = (T^{-1})^{n-1}$, the fact that τ^{-1} is cyclic implies that each vertex distinct from the sink n is toppled $n - 2$ times and the sink n is toppled $n - 1$ times. Since toppling $n - 2$ times each vertex does not change the configuration and τ^{1-n} is the identity we deduce that T^{1-n} is equivalent to one toppling of the sink n as claimed.

6. First we suppose that u is parking. The cells at west of the oriented segment $(a_{k \uparrow i})_{i=1 \dots n-1}$ are $(k + n(i - 1))_{i=1 \dots n-1}$. We consider the vector $(l_i)_{i=1 \dots n-1}$ where l_i is the minimal spiral coordinate of a cell in the left component of $L[m \circlearrowleft w]$ on the same row as $a_{k \uparrow i}$. Hence by definition of $\text{scompact}[k \Rightarrow m \circlearrowleft w; s]$, we have for $i \neq n$, $u_i = \frac{k + n(i-1) - l_i}{n-1}$. The configuration u is sorted and parking so we have $u_i < i$. This is equivalent to $\frac{k + n(i-1) - l_i}{n-1} < i \iff l_i > k - 1 \iff l_i \geq k$. Moreover, $u_1 = 0$ so $k = l_1 \in L[m \circlearrowleft w]$ and $k = \min L[m \circlearrowleft w]$.

Conversely, we suppose that $k = \min L[m \circlearrowleft w]$. The cut is made of east or south steps so $k \in L[m \circlearrowleft w]$ implies recursively via the north sides of cells that $k + n(i - 1) \in L[m \circlearrowleft w]$ for $i = 1, \dots, n - 1$. Combined with the backward equivalences of the previous argument we have $k \leq l_i \leq k + n(i - 1)$ and so $0 \leq u_i < i$. So u is a sorted parking configuration.

The power of T in the description of $\text{park}(\text{scompact}[k \Rightarrow m \circlearrowleft w; s])$ is chosen so that, in the resulting configuration, the oriented segment starts at the vertex in the south-east

corner of the cell $\min L[m \circlearrowleft w]$. This choice is correct by uniqueness of the toppling and permuting equivalent sorted and parking configuration.

7. Let $u = \text{scompact}[k \Rightarrow m \circlearrowleft w; s]$ be a parking configuration. According to 6., $k = \min L[m \circlearrowleft w] \in L[m \circlearrowleft w]$. Hence the cell $k - (n - 1)$ sharing the vertical east side of the cell k belongs to $R[m \circlearrowleft w]$. So this vertical side is a step of the cut of $\text{cylinder}[m \circlearrowleft w]$ and according to 2. there exists a cyclic conjugate w' of w and $k' \in \mathbb{Z}$ such that $u = \text{scompact}[k' \Rightarrow k' \circlearrowleft w'; s]$. According to 3., we have $u = \text{scompact}[0 \Rightarrow 0 \circlearrowleft w'; s]$. We have $\min L[0 \circlearrowleft w'] = 0$ so the cells $(-i)_{i=1 \dots n-1}$ belong to $R[0 \circlearrowleft w']$. This implies that $w' = vb$ where v is a Dyck word of size $n - 1$.

Conversely, if $u = \text{scompact}[0 \Rightarrow 0 \circlearrowleft vb; s]$ where v is a Dyck word of size $n - 1$, then $\min L[0 \circlearrowleft vb] = 0$ so u is sorted and parking.

The map $(v, s) \rightarrow \text{scompact}[0 \Rightarrow 0 \circlearrowleft vb; s]$ is obviously injective. Hence this map is the expected bijection.

4. Let u be a sorted and compact configuration and $u' = \text{sort} \circ \text{park}(u)$. According to 6., there exists $m \in \mathbb{Z}$ such that $u = T^m.u'$. According to 8., $u' = \text{scompact}[0 \Rightarrow 0 \circlearrowleft vb; s]$ where v is a Dyck word of size $n - 1$ and $s \in \mathbb{Z}$ and the pair (v, s) is unique. Hence, according to 5., $u = T^m.\text{scompact}[0 \Rightarrow 0 \circlearrowleft vb; s] = \text{scompact}[m \Rightarrow 0 \circlearrowleft vb; s - m]$. The observation of the value on the sink n shows that all sorted and compact configurations $(\text{scompact}[m \Rightarrow 0 \circlearrowleft vb; s - m])_{m \in \mathbb{Z}}$ are distinct and toppling and permuting equivalent to u' and also u .

8. By definition, $m = \min L[m \circlearrowleft afbgb]$. Inspection shows that the explicit occurrence of the letter b in the factor afb corresponds to the south side of the cell j . Hence $j = m + |afb|_a$ and the vertex j belongs to the cut. Changing the start of the cut from m to j , we also have $u = \text{scompact}[m \Rightarrow j \circlearrowleft gbafb; s]$. By definition the occurrence of the letter a in $gbafb$, appears in the definition of u_1 . Hence the commutation of this occurrence of a with the preceding occurrence, which is necessarily a letter b , has two consequences. First, exactly u_1 is decremented by one, leading to the sorted and compact configuration $E.u = \text{scompact}[m \Rightarrow j \circlearrowleft gbafb; s]$. Second, the left cell m in $\text{scompact}[m \Rightarrow j \circlearrowleft gbafb; s]$ becomes a right cell in $\text{scompact}[m \Rightarrow j \circlearrowleft gbafb; s]$ since the cut visiting the south and east side of cell m in the first case, visits the west and north side of this cell after the commutation. As expected, we have $L[j \circlearrowleft gbafb] = L[m \circlearrowleft afbgb] - \{m\}$. Moreover, after the deletion of the smallest value m , the second smallest value j becomes minimal: $\min L[j \circlearrowleft gbafb] = j$. By application of 6., we deduce that $\text{sort}(\text{park}(E.u)) = T^{m-j}.E.u = \text{scompact}[j \Rightarrow j \circlearrowleft gbafb; s - (j - m)]$. \square

3.3 An explicit formula for the rank of sorted parking configurations on K_n

We describe a possible run of our correct first greedy algorithm for the rank presented in Section 2 combined with the action of the symmetric group $S_{n-1} \subset S_n$. In a sorted parking configuration u , we necessarily have $u_1 = 0$ which will be our systematic choice of $i \neq n$ such that $u_i = 0$. Hence a possible run of the correct algorithm on parking configurations in terms of the sorted configurations is

Algorithm selecting a run for the rank on sorted configurations:

Input: A sorted parking configuration u on K_n
 $rank \leftarrow -1$;
 While $u_n \geq 0$ do $u \leftarrow \text{sort}(\text{park}(E.u))$; $rank \leftarrow rank + 1$; od;
Output: $rank(= \text{rank}(u))$

This run is tightly related to the following spiral traversal of some cells of the cut skew cylinder such that $u = \text{scompact}[m \Rightarrow m \circ vb; s]$. We define $S_{[m \circ vb]}(s) = \min\{s' | s' > s \text{ and } s' \in L[vb \circ s]\}$ which is the minimal spiral coordinate strictly greater than s of a left cell in $\text{cylinder}[m \circ vb]$.

Algorithms for the rank via a spiral traversal on a cut skew cylinder:

Input: A sorted parking configuration $u = \text{scompact}[m \Rightarrow m \circ vb; s]$ on K_n .
 $rank \leftarrow -1$;
 For k from m to $F(u)$ do
 ---- if k is a left cell ($k \in L[m \circ vb]$) then $rank \leftarrow rank + 1$; od;
Output: $rank(= \text{rank}(u))$.

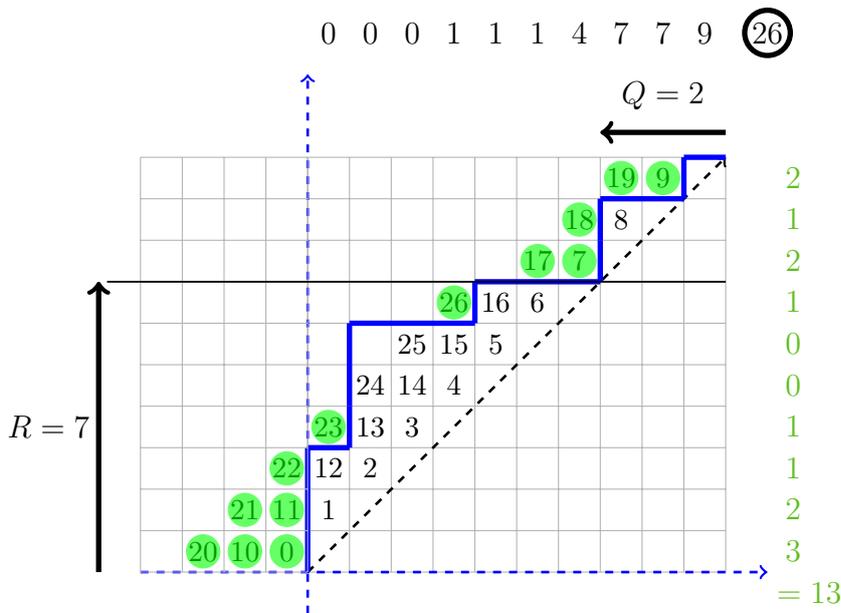


Figure 2: Calculation of the rank on a Dyck path

We use two versions of this algorithm corresponding to the two distinct definitions of $F(u)$. If $F(u) := S_{[m \circ vb]}(m + s) - 1$, it defines the *extended spiral traversal for the rank*. If $F(u) := m + s$, it defines the *shortest spiral traversal for the rank*. In Figure 2, we labeled the cells visited by the shortest spiral traversal for the rank of

$$u = (0, 0, 0, 1, 1, 1, 4, 7, 7, 9; 26) = \text{scompact}[0 \Rightarrow 0 \circ aaabaaabbbabbbbaabbabb; 26].$$

The 14 visited cells in the left component contains a green disk so the returned value for the rank is 13. These two versions of the spiral traversal produce exactly the same values since all the additional cell visited in the extended version belongs to the right component by definition of $S_{[m \circ vb]}(m+s)$. The extended spiral is closer to the algorithm selecting a run for the rank while the shortest spiral induces more symmetries in the forthcoming the enumerative studies.

This paragraph consider the connection between the extended spiral traversal for the rank and the selected run on sorted configurations for the rank. We discuss the beginning of this spiral traversal, if any. If $s < 0$, then $S_{[m \circ vb]}(m+s) \leq m$ so there is no loop iteration and the output $rank = -1$ as expected. Otherwise, let $j = S_{[m \circ vb]}(m)$. We consider the iterations for k from m to $j-1$. It visits $j-m$ cells $m, m+1, \dots, j-1$ where m is a left cell and all other cells are right while j is the next left cell. This part may be interpreted as the computation of $\text{sort}(\text{park}(E.u)) = \text{scompact}[j \Rightarrow j \circ gabfb; s - (j - m)]$ as described by 8. in Proposition 9: move via operator E the left cell $m = \min L[m \circ vb]$ to the right component, increment the rank, then visit via operator T^{-1} all the $j-m$ cells starting at m in spiral order until you find the next left cell $j = \min L[j \circ gabfb]$. In addition, the value of s is decremented at each visit of a cell so we have $S_{[m \circ vb]}(m+s) = S_{[j \circ gabfb]}(j+s)$. Hence, by induction, we observe that this traversal simulates the run of the sorted version of the correct algorithm.

Proposition 10. (*Reading a proof for the rank using spiral traversals*)

Let $u = \text{scompact}[m \Rightarrow m \circ vb; s]$ be a sorted parking configuration. The extended spiral traversal for the rank on cylinder $[m \circ vb]$ returns $\text{rank}(u)$.

Proof. The discussion preceding this proposition shows that the correct algorithm on sorted configurations with input u is simulated by the traversal algorithm on cylinder $[m \circ vb]$ so that both return $\text{rank}(u)$. \square

We recall that by definition all the cells of spiral indices from $m+s+1$ to $S_{[m \circ vb]}(s)-1$, if any, are right cells so the extended and shortest spiral traversals, which differs only by these cells, returns the same value. We can also recover from the shortest traversal a proof g for the rank of the sorted parking configuration $u = \text{scompact}[m \Rightarrow m \circ vb; s]$. For a cut skew cylinder and $i = 1 \dots n-1$ we define the set $L_i[m \circ vb; s]$ of left cells of spiral coordinate at most $m+s$ and on the same row as the step related to $a_{m \uparrow i}$ the i -th step of the oriented segment a^{n-1} starting at m .

