Algorithm for computing the rank of configurations on Wheel Graphs

Robert CORI[1], PHAN Thi Ha Duong[2,*], TRAN Thi Thu Huong[3]

[1] LaBRI, Université de Bordeaux, France

cori@labri.u-bordeaux.fr

[2] ICRTM - Institute of Mathematics, Vietnam Academy of Science and Technology, Vietnam

phanhaduong@math.ac.vn

[3] Vietnamese-German University, Vietnam

huong.ttt@vgu.edu.vn

[*] Corresponding author: phanhaduong@math.ac.vn

Abstract

The rank of configurations on graphs has been extensively studied in combinatorics due to its deep connections with algebraic geometry and chip-firing games. Computing this rank is known to be NP-hard for general graphs, and efficient algorithms exist only for certain specific families. In this paper, we focus on wheel graphs, which consist of a cycle with a central vertex connected to all cycle vertices. We present an algorithm to compute the rank of a configuration $[u, p]$, where u represents the values on the cycle and p is the value at the central vertex.

Our approach allows for the simultaneous computation of the rank for all values of p and provides an explicit proof configuration for each $[u, p]$ - a construction that can be computed iteratively. The main idea is the use of basic operations on 0 - sequences in a superstable configuration on the graph. The proposed algorithm runs in $O(n^2)$ time complexity.

1 Introduction

The rank of divisors on graphs plays an important role in combinatorics because of its deep connections with algebraic geometry. The concept of rank was first introduced in the seminal paper by Baker and Norine, where they established a graph-theoretic analogue of the classical Riemann-Roch theorem [1]. This work was motivated by the study of tropical geometry and chip-firing games, which provide a combinatorial framework for understanding the theory of divisors in graphs.

In this context, a divisor on a graph is simply a function from the vertex set to the set of integers $\mathbb{Z}$, which corresponds to a chip configuration in the well-known chip-firing game [3, 4, 8]. In this paper, we will use the terminology "configuration" instead of "divisor". Calculating the rank of a configuration involves solving a max-min optimization problem, which is naturally difficult. In fact, it has been proven that computing the rank of a configuration is NP-hard for general graphs [10].

The theory of configuration rank is based on the linear equivalence of configurations and the properties of the Laplacian operator. Furthermore, it is closely related to the critical group of a graph, a widely studied algebraic structure in graph theory. Various approaches have been explored to approximate the rank of a configuration, such as those in [5]. Baker and Shokrieh [2] provided an algorithm that can efficiently check whether the rank of a configuration on a graph is at least c for any fixed constant c. However, exact computations remain limited. Efficient algorithms have been developed for specific families of graphs, such as paths, cycles, and trees. Notably, a linear-time algorithm was obtained for complete graphs in [7].

In this paper, we focus on the family of wheel graphs W_{n+1}, which consist of a cycle C_n of length n and a central vertex connected to all vertices of the cycle. Wheel graphs have been widely studied due to their applications in network design, shortest path algorithms, Hamiltonian cycle detection, and planar graph theory. Their high connectivity, small diameter, and symmetric structure make them an interesting case for efficient algorithmic analysis; many computational problems that are NP-hard on general graphs become more tractable on wheel graphs [6, 9, 11, 13]. Moreover, the critical group of wheel graphs was studied by Rossin in [12], where he showed that it is the product of two cyclic groups, whose orders are determined by Fibonacci numbers.

In this paper, we conduct a detailed analysis of superstable configurations (which are in bijection with critical configurations) on wheel graphs and study the effect of the Laplacian operator on these configurations. Typically, to determine the rank $\rho(D)$ of a configuration D, one must also provide a proof of a function λ of degree $r + 1$, which is often difficult to construct.

Our main contributions are as follows. We present an algorithm to compute the rank of a configuration $[u, p]$, where u represents the values on the cycle C_n and p is the value at the central vertex. We compute the rank for all values of p simultaneously. We provide an explicit construction of the proof for all configurations and show that it can be computed iteratively. The main idea is the use of basic operations on O - sequences in superstable configurations. Our algorithm has a computational complexity of $O(n^2)$.

The structure of this paper is as follows:

- Section 2 presents the theory of chip-firing games on graphs, focusing on Laplacian operators and the rank of configurations.
- Section 3 introduces wheel graphs and discusses their configurations.
- Section 4 provides a detailed analysis of superstable configurations on wheel graphs and a precise description of our algorithm.

2 Rank of Configuration on graph

2.1 Configurations on graphs

For a reference on this subsection, the reader may refer to the paper by Björner, Lovász, and Shor [4].