Proposition 11. (*Reading a proof for rank in spiral traversals*) Let $u = \text{scompact}[m \Rightarrow m \circ vb; s]$ be a sorted parking configuration on K_n . The configuration $g = (g_i; g_n) := (|L_i[m \circ vb; s]|; 0)$ is a proof for $\text{rank}(u)$.

Proof. In the correspondence between the selected run on sorted configurations and the extended traversal, a grain $\epsilon^{(i)}$ is removed on the vertex i each time a cell of the row related to u_i is moved from the left to the right component. All such left cells are the left cells in $L[m \circ vb]$ of spiral coordinate between m and $m+s$ since between $m+s+1$ and $m+S_{[m \circ vb]}(m+s)-1$ there are only right cells, if any. When those cells are partitioned

into rows, it corresponds exactly to the definition of $L_i[m \circlearrowleft vb; s]$. Hence g_i count the number of removed grains on vertex i during the algorithm so g is a proof for the rank of u according to the sorted version of the correct greedy naive algorithm. \square

It appears that the $(|L_i[m \circlearrowleft vb; s]|)_{i=1\dots n-1}$ may be efficiently computed via a single euclidean division.

Theorem 12. (Formula for the rank of a sorted parking configuration) *Let u be a sorted parking configuration on K_n and $u_n + 1 = q(n - 1) + r$ the euclidean division of $u_n + 1$ by $n - 1$.*

$$\text{rank}(u) = \left(\sum_{i=1}^{n-1} \max(0, q - i + 1 + u_i + \chi(i \leq r)) \right) - 1$$

and the proof for the rank found from the algorithm selecting a run for the rank by collecting removed grains is

$$g = (\max(0, q - i + 1 + u_i + \chi(i \leq r)); 0).$$

Proof. Let $u = \text{scompact}[m \Rightarrow m \circlearrowleft vb; s]$ with $s = u_n$. In this proof, we consider that the visited cells during the traversal algorithm are exactly the cells with spiral coordinate from m to $m + s$, if any. It means that we deliberately ignore the cells of spiral coordinates from $m + s + 1$ to $m + S_{[m \circlearrowleft vb]}(m + s) - 1$. These cells are necessarily right cells by definition of $S_{[m \circlearrowleft vb]}(m + s)$. With this convention, the number of visited cells on the row crossed by $a_{m \uparrow i}$ is $q + \chi(i \leq r)$. Moreover, on this row the traversal begins by visiting all the right cells of spiral coordinate at least m . These cells are counted by $i - 1 - u_i \geq 0$ by definition of $\text{scompact}[m \Rightarrow m \circlearrowleft vb; s]$. We conclude that

$$|L_i[m \circlearrowleft vb; s]| = \max(0, (q + \chi(i \leq r)) - (i - 1 - u_i))$$

since the visited left cell on the row crossed by $a_{m \uparrow i}$ are the visited cells minus the possibly first visited right cells. This shows that g is the proof computed by the traversal algorithm and a summation leads to the formula for the rank. \square

3.4 Optimizing the search of the equivalent sorted parking configuration

We describe here an efficient algorithm to compute for any configuration u on K_n the toppling and permuting equivalent configuration $\text{sort}(\text{park}(u))$.

Search toppling and permuting equivalent sorted parking configuration

Input: A configuration u on K_n .

$u' \leftarrow \text{compact}(u)$;

$u' \leftarrow \text{sort}(u')$;

For i from 1 to $n - 1$ do $k_i \leftarrow (n - 1)u'_i - n(i - 1)$;od;

$i_{\max} \leftarrow i$ such that $k_i = \max_{j=1\dots n-1}(k_j)$;

Let $k_{i_{\max}} = q(n - 1) + r$ be the euclidean division by $n - 1$ defining q and r ;

For i from 1 to $n - 1$ do $u''_i \leftarrow u'_{((i-1-r) \bmod (n-1))+1} - q + r + \chi(i \leq r)$;od;

$u''_n \leftarrow u'_n + k_{i_{\max}}$;

Output: The configuration $u'' (= \text{sort}(\text{park}(u)))$.

Proposition 13. *The preceding algorithm correctly returns $u'' = \text{sort}(\text{park}(u))$ and may be implemented with a linear arithmetic complexity.*

Proof. First we show that this algorithm is correct.

We implicitly use a formula for $T^j.u$ where $j \in \mathbb{Z}$ and u is a sorted compact configuration:

$$T^j.u = (T^{(1-n)})^{-q}T^r.u = T^r.u + q\Delta^{(n)} = (u_{(i-1-r \bmod n-1)+1} - q + r - n\chi(i \leq r); u_n + j)$$

where $j = q(n-1)+r$ is the euclidean division of j by $n-1$. This formula is obtain by iteration of the formula in the proof of 5. in Proposition 9. Roughly speaking, $u_{(i-1-r \bmod n-1)+1}$ takes into account the cyclic permutations induced by T^r , $-q$ corresponds to the $-q$ topplings of the sink, $+r$ to the r topplings of vertices in T^r and $-n\chi(i \leq r)$ compensate in the case where the vertex is toppled once during the computation of T^r .

The configuration $u' = \text{sort}(\text{compact}(u))$ is a sorted and compact configuration. We search the power $j \in \mathbb{Z}$ such that $T^j.u' = u'' = \text{sort}(\text{park}(u'))$. If $u'' = T^j.u'$ is sorted and parking then in particular $u''_1 = 0$. We use this equality as a criterion to filter the possible values of j . We discuss the $n-1$ possible values of r and indeed the two values of $\chi(1 \leq r)$. It will appear that for each value there exists a single value j_r such that $u^{(r)} := T^{j_r}.u'$ satisfies $u_1^{(r)} = 0$. For any choice of r , let $j_r = q_r(n-1) + r$ the euclidean division by $n-1$ of this hypothetical j_r and $i = (1-r) \bmod (n-1) + 1$. We then use the preceding formula for $T^{j_r}.u'$. If $r = 0$ then $i = r+1$ and $u_1^{(r)} = 0 \iff u'_i - q_r + r = 0 \iff j_r = (n-1)u'_i + nr = (n-1)u'_i + n(i-1)$. If $r > 0$ then $i = n-r$ and $u_1^{(r)} = 0 \iff u'_i - q_r + r - n = 0 \iff j_r = (n-1)u'_i + nr - n(n-1) = (n-1)u'_i + n(i-1)$. Hence, the expected value of j appears in $\{(n-1)u'_i + n(i-1)\}_{i=1\dots n-1}$ also called the $\{k_i\}_{i=1\dots n-1}$ in the algorithm.

We may write $u' = \text{scompact}[k \Rightarrow 0 \odot w; s]$ for $k = (n-1)u'_1$, $s = u'_n$ and some almost balanced word w . Since by definition $T^{j_r}.u' = u^{(r)} = \text{scompact}[k - j_r \Rightarrow 0 \odot w; s + j_r]$ satisfies $u_1^{(r)} = 0$, the cell of spiral coordinate $k - j_r$ is a left cell of cylinder $[0 \odot w]$. In the cut skew cylinder related to the sorted and parking $u'' = T^j.u'$ the cell $k - j$ should also be $\min L[0 \odot w]$ according to 6. in Proposition 9. This means that j is the maximal value among the $\{j_r\}_{r=1\dots n-1} = \{k_i\}_{i=1\dots n-1}$. This value is called $k_{i_{\max}}$ in the algorithm. This algorithm is correct since it returns $u'' = T^{k_{i_{\max}}}.u'$ computed via the first formula of this proof.

Then we discuss how this algorithm may be implemented within a linear arithmetic complexity also called $O(n)$.

The explicit definition of $\text{compact}(u)$ may be evaluated in $O(n)$. The sort of the compact configuration $u' = \text{compact}(u)$ may be performed in $O(n)$ using a classical sort by *values* since those values are between 0 and $n-1$. More precisely we use a vector $v = (v_i)_{i=0\dots n-1}$ initialized to $v = 0 \in \mathbb{Z}^{n-1}$, then we increment $v[u'_i]$ for $i = 1 \dots n-1$ and finally we compute $\text{sort}(u')$ has the concatenation of blocks of v_i values i for i from 0 to $n-1$. Our model of complexity is called arithmetic since we assume that an euclidean division or a comparison may be performed in constant time, no matter of the size of the involved integers. Within this framework, the evaluation of the k_i , the search of the

the change of variable (degree, rank) \rightarrow (rank + 1, $\binom{n-1}{2}$ - degree + rank) lay the values out so that the symmetry is the more explicit symmetry along the main diagonal. This motivates the definition of the two parameters $xpara(u) := rank(u) + 1$ and $ypara(u) := \binom{n-1}{2} - degree(u) + rank(u)$ on a configuration u and the definition of the generating function

$$M_n(x, y) := \sum_u x^{xpara(u)} y^{ypara(u)}$$

where u runs over sorted parking configurations on K_n .

This generating function is a power series in x and y since by definition $rank(u) \geq -1$ so $xpara(u) \geq 0$ and $ypara(u) \geq 0$ will be deduced from the proof of the conjectural symmetry $M_n(x, y) = M_n(y, x)$. This symmetry will be proven by the involution of Riemann-Roch theorem detailed in Subsection 4.1. We will interpret this involution in terms of cut skew cylinders. This interpretation will provide in Subsection 4.2 an alternative definition of the parameters $xpara$ and $ypara$ in terms of the traversal computing the rank. Moreover, in Subsection 4.3 geometric sums will appear along these computations and their summations will lead to an expression for

$$M(x, y; z) = \sum_{n \geq 1} M_n(x, y) z^n$$

as a rather simple rational function involving two copies of a classical generating function on Dyck words.

4.1 On the involution in Riemann-Roch theorem

For the graph G , the Riemann-Roch theorem relates the ranks of configurations u and $K_G - u$ on G . The map $u \rightarrow K_G - u$ is an involution. For the complete graph $G = K_n$, we have $K_G = (n-3; n-3)$. In this case, this involution combines nicely with the equivalence by toppling and permutation: if u and v are toppling and permuting equivalent then also are $K_G - u$ and $K_G - v$. This leads to the involution on sorted parking configurations on K_n ,

$$\xi.u := \text{sort} \circ \text{park}(K_G - u).$$

Proposition 15. *Let u be a sorted parking configuration on K_n , we have*

$$(xpara(\xi.u), ypara(\xi.u)) = (ypara(u), xpara(u)).$$

Proof. The proof is similar for any configuration u on K_n . The Riemann-Roch theorem applied to u implies that

$$xpara(K_G - u) = rank(K_G - u) + 1 = rank(u) - degree(u) + \binom{n}{2} - n + 1 = ypara(u).$$

Applying the same proof to $K_G - u$, we obtain $ypara(K_G - u) = xpara(u)$. □

Corollary 16. *For any $n \geq 1$, we have $M_n(x, y) = M_n(y, x)$.*

Lemma 18. (*Superimposition principle*) Let $u = \text{scompact}[k \Rightarrow m \circlearrowleft w; s]$ be a sorted and compact configuration on K_n .

$$u' := (-u_{n-i}; u_n) = \text{scompact}[k' \Rightarrow m' \circlearrowleft w'; s']$$

where $k' = n(1 - n) + (2n - 1) - k$, $m' = (2n - 1) - m$, $w' = \tilde{w}$ and $s = s'$.

Proof. The configuration $\text{scompact}[k \Rightarrow m \circlearrowleft w; s]$ is defined by a cut skew cylinder cylinder $[m \circlearrowleft w]$ and an oriented segment $k \uparrow$. The claim is the consequence of the following symmetries: reverse the orientation of the segment $k \uparrow$, reverse the order in which the cells are visited, the cell k becoming the cell $-k$, and change the corner related to a cell from south-east to north-west. This leads to a new cut skew cylinder cylinder $[m' \circlearrowleft w']$ and a new oriented segment $k' \uparrow$ (toward south). This data defines the new configuration $u' = \text{scompact}[k' \Rightarrow m' \circlearrowleft w'; s]$ as the image via the central symmetry with respect to the origin exchanging left and right and also top and bottom. An example is provided in Figure 4 where in red are the initial data and in green the new data. In this case, the green oriented segment $21 \uparrow$ is reversed to become the red oriented segment $\overline{62} \uparrow$. Inspection shows that $k' = -n^2 + 3n - 1 - k$, $m' = 2n - 1 - m$, and $w' = \tilde{w}$. We also have

$$u' := (u'_i; u'_n) = (-u_{n-i}; u_n)$$

since the differences of abscissa are opposite and the order of steps in the oriented segment $k' \uparrow$ reversed compared to $k \uparrow$. \square

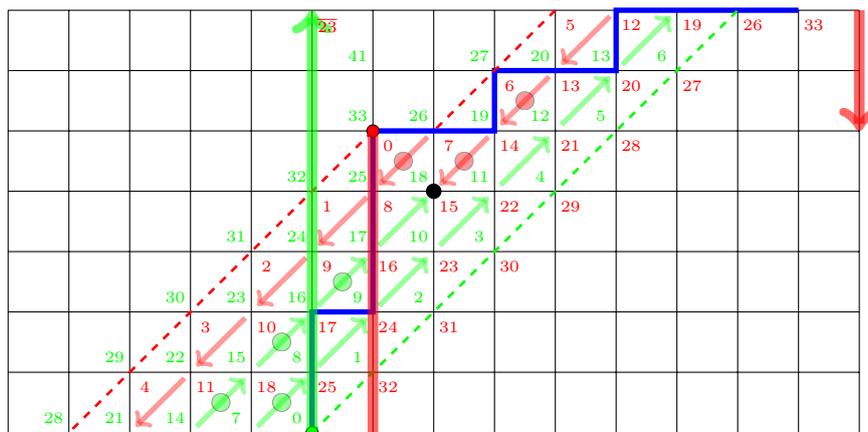


Figure 5: Superimposition principle for the traversals related to the computation of the rank of $\text{scompact}[0 \Rightarrow 0 \circlearrowleft aabaaabbabbabb; 10]$ and $\text{scompact}[0 \Rightarrow 0 \circlearrowleft aaabaabbbabbabb; 7]$.

We then use this superimposition principle in the proof of Proposition 17.

Proof. By definition, we have

$$\text{sort}(K_G - u) = (n - 3 - u_{n-i}; n - 3 - u_n) = (-u_{n-i}; -u_n) - (n - 3)\Delta^{(n)} + n(n - 3)\epsilon^{(n)}.$$

We factorize w in $w = fabgb$ where $v = fabg$ is the factorization at last maximal height of the Dyck word v defined by $w = vb$. Using the superimposition principle of Lemma 18 and changing the value of u_n to its opposite, we have

$$(-u_{n-i}; -u_n) = \text{scompact}[-n^2 + 3n - 1 \Rightarrow 2n - 1 \circlearrowleft b\tilde{g}baf\tilde{f}; -s].$$

It appears that $a\tilde{f}b\tilde{g}b = \xi(v)b$ is a cyclic conjugate of $b\tilde{g}baf\tilde{f}$. According to 2. in Proposition 9, we change the start of the cut to make appear this cyclic conjugate:

$$(-u_{n-i}; -u_n) = \text{scompact}[-n^2 + 3n - 1 \Rightarrow 2n - 1 + n|b\tilde{g}b|_a - (n - 1)|b\tilde{g}b|_b \circlearrowleft \xi(v)b; -s].$$

On one hand w is a cut, so we have $n|w|_a + (n - 1)|w|_b = 0$, hence $n|b\tilde{g}b|_a - (n - 1)|b\tilde{g}b|_b = -n|fa|_a + (n - 1)|fa|_b$. On another hand, since $m = 0 = \min L[0 \circlearrowleft w]$, $C(w) = \max R[0 \circlearrowleft fabg]$ is also the spiral label of the cell whose west vertical side is related to the explicit occurrence of a in $w = fabgb$, hence $C(w) = n|fa|_a - (n - 1)|fa|_b - (2n - 1)$. These two observations leads to

$$(-u_{n-i}; -u_n) = \text{scompact}[-n^2 + 3n - 1 \Rightarrow -C(w) \circlearrowleft \xi(v)b; -s].$$

Using 3. of Proposition 9, we increment the labeling of cells by $C(w)$ so that the cut starts in cell 0:

$$(-u_{n-i}; -u_n) = \text{scompact}[-n^2 + 3n - 1 + C(w) \Rightarrow 0 \circlearrowleft \xi(v)b; -s].$$

Adding $n(n - 3)\epsilon^{(n)}$ modifies only the value on the sink n and the $n - 3$ topplings of this sink n are equivalent to $T^{(1-n)(n-3)}$ hence using 5. of Proposition 9 we obtain

$$\text{sort}(K_{K_n} + (-u_{n-i}; -u_n)) = \text{scompact}[3(1 - n) + C(w) \Rightarrow 0 \circlearrowleft \xi(v)b; -s + (n - 3)].$$

According to 6. (and then 5.) in Proposition 9, we apply $T^{3(1-n)+C(w)}$ to obtain the toppling and permuting equivalent parking configuration $\text{sort}(\text{park}(K_{K_n} - u)) = \xi.u$ with the claimed expression in the proposition. \square

4.2 Interpretation of (xpara, ypara) on cut skew cylinders

Let $u = \text{scompact}[0 \Rightarrow 0 \circlearrowleft vb; s]$ be a sorted parking configuration on K_n . Let w a cyclic conjugate of vb and $m \in \mathbb{Z}$ be such that $\text{cylinder}[m \circlearrowleft w] = \text{cylinder}[0 \circlearrowleft vb]$ proven by a change of the start for the cut. The parameters $\text{xpara}(u)$ and $\text{ypara}(u)$ admits a description in terms of $\text{cylinder}[m \circlearrowleft w]$ and s . The *crossed left cells* of $u = \text{scompact}[k \Rightarrow m \circlearrowleft w; s]$ are

$$\text{CL}[m \circlearrowleft w; s] := \{k | k \leq s \text{ and } k \in L[m \circlearrowleft w]\}.$$

We also consider the *uncrossed right cells* defined by

$$\text{UR}[m \circlearrowleft w; s] := \{k | k > s \text{ and } k \in R[m \circlearrowleft w]\}.$$

We denote the cardinalities of these sets by

$$\text{left}[m \circlearrowleft w; s] := |\text{CL}[m \circlearrowleft w; s]| \text{ and } \text{right}[m \circlearrowleft w; s] := |\text{UR}[m \circlearrowleft w; s]|$$

We illustrate these definitions in Figure 5 in the particular case $w = vb$ where $v = aabaaabbabbabb$ is a Dyck word and so $m = 0$, and $s = 10$. This factorization $w = vb$ is also proven by the dashed green diagonal. The traversal of cells computing $\text{rank}(\text{scompact}[0 \Rightarrow 0 \circlearrowleft vb; s])$ visits the cells of spiral coordinate from 0 to s and those cells are crossed by a green diagonal arrow in north-west direction. There the cells of $\text{CL}[0 \circlearrowleft vb; s] = \{0, 7, 8, 9\}$ are marked by a green disk and the cells of $\text{UR}[0 \circlearrowleft vb; s] = \{11, 12, 18\}$ by a red disk.

Proposition 19. *Let $u = \text{scompact}[0 \Rightarrow 0 \circlearrowleft vb; s]$ be a sorted parking configuration on K_n , we have*

$$\text{xpara}(u) = \text{left}[0 \circlearrowleft vb; s] \text{ and } \text{ypara}(u) = \text{right}[0 \circlearrowleft vb; s].$$

Proof. In this case $\min L[0 \circlearrowleft vb] = 0$, so $\text{CL}[0 \circlearrowleft vb; s] = \{k | 0 \leq k \leq s \text{ and } k \in L[0 \circlearrowleft vb]\} = \cup_{i=1}^{n-1} L_i[0 \circlearrowleft vb; s]$. Hence the identity

$$|\text{CL}[m \circlearrowleft vb; s]| = \sum_{i=1}^{n-1} |L_i[m \circlearrowleft vb; s]| = \text{rank}(u) + 1 = \text{xpara}(u)$$

directly follows from the Proposition 11 describing a proof for the rank of u via the traversal.

The other identity is deduced from this first identity via an extension of the superimposition principle to the traversals computing the rank of u and $\xi.u$. An example of this superimposition with traversals is given in Figure 5. The green oriented vertical segment toward north, which defines u , starts from the south-east corner of spiral green coordinate 0. In red, we describe the same notions as in green but on $\xi.u = \text{scompact}[0 \Rightarrow 0 \circlearrowleft \xi(w)b; C(w) - s - 1]$ where $C(w) = 18$ so that $C(w) - s - 1 = 7$. The dashed red diagonal (drawn in two parts) matches $\xi(w)$, the red oriented vertical segment toward south, which defines $\xi.u$, starts from the north-west corner of red spiral coordinate 0 and the visited cells during the computation of the rank of $\xi.u$ are crossed by red diagonal arrows toward the south-west.

First we assume that $s \geq 0$ and $C(w) - s - 1 \geq 0$ as in Figure 5. It means that $(s+1) + (C(w) - s - 1 + 1) = C(w) + 1$ cells are crossed. This number corresponds exactly to the number of cells of spiral between the cell of spiral green coordinate 0 and the cell of spiral red coordinate 0. Hence the green and red sequences of diagonals both ends in the black disk which is the south-east corner of the cell of green spiral coordinate $s+n (= 18$ on example). Since the involution ξ on skew cylinders exchange left and right components, we deduce that the cells of spiral green coordinate $\text{UR}[0 \circlearrowleft vb; s]$ are also the cells of spiral red coordinate $\text{CL}[0 \circlearrowleft \xi(w)b; C(w) - s - 1]$. Hence $|\text{UR}[0 \circlearrowleft vb; s]| = \text{xpara}(\xi.u) = \text{ypara}(u)$.

If $s < 0$, then only red diagonal arrows may exist. Inspection shows that $\text{UR}[0 \circlearrowleft vb; s] = R[0 \circlearrowleft vb] \cap \{s, \dots, \max R[0 \circlearrowleft vb]\} = \text{CL}[0 \circlearrowleft \xi(w)b; C(w) - s - 1]$ hence $|\text{UR}[0 \circlearrowleft vb; s]| = \text{xpara}(\xi.u) = \text{rank}(\xi.u) + 1 = \text{ypara}(u)$.

If $C(w) - s - 1 < 0$, then only green diagonal arrows may exist. We have $s \geq C(w) = \max(R[0 \circlearrowleft vb]s)$ so $\text{UR}[0 \circlearrowleft vb; s]$ is empty. In addition, the value $C(w) - s$ on the sink of $\xi.u$ implies that this configuration is non-effective so $\text{rank}(\xi.u) = -1$. Hence, in this case one also has $|\text{UR}[0 \circlearrowleft vb; s]| = 0 = \text{xpara}(\xi.u) = \text{ypara}(u)$. \square

4.3 Enumeration of sorted parking configurations according to (degree, rank)

According 7. in Proposition 9, the bijective decomposition $u = \text{scompact}[0 \Rightarrow 0 \circ vb; s]$ of sorted parking configuration on K_n where v is a Dyck word of size $n - 1$ implies that

$$M_n(x, y) = \sum_v \sum_{s \in \mathbb{Z}} x^{\text{left}[0 \circ vb; s]} y^{\text{right}[0 \circ vb; s]}$$

where v runs over Dyck words of size $n - 1$. We introduce a partial sum of $M_n(x, y)$ indexed by a Dyck word v :

$$M_v(x, y) = \sum_{s \in \mathbb{Z}} x^{\text{left}[0 \circ vb; s]} y^{\text{right}[0 \circ vb; s]}.$$

We also introduce a slightly more general and redundant notation for monomials in these sums

$$M_{[m \circ w; s]} = x^{\text{left}[m \circ w; s]} y^{\text{right}[m \circ w; s]}.$$

First, we study independently each $M_v(x, y)$ for each Dyck word v defining the sorted and parking configurations $(\text{scompact}[0 \Rightarrow 0 \circ vb; s])_{s \in \mathbb{Z}}$. The partition $(L[0 \circ vb], R[0 \circ vb])$ of cells indexed by their spiral coordinate induces a partition of \mathbb{Z} into maximal intervals totally contained either in $L[0 \circ vb]$ or $R[0 \circ vb]$. We describe this partition in intervals with the maximal elements of each interval, if any. This leads to the sets

$$LR[0 \circ vb] := \{s | s \in L[0 \circ vb], s + 1 \in R[0 \circ vb]\}$$

and

$$RL[0 \circ vb] := \{s | s \in R[0 \circ vb], s + 1 \in L[0 \circ vb]\}.$$

Lemma 20. *Let v be a Dyck word and $H(x, y) := \frac{1-xy}{(1-x)(1-y)}$. We have*

$$M_v(x, y) = H(x, y) \left(\sum_{s \in RL[0 \circ vb]} M_{[0 \circ vb; s]} - \sum_{s \in LR[0 \circ vb]} M_{[0 \circ vb; s]} \right).$$

The notation of the proof are illustrated in Figure 6.

Proof. If $s < 0$, then by convention $s \in R[0 \circ vb]$. If $s \geq (n + 1)(n - 1)$, then $s \in L[0 \circ vb]$ since the greatest possible integer in the right cut appears only in the case $\text{cylinder}[0 \circ a^{n-1}b^n]$ and is $n^2 - 3n + 1$. Hence $|RL[0 \circ vb]| = |LR[0 \circ vb]| + 1$ and let $k := |LR[0 \circ vb]|$. We sort in increasing order the elements of $RL[0 \circ vb] = \{r_1, \dots, r_{k+1}\}$ and $LR[0 \circ vb] = \{l_1, \dots, l_k\}$. So the partition of \mathbb{Z} in maximal intervals induced by $(L[0 \circ vb], R[0 \circ vb])$ is

$$] - \infty, r_1], [r_1 + 1, l_1], [l_1 + 1, r_2], \dots, [r_k + 1, l_k], [l_k + 1, r_{k+1}], [r_k + 1, +\infty[.$$

To observe that on a given interval I of this partition, the sum $\sum_{s \in I} M_{[0 \circ vb; s]}$ is a geometric sum of reason x or $1/y$, the following key discussion comes directly from the definition of $CL[0 \circ vb; s]$ and $UR[0 \circ vb; s]$:

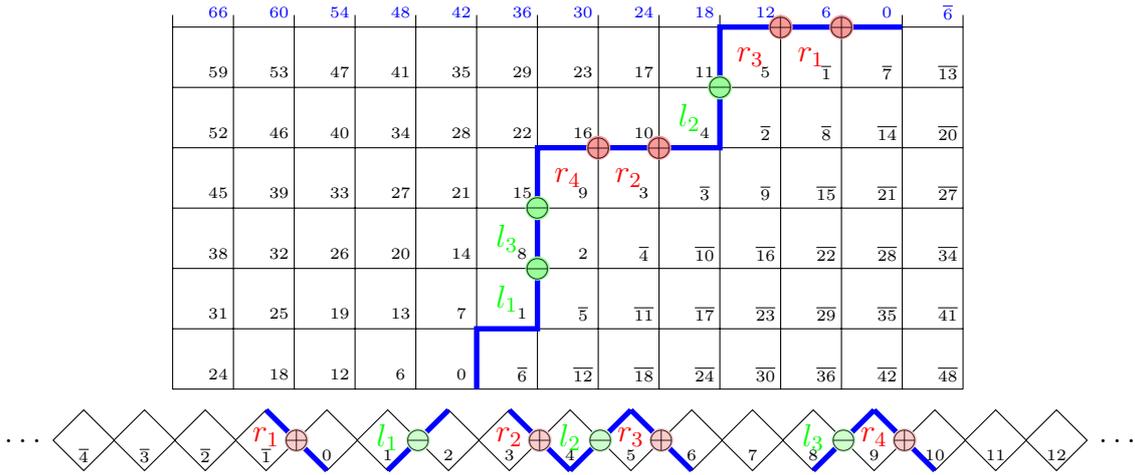


Figure 6: Example of decomposition of $M_v(x, y)$ into an alternation of geometric sums.

- either $s + 1 \in L[0 \circlearrowleft vb]$, then $\text{CL}[0 \circlearrowleft vb; s + 1] = \text{CL}[0 \circlearrowleft vb; s] \cup \{s + 1\}$ and $\text{UR}[0 \circlearrowleft vb; s + 1] = \text{UR}[0 \circlearrowleft vb; s]$.
- or $s + 1 \in R[0 \circlearrowleft vb]$, then $\text{CL}[0 \circlearrowleft vb; s + 1] = \text{CL}[0 \circlearrowleft vb; s]$ and $\text{UR}[0 \circlearrowleft vb; s + 1] = \text{UR}[0 \circlearrowleft vb; s] - \{s + 1\}$.

Hence for $i = 1, \dots, k$, for any j in the interval $[r_i + 1, l_i]$ we have $M_{[0 \circlearrowleft vb; j]} = xM_{[0 \circlearrowleft vb; j-1]}$ using previous discussion for $s = j - 1$. This leads to the geometric sum of reason x

$$\sum_{j=r_i+1}^{l_i} M_{[0 \circlearrowleft vb; j]} = \frac{xM_{[0 \circlearrowleft vb; r_i]} - xM_{[0 \circlearrowleft vb; l_i]}}{1 - x}.$$

Similarly, for $i = 1, \dots, k$ and the intervals $[l_i + 1, r_{i+1}]$, we obtain geometric sums of reason $1/y$ leading to

$$\sum_{j=l_i+1}^{r_{i+1}} M_{[0 \circlearrowleft vb; j]} = \frac{(1/y)M_{[0 \circlearrowleft vb; l_i]} - (1/y)M_{[0 \circlearrowleft vb; r_{i+1}]}}{1 - 1/y} = \frac{-M_{[0 \circlearrowleft vb; l_i]} + M_{[0 \circlearrowleft vb; r_{i+1}]}}{1 - y}.$$

For the two remaining infinite intervals $]-\infty, r_1]$ and $[r_{k+1} + 1, +\infty[$ we obtain respectively, after an inversion of summation order over first interval,

$$\sum_{j=-\infty}^{r_1} M_{[0 \circlearrowleft vb; j]} = \frac{M_{[0 \circlearrowleft vb; r_1]}}{1 - y} \quad \text{and} \quad \sum_{j=r_{k+1}+1}^{+\infty} M_{[0 \circlearrowleft vb; j]} = \frac{xM_{[0 \circlearrowleft vb; r_{k+1}]}}{1 - x}.$$

In the sum $M_v(x, y) = \sum_{j=-\infty}^{+\infty} M_{[0 \circlearrowleft vb; j]}$ decomposed into geometric sum on these intervals, each term $M_{[0 \circlearrowleft vb; r_i]}$ appears twice for a total factor of $H(x, y) = \frac{x}{1-x} + \frac{1}{1-y}$ and each term $M_{[0 \circlearrowleft vb; l_i]}$ also twice for a total factor $-H(x, y)$ as expected. \square

Proof. The assumption $n \geq 3$ is comfortable in our generic proof that may degenerate for $n = 1$ or $n = 2$.

Hence, we first consider that $n = 1$ and then $n = 2$. If $n = 1$, the sorted parking configurations are the $((; s))_{s \in \mathbb{Z}}$ and $\text{degree}((; s)) = s$ and $\text{rank}((; s)) = \max(-1, s)$. So $K_1(r, d) = \sum_{s \in \mathbb{Z}} d^s r^{\max(-1, s)}$ and after the change of variable, we have $L_1(x, y) = \frac{1-xy}{(1-x)(1-y)} = H(x, y)$. If $n = 2$, the sorted parking configurations are the $((0; s))_{s \in \mathbb{Z}}$ and $\text{degree}((0; s)) = s$ and $\text{rank}((0; s)) = \max(-1, s)$. So $K_2(r, d) = K_1(r, d)$ and after the change of variable one obtain $L_2(x, y) = L_1(x, y)$.

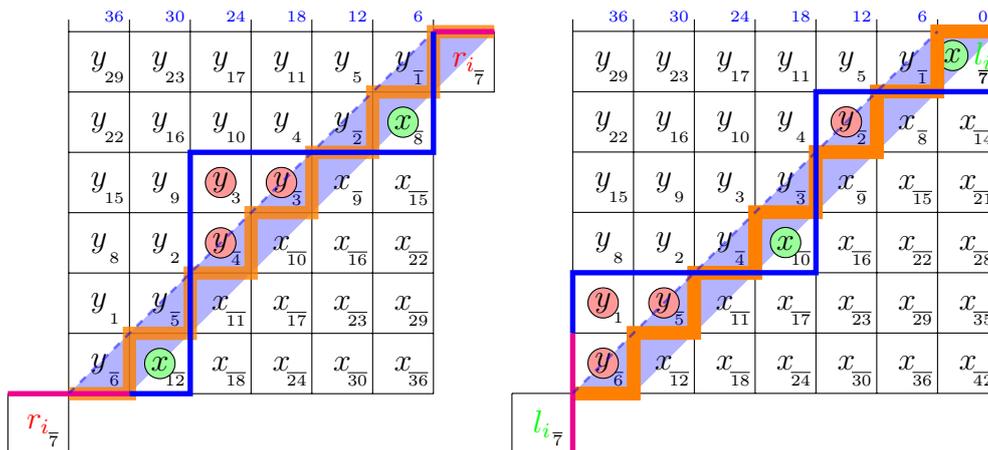


Figure 8: r_i leads to $bfb \in b\mathbb{X}a\mathbb{Y}(b\mathbb{X}a\mathbb{Y})^*b$ and l_i leads to $afa \in (\mathbb{Y} - \epsilon)b(\mathbb{X}a\mathbb{Y}b)^*(\mathbb{X} - \epsilon)$

We assume that $n \geq 3$. Starting from the expression of $M_v(x, y)$ in Lemma 20, we provide alternative descriptions of $M_{[0 \circ vb; r_i]}$ where $r_i \in RL[0 \circ vb]$ and of $M_{[0 \circ vb; l_i]}$ where $l_i \in LR[0 \circ vb]$.

The sequence of cells $(r_i + k)_{k=1 \dots n-1}$ contains the minimal cell in each row of spiral coordinate strictly greater than r_i . In Figure 8 at left, it corresponds to the cells crossed by the dashed blue diagonal. This observation proves that $L[r_i + n \circ (ba)^{n-1}b] = \{k | k > r_i\}$ and $R[r_i + n \circ (ba)^{n-1}b] = \{k | k \leq r_i\}$. In Figure 8, the cut skew cylinder $\text{cylinder}[r_i + n \circ (ba)^{n-1}b]$ corresponds to the orange path.

By choice, $r_i \in R[0 \circ vb]$ and $r_i + 1 \in L[0 \circ vb]$ so the vertex in south-east corner of cell $r_i + n$ belongs to the cut of cylinder $[0 \circ vb]$ and is preceded and followed by east steps. Changing the start of the cut to $r_i + n$, we obtain a cyclic conjugate bfb of vb such that $\text{cylinder}[0 \circ vb] = \text{cylinder}[r_i + n \circ bfb]$.

Intersecting the two preceding observations, we have

$$\text{CL}[0 \circ vb; r_i] = R[r_i + n \circ (ba)^{n-1}b] \cap L[r_i + n \circ bfb]$$

and

$$\text{UR}[0 \circ vb; r_i] = L[r_i + n \circ (ba)^{n-1}b] \cap R[r_i + n \circ bfb].$$

The embedding of $(ba)^{n-1}b$ and bfb both starts at the same vertex $r_i + n$. We factorize bfb at its steps which are common with those of $(ba)^{n-1}b$. In Figure 8 at left, the steps of

$bf b$ are magenta if forced and otherwise blue while the steps of $(ba)^{n-1}b$ are orange. Hence we factorize $bf b$ at the blue (or magenta) and orange steps. This example should help to convince the reader that $bf b = b(ba)a(aabb)b(ba)a()b$ is in general factorized as a word of $b\mathbb{X}a\mathbb{Y}(b\mathbb{X}a\mathbb{Y})^*b$. This factorization is non-ambiguous since the word $(ba)^{n-1}b$ is constant and the factors in \mathbb{X} and \mathbb{Y} are well defined. Conversely, any word $w \in b\mathbb{X}a\mathbb{Y}(b\mathbb{X}a\mathbb{Y})^*b$ such that $|w|_a \geq 2$ is almost balanced and may be written $bf b$ and is a cyclic conjugate of some word vb where v is a Dyck word.

Moreover, in the decomposition showing $bf b \in b\mathbb{X}a\mathbb{Y}(b\mathbb{X}a\mathbb{Y})^*b$, the augmented area of factors in \mathbb{X} counts the cells of $\text{CL}[0 \circ vb; r_i]$ and the augmented area of factors in \mathbb{Y} counts the cells of $\text{UR}[0 \circ vb; r_i]$. On the example of $bf b = b(ba)a(aabb)b(ba)a()b$, the first factor (ba) corresponds to $\{\overline{12}\} \subset \text{CL}[0 \circ vb; r_i]$, the factor $(aabb)$ corresponds to $\{\overline{4}, \overline{3}, 3\} = \text{CL}[0 \circ vb; r_i]$ and the second factor (ba) corresponds to $\{\overline{8}\} \subset \text{CL}[0 \circ vb; r_i]$ and the empty factor $()$ in \mathbb{Y} adds no cell to $\text{UR}[0 \circ vb; r_i]$. In terms of generating function, it leads to

$$\sum_v \sum_{r_i \in RL[0 \circ vb]} M_{[0 \circ vb; r_i]} = \frac{C(x; xz)zC(y; yz)}{1 - C(x; xz)zC(y; yz)} - z$$

where v runs over Dyck words of size at least 2. More precisely, $\frac{C(x; xz)zC(y; yz)}{1 - C(x; xz)zC(y; yz)}$ is the generating function of all the almost balanced words written $bf b$ according to the augmented area of factors \mathbb{X} , counted by the variable x , and the augmented area of factors \mathbb{Y} , counted by the variable y , and the size $|bf b|_a \geq 1$ counted by z . The subtracted term z corresponds to the excluded almost balanced word of size 1, bab .

The proof for a cell l_i is very similar and illustrated by the right example in Figure 8. The choice $l_i \in L[0 \circ vb]$ and $l_i + 1 \in R[0 \circ vb]$ implies that the vertex $l_i + n$ belongs to the cut of cylinder $[0 \circ vb]$ and is preceded and followed by north steps. Hence when the vertex $l_i + n$ is chosen as the start of the cut, it makes appear a cyclic conjugate afa of vb such that $\text{cylinder}[0 \circ vb] = \text{cylinder}[l_i + n \circ afa]$. We have

$$\text{CL}[0 \circ vb; l_i] = R[l_i + n \circ (ba)^{n-1}b] \cap L[l_i + n \circ afa]$$

and

$$\text{UR}[0 \circ vb; l_i] = L[l_i + n \circ (ba)^{n-1}b] \cap R[l_i + n \circ afa].$$

The factorization of afa at the common steps of the embedding of afa and $(ba)^{n-1}b$ leads to words in $(\mathbb{Y} - \epsilon)b(\mathbb{X}a\mathbb{Y}b)^*(\mathbb{X} - \epsilon)$ where the first \mathbb{Y} factor and the last \mathbb{X} factor are non empty due to the distinct extreme steps of these two embedded words. Hence we have,

$$\sum_v \sum_{l_i \in LR[0 \circ vb]} M_{[0 \circ vb; l_i]} = \frac{(C(x; xz) - 1)(C(y; yz) - 1)}{1 - C(x; xz)zC(y; yz)}$$

where v runs over Dyck words of size at least 2.

We conclude the proof by a summation following Lemma 20 to get the expected expression. \square

5 Enumerative byproducts related to q, t -Catalan numbers

In a Dyck word v we denote by a_i the i -th of the occurrence of the letter a in the word v . The *height* $h(a_i)$ of a_i is defined by $|p|_a - |p|_b$ where $v = pa_iq$ is a decomposition defining the prefix p . The letter a_i and a_j in the Dyck word v are in *dinv-interaction* if either $i < j$ and $h(a_i) = h(a_j)$ or $j < i$ and $h(a_j) = h(a_i) + 1$. The *parameter* $\text{dinv}(v)$ of the Dyck word v is the number of *dinv-interactions* in v . Haiman, quoted by Haglund [23], devised this statistic to provide a combinatorial interpretation of

$$C_n(q, t) := \sum_v q^{\text{area}(v)} t^{\text{dinv}(v)}$$

where v runs over Dyck words of size n . These polynomials initially appears as Hilbert series of some space of multivariate polynomials, see again [23] for the context. The algebraic definition of these Hilbert series has an obvious symmetry implying that $C_n(q, t) = C_n(t, q)$. This symmetry has still no elementary explanation in terms of the combinatorial interpretation, see [24] a recent chapter due to Haglund.

This work is not the first connection between the sandpile model and these q, t -Catalan numbers. For example, in [19], one of the author with his co-author interprets on sorted recurrent configurations the pair of statistics (area, bounce) also defining the same q, t -Catalan numbers. Classically the area parameter corresponds to the number of grains outside the sink. The bounce parameter, not defined in this paper, corresponds to the cumuled elapsed time on vertices before toppling in the maximally parallel execution of the Dhar criterion which topples the sink and then once all other vertices.

Hence, the additional results of this section are strengthening the connection between the sandpile model on K_n and these q, t -Catalan numbers. Subsection 5.1 shows that the parameter dinv is preserved by the map ξ . Subsection 5.2 details a relation between ξ and a map ζ due to Haglund which maps the bivariate (dinv, area) to the bivariate (area, bounce). Subsection 5.3 introduces the parameter $\text{prerank}(w) := \text{area}(\xi(w))$ which appear in one of our early optimisation of the algorithm for the rank. The bivariate (prerank, dinv) is another possible definition of $C_n(q, t)$.

5.1 A dinv statistic on cut skew cylinders.

The involution ξ on Dyck words related to the Riemann-Roch theorem for graphs preserves the dinv parameter.

Proposition 22. *For any Dyck word v , we have $\text{dinv}(\xi(v)) = \text{dinv}(v)$.*

The postponed proof of this proposition is based on a definition of a parameter cdinv on the cylinder $\text{cylinder}[m \circlearrowleft w]$ such that $\text{dinv}(v) = \text{cdinv}(\text{cylinder}[0 \circlearrowleft vb])$. Then the superimposition principle will imply the proposition.

The (*left-*)*contacts* $(c_i)_{i=1 \dots n-1}$ in the cut skew cylinder $\text{cylinder}[m \circlearrowleft w]$ are the cells where c_i is the smallest spiral coordinate for a left cell in the row crossed by $a_{m \uparrow i}$. Two distinct contacts c_i and c_j in $\text{cylinder}[m \circlearrowleft w]$ are in *cdinv-interaction* if $|c_i - c_j| < n - 1$.

The *cyclic*-dinv parameter $\text{cdinv}(\text{cylinder}[m \circlearrowleft w])$ is the number of pairs of contacts in cdinv -interaction.

In Figure 5 the green spiral coordinates of contacts for $\text{cylinder}[0 \circlearrowleft aabaaabbbabbabb]$ are $(c_i)_{i=1\dots n-1} = (0, 8, 9, 17, 25, 19, 13)$ so

$$\begin{aligned} \text{cdinv}(v) &= |\{\{8, 9\}, \{8, 13\}, \{9, 13\}, \{13, 17\}, \{13, 19\}, \{17, 19\}, \{19, 25\}\}| = 7 \\ &= \text{dinv}(v) = |\{\{a_2, a_3\}, \{a_2, a_7\}, \{a_3, a_7\}, \{a_7, a_4\}, \{a_7, a_6\}, \{a_4, a_6\}, \{a_6, a_5\}\}| \end{aligned}$$

Proof. The contacts c_i and c_j where $i < j$ are in cdinv -interaction in $\text{cylinder}[0 \circlearrowleft vb]$ if and only if either $c_i < c_j$ and then $h(a_i) = h(a_j)$ so a_i and a_j are in the first kind of dinv -interaction or $c_j < c_i$ and then $h(a_i) = h(a_j) + 1$ so a_j and a_i are in the second kind of dinv -interaction. Hence $\text{cdinv}(\text{cylinder}[0 \circlearrowleft vb]) = \text{dinv}(v)$.

Then we consider the superimposition of $u = \text{scompact}[0 \Rightarrow 0 \circlearrowleft vb; s]$ and $\xi.u = \text{scompact}[0 \Rightarrow 0 \circlearrowleft \xi(v)b; C(vb) - s]$ like in Figure 5. We remind that the spiral green coordinate $k \in \mathbb{Z}$ of a cell corresponds to the usual spiral coordinate in $\text{cylinder}[0 \circlearrowleft vb]$ while the red spiral coordinate $C(w) - k$ corresponds to the coordinate in the superimposed $\text{cylinder}[0 \circlearrowleft \xi(v).b]$. We consider the contact of green spiral coordinate c_i in $\text{cylinder}[0 \circlearrowleft vb]$. By definition, the east side of c_i belongs to the cut. The other cell at east sharing this side has green coordinate $c_i - (n - 1)$ and red coordinate $d_{i'} = C(w) - c_i + (n - 1)$ and is one of the contacts $(d_{i'})_{i'=1\dots n}$ in $\text{cylinder}[0 \circlearrowleft \xi(v)b]$. For any other contact c_j , a similar proof leads to the contact $d_{j'} = C(w) - c_j + (n - 1)$ in $\text{cylinder}[0 \circlearrowleft \xi(v)b]$. Since $d_{i'} - d_{j'} = c_j - c_i$ we deduce that c_i and c_j are in cdinv -interaction if and only if $d_{i'}$ and $d_{j'}$ are in cdinv -interaction. Hence $\text{cdinv}(\text{cylinder}[0 \circlearrowleft vb]) = \text{cdinv}(\text{cylinder}[0 \circlearrowleft \xi(v)b])$.

For example, in Figure 5 the red spiral coordinates of contacts for $\text{cylinder}[0 \circlearrowleft aaabaabbbabbabb]$ are $(d_j)_{j=1\dots n-1} = (0, 8, 16, 17, 25, 12, 6)$ and

$$\begin{aligned} \{C(w) - d_j + (n - 1)\}_{j=1\dots n-1} &= \{18 - d_j + 7\}_{j=1\dots n-1} \\ &= \{25, 17, 9, 8, 0, 13, 19\} \\ &= \{c_i\}_{i=1\dots n-1}. \quad \square \end{aligned}$$

5.2 A comparison of the involution ξ in Riemann-Roch and ζ map of Haglund

The involution ξ on Dyck words preserves the dinv statistic. Haglund defined a bijective map ζ such that for any Dyck word v , $\text{dinv}(v) = \text{area}(\zeta(v))$ and $\text{area}(v) = \text{bounce}(\zeta(v))$ where bounce is a statistic still undefined in this paper, see [23]. An anonymous referee for a conference where this work was presented [12] suggested to study possible relations between the maps ζ and ξ . We use the notation $\text{R}(w) := \kappa(\tilde{w})$ for a word w to describe the relation we found between ξ and ζ . In other words, the operator R reverses the letters of v and then replace each letter a by a letter b and each letter b by a letter a .

Proposition 23. *For any Dyck word v , we have $\zeta(\xi(v)) = \text{R}(\zeta(v))$.*

The proof of this proposition is based on reformulations of the ζ map in our framework and leads to another application of the superimposition principle. The *height vector* of a Dyck word v of size n is $(h(a_i))_{i=1\dots n}$ where we recall that a_i is the i -th occurrence of a letter a in v .

Proof. We reformulate Haglund’s definition of $\zeta(v)$ by the following equivalent algorithm on the height vector:

Haglund’s ζ map via height vector

Input: v of size $n - 1$
 $w \leftarrow \epsilon$;
 For h from 0 to n do
 .. $\eta^h \leftarrow \epsilon$;
 ..For i from 1 to $n - 1$ do
if $h(a_i) = h$ then $\eta^h \leftarrow \eta^h a$ fi;
if $h(a_i) = h - 1$ then $\eta^h \leftarrow \eta^h b$ fi;od;
 .. $w \leftarrow w\eta^h$;od;
Output: $w (= \zeta(v))$;

Haglund’s ζ map via traversal

Input: A cut skew cylinder $\text{cylinder}[0 \cup vb]$.
 Compute the contacts $(c_i)_i$ in each row.
 For i from 1 to $n - 1$ do
 ..add a letter a in cell c_i
 ..add a letter b in cell $c_i + n - 1$; od;
 $w \leftarrow \epsilon$
 For k from 0 to $\max_i(c_i + n - 1)$ do
 ..append letter in cell k to w , if any;od;
Output: $w (= \zeta(v))$;

Haglund’s definition is slightly different and a bit more trickier. It distinguishes as particular the two degenerated loop iterations for $h = 0$, where no letter b can be read and for $h = 1 + \max_i h(a_i)$ where no letter a can be read. Moreover it reads the height vector and the factors η^h from right to left. It is left to the reader to figure out that our definition is equivalent.

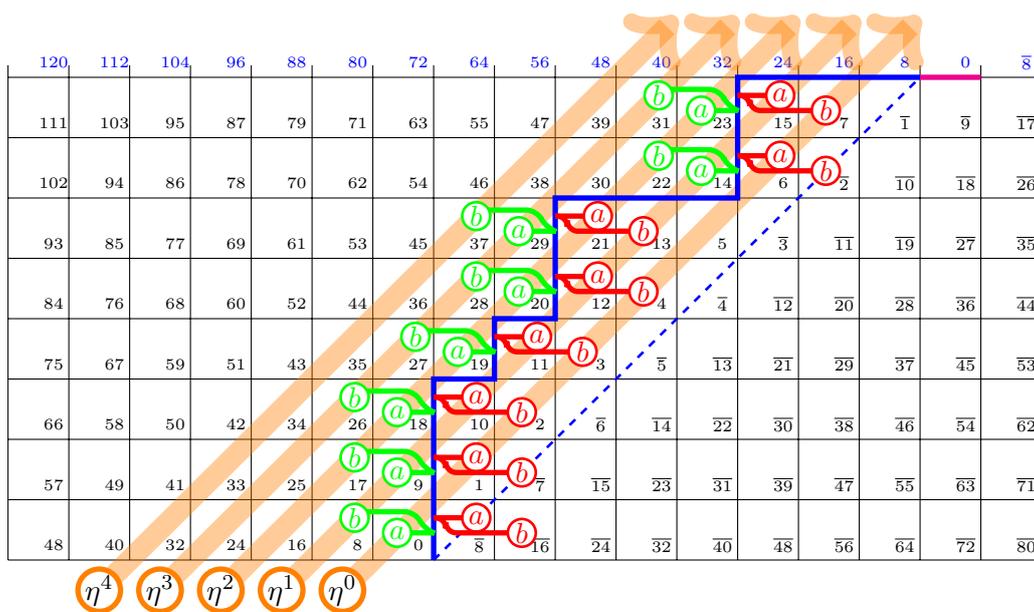


Figure 9: Description of Haglund’s ζ map via traversal of a cut skew cylinder

In Figure 9, we use the same example as Haglund (see page 50 in [23]). We embed our first algorithm describing the ζ map in a cut skew cylinder. The spiral traversal visit the cells diagonals by diagonals drawn in orange. Split diagonal by diagonal, this traversal corresponds to the nested loops on h and i in the algorithm defining ζ via the height vector. We consider the cells visited during the traversal of the h -th diagonal supposed

to define $\eta^{(h)}$. If a cell is a contact c_i in this diagonal, it means that the a step in vb in the same row is at height h in v . So it exactly corresponds to adding a letter a to $\eta^{(h)}$. We mark such a cell by a *green* letter a in the cell c_i . If a cell is $c_j + (n - 1)$ where c_j is a contact, it means that the a step in vb in the same row is at height $h - 1$ in v . So it exactly corresponds to adding a letter b in $\eta^{(h)}$. We mark such a cell by a *green* letter b in the cell $c_j + (n - 1)$. Since the traversal of the diagonal respect the order of the inner loop on i in the algorithm, we deduce that $\eta^{(h)}$ is also define by the order in which the preceding cells c_i or $c_j + (n - 1)$ are visited. These observation should convince the reader that the second algorithm entitled “Haglund’s ζ map via traversal” also defines the ζ map. On the example, collecting the *green* letters on each orange diagonal leads to $\eta^{(0)} = a, \eta^{(1)} = baa, \eta^{(2)} = baaaba, \eta^{(3)} = bbbab$ and $\eta^{(4)} = b$.

We reformulated the definition of ζ into a spiral traversal since it is well suited for the superimposition principle. The proof is similar to the proof of Proposition 22 that ξ preserves the dinv parameter. We keep only the (green) spiral coordinates here. The vertical step defining the contact c_i in vb shows that $\zeta(v)$ add a letter a in cell c_i and a letter b in cell $c_i + (n - 1)$. But this vertical step also appears in $\xi(v)b$ and in the computation of $\zeta(\xi(v))$, we add a (red) letter a in cell $c_i - (n - 1)$ and a letter (red) b in cell $c_i - 2(n - 1)$. Moving the red letters two west steps and exchanging the red letters a and b on each row leads exactly to the position of the green letters. Then inspection shows that the reversed traversal in the computation of $\zeta(\xi(v))$ leads also to $R(\zeta(v))$. \square

5.3 A prerank statistic so that (prerank, dinv) defines expected q, t -Catalan

The *prerank* of a Dyck word v is

$$\text{prerank}(v) := \text{area}(\xi(v)).$$

Proposition 24. *For any $n \geq 0$, we have*

$$C_n(q, t) := \sum_v q^{\text{dinv}(v)} t^{\text{area}(v)} = \sum_v q^{\text{dinv}(v)} t^{\text{prerank}(v)}$$

where v runs over Dyck words of size n .

Proof. The involution ξ satisfies $\text{prerank}(v) = \text{area}(\xi(v))$ and $\text{dinv}(v) = \text{dinv}(\xi(v))$. \square

We call this parameter prerank because of an early partial optimization of our algorithms computing the rank. A *staircase* (sorted parking) configuration u on K_n is such that for $i \neq n$, $u_i = i - 1$ and there are no restriction on u_n . Any staircase configuration may be written $u = \text{scompact}[m \Rightarrow m \circ (ab)^{n-1}b; s]$. Applying 8. in Proposition 9, we have $\text{sort}(\text{park}(u - \epsilon^{(1)})) = \text{scompact}[m + 1 \Rightarrow m + 1 \circ (ab)^{n-1}b; s - 1]$ which is also a staircase configuration. By induction, we have

$$\text{rank}(\text{scompact}[m \Rightarrow m \circ (ab)^{n-1}b; s] = \max(-1, s).$$

Hence one may accelerate the algorithm computing the rank as soon as we reach a staircase configuration. It appears that, if s is big enough, the number of loops before we reach such a staircase configuration is finite and is exactly the prerank of the initial Dyck word in the cut vb .

References

- [1] O. Amini and M. Manjunath. Riemann-Roch for sub-lattices of the root lattice a_n . *Electron. J. Combin.*, 17:R124, 2010.
- [2] Jean-Christophe Aval, Michele D’Adderio, Mark Dukes, Angela Hicks, and Yvan Le Borgne. Statistics on parallelogram polyominoes and a q, t -analogue of the Narayana numbers. *J. Combin. Theory Ser. A*, 123:271–286, 2014.
- [3] S. Backman. Riemann-Roch theory for graph orientations. [arXiv:1401.3309](https://arxiv.org/abs/1401.3309), 2014.
- [4] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality: An explanation of $1/f$ noise. *Phys. Rev Letters*, 59:381–384, 1987.
- [5] P. Bak, C. Tang, and K. Wiesenfeld. Self-organised criticality. *Physical Review A.*, 38:364–374, 1988.
- [6] M. Baker and S. Norine. Riemann-Roch and Abel-Jacobi theory on a finite graph. *Advances in Mathematics*, 215:766–788, 2007.
- [7] M. Baker and F. Shokrieh. Chip-firing games, potential theory on graphs, and spanning trees. *J. Combin. Theory Ser. A*, 120:164–182, 2013.
- [8] B. Benson, D. Chakrabarty, and P. Tetali. G-parking functions, acyclic orientations and spanning trees. *Discrete Mathematics*, 310:1340–1353, 2010.
- [9] N. Biggs. Chip-firing and the critical group of a graph. *J. Algebraic Combin.*, 9:25–45, 1999.
- [10] A. Björner, L. Lovász, and P. Shor. Chip-firing game on graphs. *European J. Combin.*, 12:283–291, 1991.
- [11] R. Cori and Y. Le Borgne. The sand-pile model and Tutte polynomials. *Advances in App. Math.*, 30:44–52, 2003.
- [12] R. Cori and Y. Le Borgne. On ranks of configurations on the complete graph. *DMTCS Proceedings, 25th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2013)*, 2013.
- [13] R. Cori and D. Rossin. On the sandpile group of dual graphs. *European J. Combin.*, 21:447–459, 2000.
- [14] D. Dhar. Self-organized critical state of the sandpile automaton models. *Phys. Rev. Lett.*, 64:1613–1616, 1990.
- [15] D. Dhar. Theoretical studies of self-organized criticality. *Physica A*, 369 (1), 2006.
- [16] D. Dhar and S. Majumdar. Equivalence between the Abelian sandpile model and the $q \rightarrow 0$ limit of the Potts model. *Physica A*, 185:129–135, 1992.
- [17] D. Dhar and R. Ramaswamy. Exactly solved model of self-organized critical phenomena. *Phys. Rev. Lett.*, 63:1659–1662, 1989.
- [18] D. Dhar, P. Ruelle, S. Sen, and D. Verma. Algebraic aspects of abelian sandpile model. *J. Phys. A*, A28:805–831, 1995.

- [19] Mark Dukes and Yvan Le Borgne. Parallelogram polyominoes, the sandpile model on a complete bipartite graph, and a q, t -Narayana polynomial. *J. Combin. Theory Ser. A*, 120(4):816–842, 2013.
- [20] A. Dvoretzky and T. Motzkin. A problem of arrangements. *Duke Math. J.*, 14:305–313, 1947.
- [21] H. M. Farkas and I. Kra. *Riemann Surfaces*. Graduate Texts in Mathematics, Springer, 1992.
- [22] Kramers H. and Wannier C. Statistics of the two-dimensional ferromagnet. part I. *Physical Review*, 60:252–262, 1941.
- [23] J. Haglund. *The q, t -Catalan numbers and the space of diagonal harmonics*. AMS University Lectures Series, 2008.
- [24] Jim Haglund. *Handbook of Enumerative Combinatorics*, chapter Catalan Paths and q, t -Enumeration. 2015.
- [25] V. Kiss and L. Thóthmérész. Chip-firing games on eulerian digraphs and np-hardness of computing the rank of a divisor on a graph. [arXiv:1407.6958](https://arxiv.org/abs/1407.6958), 2014.
- [26] D. E. Knuth. *The art of Computer Programming, Volume 4 Combinatorial Algorithms, Fasc 4A*. Addison Wesley, 2004.
- [27] Yvan Le Borgne. Counting upper interactions in Dyck paths. *Sém. Lothar. Combin.*, 54:Art. B54f, 16 pp. (electronic), 2005/07.
- [28] D. Lorenzini. Two-variable zeta-functions on graphs and riemann-roch theorems. *Int. Math. Res. Notices.*, 2012,22:5100–5131, 2012.
- [29] C. Merino-Lopez. Chip firing and Tutte polynomials. *Ann. Combin.*, 3:253–259, 1997.
- [30] Grigory Mikhalkin and Ilia Zharkov. Tropical curves, their Jacobians and theta functions. In *Curves and abelian varieties*, volume 465 of *Contemp. Math.*, pages 203–230. Amer. Math. Soc., Providence, RI, 2008.
- [31] D. Perkinson, J. Perlman, and J. Wilmes. Primer for the algebraic geometry of sandpiles. [arXiv:1112.6163](https://arxiv.org/abs/1112.6163), 2011.

6 Annex: A proof of the Riemann Roch Theorem for graphs

We consider the following solitary game on an undirected (non oriented) connected graph $G = (X, E)$ without loops: at the beginning integer values f_i are attributed to the n vertices x_1, x_2, \dots, x_n of the graph, these values can be positive or negative and define a configuration f . At each step a toppling can be performed by the player on a vertex x_i , it consists in subtracting d_i (the number of edges incident to x_i) to the amount f_i and for each neighbor x_j of x_i increase f_j by the number of edges between these two vertices. In this operation the amount of vertex x_i may become, or stay negative. The aim of the player is to find a sequence of toppling operations which will end with a configuration where all the f_i are non negative. Since the sum of the f_i is invariant by toppling, a

necessary condition to succeed is that in the initial configuration this sum should be non negative. We will see that this condition is not sufficient.

This game has much to do with the chip firing game (see [10], [9]) and the sandpile model (see [5], [15], [16]), for which recurrent configurations were defined and proved to be canonical representatives of the classes of configurations equivalent by a sequence of topplings (for a more algebraic treatment see also [31]).

The game was introduced and studied in detail by Baker and Norine in [6] who also introduced a new parameter on graph configurations: the rank. One characteristic of the rank $\rho(f)$ of a configuration f is that it is non negative if and only if one can get from f a configuration non-negative on every vertices by performing a sequence of topplings. For this parameter they obtain a simple formula expressing a symmetry similar to the Riemann-Roch formula for surfaces and curves (a classical reference to this formula is the book by H. Farkas and I. Kra [21]).

Our aim here is to give a simple presentation of this Riemann-Roch like theorem for graphs. It's worthwhile in this context to mention the independant work of Backman [3] and the precedings works [30, 1].

6.1 Configurations on a graph

Let $G = (X, E)$ be a multi-graph with n vertices, where $X = \{x_1, x_2, \dots, x_n\}$ is the vertex set and E is a symmetric matrix such that $e_{i,j}$ is the number of edges with endpoints x_i, x_j , hence $e_{i,j} = e_{j,i}$. In all this paper n denotes the number of vertices of the graph G and m the number of its edges. We assume that G is connected and has no loops, so that $e_{i,i} = 0$ for all i .

We will consider *configurations* on a graph, these are elements of the discrete lattice \mathbb{Z}^n . Each configuration f may be considered as assigning (positive or negative) tokens to the vertices. The symbol $\varepsilon^{(i)}$ will denote the configuration in which the value 1 is assigned to vertex x_i and the value 0 is assigned to all other vertices.

The *degree* of the configuration f is the sum of the f_i 's it is denoted by $deg(f)$.

6.1.1 The Laplacian configurations

These configurations correspond to the rows of the Laplacian matrix of a graph, a classical tool in Algebraic Graph Theory.

The *Laplacian configuration* $\Delta^{(i)}$ is given by: $\Delta^{(i)} = d_i\varepsilon^{(i)} - \sum_{j=1}^n e_{i,j}\varepsilon^{(j)}$, where $d_i = \sum_{j=1}^n e_{i,j}$ is the degree of the vertex x_i . These configurations which degrees are equal to 0 play a central role.

We denote by L_G the subgroup of \mathbb{Z}^n generated by the $\Delta^{(i)}$, and two configurations f and g will be said *toppling equivalent* if $f - g \in L_G$, which will also be written as $f \sim_{L_G} g$.

In the sandpile model, the transition from configuration f to the configuration $f - \Delta^{(i)}$ is allowed only if $f_i \geq d_i$ and is called a toppling. The same condition is assumed in the theory of chip firing games, and toppling is called firing; here we omit this condition and perform topplings even if $f_i < d_i$.

Notice that $\sum_{i=1}^n \Delta^{(i)} = 0$ and that for a connected graph this is the unique relation (up to multiplication by a constant) satisfied by the $\Delta^{(i)}$, moreover the principal minors of the Laplacian matrix are all equal to the number of spanning trees of the graph.

6.1.2 Recurrent configurations

We use in this section the notation usually considered in the sandpile model, so that we will call *sandpile configuration* a configuration f such that $f_i \geq 0$ for all $i < n$. This corresponds to the fact that in the sandpile model the vertex x_n is considered as a sink collecting tokens, so that the number of tokens of the sink has not taken into account in this context.

Definition 25. In the sandpile model, a toppling on vertex x_i , where $i \neq n$, may occur in a sandpile configuration only if $f_i \geq d_i$. A sandpile configuration f is stable if no toppling can occur, that is $f_i < d_i$ for all $i < n$.

Notice that when a toppling occurs in the sandpile model, the configuration $f - \Delta^{(i)}$ is also a sandpile configuration.

The toppling operation for a sandpile configuration will be denoted by $f \rightarrow g$. We also write:

$$f \xrightarrow{*} g$$

if f and g are sandpile configurations and if g is obtained from f by a sequence of toppling operations meeting only sandpile configurations. Notice that $f \xrightarrow{*} g$ implies $f \sim_{L_G} g$.

Sequences of topplings may be performed in any order until a stable configuration is attained as the following proposition states, the proof of which may be found in [17] or in [26] pages 42, and 70.

Proposition 26. For any sandpile configuration f there exists a unique stable configuration f' such that $f \xrightarrow{*} f'$.

A configuration is *recurrent* in an evolving system if it could be observed after a long period of the evolution of the system. In the case of the sandpile model, the system is considered to evolve by adding a token in any cell at random and then applying topplings until a stable configuration is reached. This translates into the following notion which is central :

Definition 27. A configuration f is recurrent if it is stable and there exists a sandpile configuration $g \neq 0$ such that $f + g \xrightarrow{*} f$.

The following important result, giving canonical representatives in the classes of the relation \sim_{L_G} is obtained in [14, 9, 13] by different ways.

Theorem 28. For any configuration f there exists a unique recurrent configuration g such that $f \sim_{L_G} g$.

In order to characterize the recurrent configurations D. Dhar used the configuration $\Delta^{(n)}$ and proposed the following algorithm.

Theorem 29. Burning Algorithm. *A stable configuration u is recurrent if and only if*

$$f - \Delta^{(n)} \xrightarrow{*} f$$

Moreover in this sequence of topplings each vertex different from x_n topples exactly once.

This algorithm can be translated into another characterization, giving:

Corollary 30. *A stable configuration f is recurrent if and only if for any non-empty subset Y of $X \setminus \{x_n\}$ there is at least an x_k in Y such that its degree in the subgraph spanned by Y is not greater than f_k , more precisely if the following condition is satisfied:*

$$f_k \geq \sum_{x_i \in Y} e_{i,k} \tag{1}$$

Proof. Let f be a recurrent configuration, and Y be a subset of X , then by Dhar's Burning Algorithm, starting from the configuration $f - \Delta^{(n)}$ there is a sequence of topplings of the vertices in which any vertex topples. We may suppose that the vertices are numbered in the order in which they topple, x_1 just after x_n , then x_2 and so on until x_{n-1} then for allowing a toppling at vertex x_i each f_i has to satisfies the condition:

$$d_i \leq f_i + \sum_{j=1}^{i-1} e_{i,j}$$

Now for any subset Y of X , let k be the smallest integer such that $x_k \in Y$, then since there is no $x_i \in Y$ with i less than k we have:

$$d_k \geq \sum_{j=1}^{k-1} e_{j,k} + \sum_{x_i \in Y} e_{i,k}$$

Putting $i = k$ in the first inequality and the two inequalities together gives the result.

Conversely if f is a stable configuration satisfying condition 1 we build a toppling sequence starting with vertex x_n , then taking as x_1 the vertex in $Y = \{x_1, x_2, \dots, x_{n-1}\}$ satisfying $f_1 \geq \sum_{i=2}^{n-1} e_{1,i}$, this vertex can topple after x_n since in that case $f_1 + e_{1,n} \geq \sum_{i=2}^n e_{1,i} = d_1$. Then at each step, a vertex x_j such that $f_j \geq \sum_{i=j+1}^{n-1} e_{i,j}$ exists taking $Y = X \setminus \{x_n, x_1, x_2, \dots, x_{j-1}\}$, this vertex can topple at this stage. We have thus built a sequence of topplings proving that f is recurrent. \square

6.1.3 Parking configurations

We consider a kind of dual notion to that of recurrent configuration, such configurations are often called *parking configurations* since in the case of complete graphs, these are exactly the parking functions, a central object in combinatorics.

Definition 31. A sandpile configuration f on a graph G is a parking configuration if for any subset Y of $X \setminus \{x_n\}$ there is a vertex x_k in Y such that f_k is less than the number of edges which are incident to x_k and a vertex out of Y . More precisely if the exists $x_k \in Y$ such that $f_k < \sum_{x_i \notin Y} e_{i,k}$.

In other words a sandpile configuration f is a parking configuration if and only if there is no toppling of all the vertices in a subset Y of $\{x_1, x_2, \dots, x_{n-1}\}$ leaving all the $f_i \geq 0$. For this reason these configurations are also called *superstable* (as for instance in [31]).

Proposition 32. *Let f be a stable configuration and let δ be the configuration such that $\delta_i = d_i - 1$. Define $\beta(f) = \delta - f$. Then f is recurrent if and only if $\beta(f)$ is a parking configuration.*

Proof. It suffices to compare Corollary 30 and Definition 31 and to notice that:

$$d_k = \sum_{x_j \notin Y} e_{k,j} + \sum_{x_j \in Y} e_{k,j} \quad \square$$

Corollary 33. *For any configuration f there exists a unique parking configuration g such that $f \sim_{L_G} g$.*

Proof. For any configuration f let g be the recurrent configuration such that $g \sim_{L_G} \delta - f$ then $\beta(g)$ is a parking configuration such that $f \sim_{L_G} \beta(g)$. \square

In this paper we will often consider the parking configuration in a class as a representative of this class. A parking configuration f in a graph with n vertices will be represented by the subsequence consisting of this first $n - 1$ terms and an integer s such that:

$$(f_1, f_2, \dots, f_{n-1}) \quad s = f_n \quad (2)$$

hence s represents the number of tokens on the distinguished often called "sink" vertex x_n .

Parking configurations and acyclic orientations

An orientation of G is a directed graph obtained from G by orienting each edge, so that one end vertex becomes the head and the other one the tail. A directed path in such a graph consists of a sequence of edges such that the head of an edge is equal to the tail of the subsequent one.

The orientation is acyclic if there is no directed circuit, i.e. a directed path starting and ending at the same vertex. We associate to any parking configuration f an acyclic orientation by :

Proposition 34. *For any parking configuration f on $G = (X, E)$ there exists at least one acyclic orientation \vec{G} such that for any vertex x_i , $i \neq n$, f_i is strictly less than its indegree d_i^- (i.e. the number of edges with head x_i).*

Proof. We orient the edges using an algorithm that terminates after n steps. Consider $Y = \{x_1, x_2, \dots, x_{n-1}\}$. From the definition of parking configurations, there is at least one vertex x_i such that $f_i < e_{i,n}$ then orient all these $e_{i,n}$ edges from x_n (the tail) to x_i (the head), and remove x_i from Y . Repeat the following operation until Y is empty:

- Find x_k in Y such that $f_k < \sum_{x_j \notin Y} e_{k,j}$; orient all the edges joining any vertex j outside Y to x_k from x_j to x_k and remove x_k from Y . \square

In the preceding proof one may recognize a scheduling of topplings related to the Dhar criterion applied to the recurrent configuration $\beta(f)$. Notice that more precise results involving maximal parking configurations are given in [8].

6.2 Effective configurations

In this section we define the notion of L_G -effective configuration and recall the main results of [6], the proofs we give in this section are more or less a reformulation in our terms of the proofs of them given in [6]. The game described in the introduction can be translated in determining if a configuration is L_G -effective with the following definition of effectiveness:

Definition 35. A configuration f is effective if $f_i \geq 0$ for all i . A configuration f is L_G -effective if there exists an effective configuration g toppling equivalent to f (recall that this means $f - g \in L_G$).

Since two equivalent configurations by \sim_{L_G} have the same degree, it is clear that a configuration with negative degree is not L_G -effective. However we will prove that configurations with positive degree are not necessarily L_G -effective as these two examples show:

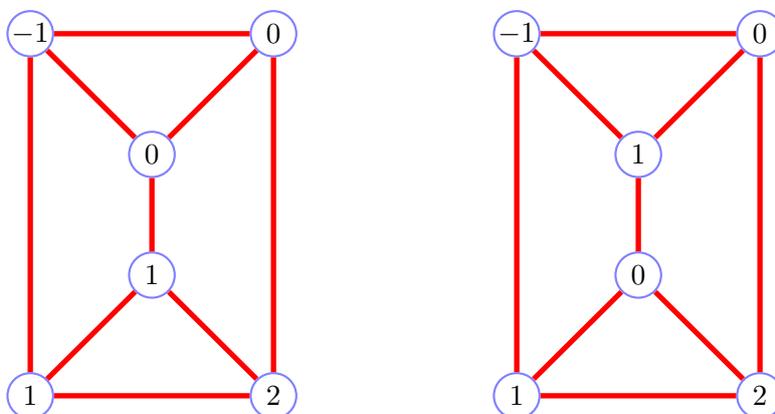


Figure 10: An L_G -effective configuration and a non L_G -effective one

6.2.1 Configuration associated to an acyclic orientation of G

As already seen in Section 6.1.3 an orientation of G is a directed graph obtained from G by orienting each edge, that is distinguishing for each edge with end points x_i and x_j which one is the head and the other being the tail. The orientation is acyclic if there is no directed circuit. Let \vec{G} be an acyclic orientation of G , we define the configuration $C(\vec{G})$ by:

$$(C(\vec{G}))_i = d_i^- - 1$$

Where d_i^- is the number of edges which have head x_i .

Notice that for any acyclic orientation \vec{G} we have:

$$\deg(C(\vec{G})) = \sum_{i=1}^n (d_i^- - 1) = m - n$$

The configuration represented in the right of Figure 10 is equal to $C(\vec{G})$ for the orientation of G represented in Figure 11.

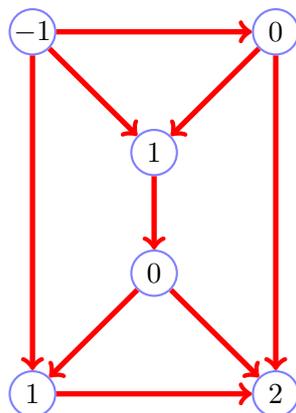


Figure 11: An orientation of G and the corresponding configuration

Proposition 36. *The configuration associated to an acyclic orientation of G is not L_G -effective.*

Proof. Let \vec{G} be an acyclic orientation of G and $f = C(\vec{G})$. We will show that for any linear combination $g = \sum_{i=1}^n a_i \Delta^{(i)}$ the sum h of g and f is not an effective configuration. Let $\varepsilon_{i,j}$ denote the number of edges with head x_j and tail x_i in \vec{G} . Then $e_{i,j} = \varepsilon_{i,j} + \varepsilon_{j,i}$ (but notice that since the orientation is acyclic, at least one of the two values in the sum above is equal to 0).

For any vertex x_i of G we have $d_i^- = \sum_{j=1}^n \varepsilon_{j,i}$ so that:

$$h_i = -1 + \sum_{j=1}^n \varepsilon_{j,i} + a_i d_i - \sum_{j=1}^n a_j e_{i,j}$$

Using $d_i = \sum_{j=1}^n e_{j,i}$ and decomposing each $e_{i,j}$ into $\varepsilon_{i,j} + \varepsilon_{j,i}$ gives for any i :

$$h_i = -1 + \sum_{j=1}^n \varepsilon_{j,i} + a_i \sum_{j=1}^n (\varepsilon_{i,j} + \varepsilon_{j,i}) - \sum_{j=1}^n a_j (\varepsilon_{i,j} + \varepsilon_{j,i}) \quad (3)$$

Separating the edges for which x_i is a head in \vec{G} from those for which it is a tail, we get:

$$h_i = -1 + \sum_{j=1}^n (1 + a_i - a_j) \varepsilon_{j,i} + \sum_{j=1}^n (a_i - a_j) \varepsilon_{i,j} \quad (4)$$

Now take i be such that $a_i \leq a_j$ for all $j \neq i$, we have $a_i - a_j \leq 0$ and $1 + a_i - a_j \leq 1$. If there is a unique minimal value in the sequence a_j we have $1 + a_j - a_i \leq 0$, hence $h_i < 0$.

If there are many a_i 's attaining the minimal value take k among them such that $\varepsilon_{j,k} = 0$ for all the j such that $a_j = a_k$. The existence of such a k follows from the acyclicity of \vec{G} . Then for this k we have $h_k < 0$. \square

6.2.2 Characterization of L_G -effective configurations

We can see that computing the parking configuration toppling equivalent to a given configuration allows to determine if it is effective since we have

Proposition 37. *A configuration f is L_G -effective if and only if the parking configuration g equivalent to f is such that $g_n \geq 0$.*

Proof. If $g_n \geq 0$ g is effective so that f is L_G -effective. If f is L_G -effective then there exists an effective configuration h such that $f \sim_{L_G} h$. Then g is the unique parking configuration such that $h \sim_{L_G} g$, it may be obtained from h performing subset topplings. These do not decrease the value of h_n , hence $g_n \geq h_n \geq 0$. \square

The following Theorem is the central result in [6].

Theorem 38. *For any configuration f one and only one of the following assertions is satisfied:*

- (1) f is L_G -effective
- (2) There exists an acyclic orientation \vec{G} such that $C(\vec{G}) - f$ is L_G -effective.

Proof. Let f be non L_G -effective, consider the parking configuration g equivalent to f and let \vec{G} be the acyclic orientation given by Proposition 34, let $h = C(\vec{G}) - g$. Then for $i \neq n$ we have

$$h_i = d_i^- - 1 - f_i \geq 0$$

and since $g_n < 0$:

$$h_n = -1 + g_n \geq 0.$$

Hence h is effective, moreover since f and g are in the same class, so are $C(\vec{G}) - f$ and $C(\vec{G}) - g = h$ showing that $C(\vec{G}) - f$ is L_G -effective.

Notice that f and $C(\vec{G}) - f$ cannot be both L_G -effective since their sum $C(\vec{G})$ would be too, contradicting Proposition 36. \square

Corollary 39. *Any configuration f with degree greater than $m - n$ is L_G -effective.*

Proof. If f such that $\deg(f) > m - n$ is not L_G -effective, by the above theorem there exists an acyclic orientation \vec{G} of G such that $C(\vec{G}) - f$ is. But the degree of this configuration is negative, giving a contradiction. \square

6.3 The rank of configurations

From now on it will be convenient to denote effective configurations using greek letters λ, μ and configurations with no particular assumptions on them by letters f, g, h .

6.3.1 Definition of the rank

Definition 40. The rank $\rho(f)$ of a configuration f is the integer equal to:

- -1 , if f is non L_G -effective,
- or, if f is L_G -effective, the largest integer r such that for any effective configuration λ of degree r the configuration $f - \lambda$ is L_G -effective.

Denoting \mathbb{P} the set of effective configurations and \mathbb{E} the set of L_G -effective configurations this definition can be given by the following compact formula which is valid in both cases:

$$\rho(f) + 1 = \min_{\lambda \in \mathbb{P}, f - \lambda \notin \mathbb{E}} \deg(\lambda)$$

In other words let f be a configuration of rank $\rho(f)$ and λ be an effective configuration such that $\deg(\lambda) \leq \rho(f)$ then $f - \lambda$ is L_G -effective; moreover there exists an effective configuration μ of degree $\rho(f) + 1$ such that $f - \mu$ is not L_G -effective.

An immediate consequence of this definition is that if $\deg(f) < 0$ or if $f = C(\vec{G})$ for an acyclic orientation \vec{G} then the rank of f is -1 . Moreover if two configurations f and g are such that $f_i \leq g_i$ for all i then $\rho(f) \leq \rho(g)$.

The following notion will be useful to prove properties of the rank:

Definition 41. An effective configuration μ is a proof for the rank $\rho(f)$ of an L_G -effective configuration f if $f - \mu$ is not L_G -effective and $f - \lambda$ is L_G -effective for any effective configuration λ such that $\deg(\lambda) < \deg(\mu)$.

Notice that if λ is a proof for $\rho(f)$ then $\rho(f) = \deg(\lambda) - 1$.

Proposition 42. A configuration f of degree greater than $2m - 2n$ has rank

$$r = \deg(f) - m + n - 1$$

Proof. We first show that for any effective configuration λ such that $\deg(\lambda) = r$, the configuration $f - \lambda$ is L_G -effective. This follows from $\deg(f - \lambda) = \deg(f) - r = m - n + 1$ by Corollary 39.

We now build a effective configuration λ of degree $r + 1$ such that $f - \lambda$ is not L_G -effective. Consider any acyclic orientation \vec{G} of G and let $g = f - C(\vec{G})$ then g is L_G -effective since its degree is equal to $\deg(f) - m + n$ hence greater than $m - n$. Let λ be the effective configuration such that $g \sim_{L_G} \lambda$, then $f - \lambda$ is such that

$$C(\vec{G}) \sim_{L_G} f - g \sim_{L_G} f - \lambda$$

so that $f - \lambda$ is not L_G -effective by Proposition 36. And the Proposition results from:

$$\deg(\lambda) = \deg(g) = \deg(f) - \deg(C(\vec{G})) = \deg(f) - m + n = r + 1 \quad \square$$

6.3.2 Riemann-Roch like theorem for graphs

We give here a proof of the following theorem first proved in [6] which we estimate shorter and simpler than the original one.

Theorem 43. *Let κ be the configuration such that $\kappa_i = d_i - 2$ for all $1 \leq i \leq n$, so that $\deg(\kappa) = 2(m - n)$. Any configuration f satisfies:*

$$\rho(f) - \rho(\kappa - f) = \deg(f) + n - m$$

Proof. The main ingredient for the proof is to use Theorem 38 and remark that for any acyclic orientation \vec{G} the orientation \overleftarrow{G} of G obtained from \vec{G} by reversing the orientations of all the edges is such that: $C(\vec{G}) + C(\overleftarrow{G}) = \kappa$.

Let f be any configuration we first give an upper bound for $\rho(\kappa - f)$, we define λ to be a proof for the rank of f if f is L_G -effective, and to be equal to 0 if f is not L_G -effective. So that $\rho(f) = \deg(\lambda) - 1$ in both cases.

Since $f - \lambda$ is not L_G -effective, we have by Theorem 38 that there exists an acyclic orientation \vec{G} of G such that $C(\vec{G}) - (f - \lambda)$ is L_G -effective, hence equivalent to an effective configuration μ . This may be written as:

$$C(\vec{G}) - (f - \lambda) \sim_{L_G} \mu \tag{5}$$

Now consider the orientation \overleftarrow{G} of G obtained from \vec{G} by reversing the orientations of all the arrows, clearly $C(\vec{G}) + C(\overleftarrow{G}) = \kappa$. Hence adding $C(\overleftarrow{G})$ to both sides of (5) we have:

$$\kappa - (f - \lambda) \sim_{L_G} \mu + C(\overleftarrow{G}) \tag{6}$$

which may be written as:

$$(\kappa - f) - \mu \sim_{L_G} C(\overleftarrow{G}) - \lambda$$

Giving that $\kappa - f - \mu$ is not L_G -effective since the reverse of an acyclic orientation is also acyclic. Hence by the definition of the rank we have

$$\rho(\kappa - f) < \deg(\mu) \tag{7}$$

The degree of μ is obtained from (5) giving:

$$\deg(\mu) = \deg(C(\vec{G})) - \deg(f) + \deg(\lambda) = m - n - \deg(f) + \rho(f) + 1$$

and:

$$\rho(\kappa - f) < m - n - \deg(f) + \rho(f) + 1 \tag{8}$$

Now to obtain a lower bound for $\rho(\kappa - f)$ we exchange the roles of f and $\kappa - f$ giving:

$$\rho(f) < m - n - \deg(\kappa - f) + \rho(\kappa - f) + 1 \tag{9}$$

Since $\deg(\kappa - f) = 2(m - n) - \deg(f)$, inequality (9) may be written as:

$$\rho(f) + m - n - \deg(f) - 1 < \rho(\kappa - f) \tag{10}$$

Comparing inequalities (8) and (10), and noticing that the rank is an integer gives

$$\rho(f) + m - n - \deg(f) = \rho(\kappa - f)$$

hence proving the Theorem. □

6.4 Enumerating the effective configurations

Proposition 44. *Let $T_G(x, y)$ be the Tutte polynomial of the graph G , and let t_i be the integer coefficients given by:*

$$T_G(1, y) = \sum_{i=0}^{m-n+1} t_i y^i$$

Then the number of non equivalent L_G -effective configurations of degree d is given by:

$$\sum_{k=m-n+1-d}^{m-n+1} t_k$$

Proof. In [29] the level of a recurrent configuration f was defined as

$$\text{level}(f) = \sum_{i=0}^{n-1} f_i - m + d_n$$

where d_n is the degree of the vertex x_n .

It was proved that this level varies from 0 to $m - n + 1$ and that the number of recurrent configurations of level p and such that $x_n = q$ does not depend on q and is equal to the coefficient t_p of y^p in the evaluation of the Tutte polynomial $T_G(x, y)$ of G for $x = 1$. A bijective proof of this result was given in [11].

Using the bijection β defined in Proposition 32 we have that the number of parking configurations g such that $\sum_{i=1}^{n-1} g_i = j$ and a given value for g_n is equal to the number of recurrent configurations f such that:

$$\sum_{i=1}^{n-1} f_i = \sum_{i=1}^{n-1} (d_i - 1 - g_i) = 2m - d_n - (n - 1) - j$$

and $f_n = d_n - 1 - g_n$, which is the number of recurrent configurations of level $k = m - n + 1 - j$ and a given value of f_n . This number is equal to t_k .

In order that the configuration g of degree d to be L_G -effective we must have $g_n \geq 0$ so that k must be greater or equal to 0 and not greater than $m - n + 1$, thus ending the proof. □

The generating function for non-equivalent L_G -effective configurations according to the degree counted by the variable y is $\frac{y^{m-n+1}}{1-y} T_G(1, y^{-1})$.