Graphs

Let $G = (X, E)$ be a connected undirected graph with n vertices, where $X = \{x_1, x_2, \ldots, x_n\}$ is the vertex set and E the set of edges. We consider also that E is defined by as a symmetric matrix also denoted E such that $e_{i,j} = 1$ if there is an edge between the vertices x_i and x_j, and $e_{i,j} = 0$ if not. We assume that G is connected and has no loops, so that $e_{i,i} = 0$ for all i. In all the paper (except for W_{n+1}) n will denote the number of vertices of a graph m the number of edges and d_i for $i \leq n$ denotes the number of neighbors of vertex x_i, that is $d_i = \sum_{i=1}^{n} e_{i,j}$, it is also called the degree of the vertex x_i in G.

2.2 Configurations on graphs

For a reference on this subsection, the reader may refer to the paper by Björner, Lovász, and Shor [4].

Graphs

Let $G = (X, E)$ be a connected undirected graph with n vertices, where $X = \{x_1, x_2, \ldots, x_n\}$ is the vertex set and E is the set of edges. We also consider that E is defined as a symmetric matrix, also denoted E, such that

$e_{i,j} = 1$ if there is an edge between the vertices x_i and x_j, and $e_{i,j} = 0$ otherwise. We assume that G is connected and has no loops, so $e_{i,i} = 0$ for all i. Throughout this paper (except for W_{n+1}), n denotes the number of vertices in a graph, m the number of edges, and d_i for $i \leq n$ denotes the number of neighbors of vertex x_i, i.e., $d_i = \sum_{j=1}^{n} e_{i,j}$. This is also called the degree of vertex x_i in G.

Configurations

We consider *configurations* on graphs; these are elements of the discrete lattice $\mathbb{Z}^n$, where $\mathbb{Z}$ is the set of integers (positive or negative). For a configuration $u = (u_1, u_2, \ldots, u_n)$, we refer to the u_i's as the *entries* of u and say that the integer u_i is assigned to vertex x_i. The symbol $\epsilon^{(i)}$ denotes 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 a configuration u is the sum of its entries and is denoted by $\deg(u)$.

To each vertex is associated a *Laplacian configuration* Δ_i, given by:

$$\Delta_i = d_i \epsilon^{(i)} - \sum_{j=1}^{n} e_{i,j} \epsilon^{(j)}.$$

The set of Laplacian configurations defines the Laplacian matrix of the graph.

$\mathcal{L}_G$-equivalence

We denote by $\mathcal{L}_G$ the subgroup of $\mathbb{Z}^n$ generated by the Δ_i. Two configurations u and v are said to be $\mathcal{L}_G$*-equivalent* if $u - v \in \mathcal{L}_G$, which we also write as $u \sim_{\mathcal{L}(G)} v$.

In the sandpile model and the chip-firing game, the transition from configuration u to configuration $u - \Delta_i$ is allowed only if $u_i \geq d_i$ and is called a toppling. Notice that in the definition of $\mathcal{L}_G$-equivalence, we omit this condition and allow topplings even if $u_i < d_i$.

Effectiveness

The notion of $\mathcal{L}_G$-effectiveness, introduced in [1], is central to this presentation.

Definition 2.1. A configuration u is effective if $u_i \geq 0$ for all i. A configuration u is $\mathcal{L}_G$-effective if there exists an effective configuration v that is toppling-equivalent to u (i.e., $u - v \in \mathcal{L}_G$).

Since two $\mathcal{L}_G$-equivalent configurations have the same degree, it is clear that a configuration with negative degree is not $\mathcal{L}_G$-effective.

Sandpile configurations

We now assume that some vertex of the graph is selected and called the sink. For simplicity of notation, we assume this vertex is x_n. A configuration u is called a *sandpile* if $u_i \geq 0$ for all $i < n$. For a configuration u, a *toppling* of vertex x_i for $i < n$ is performed if $u_i \geq d_i$, and u is replaced by $u - \Delta_i$.

Since the graphs considered are connected, it is not difficult to prove that for any configuration u, there exists a sequence of topplings leading to a configuration v where no further toppling is possible. Clearly, v is such that $v \sim_{\mathcal{L}(G)} u$.

The following proposition has a simple proof:

Proposition 2.1. *For any configuration u, there exists an integer k such that $u - k\Delta_n$ gives a sandpile configuration by performing a sequence of (allowed) topplings.*

Notice that the sandpile configuration obtained is not unique.

Stability

In sandpile configurations, stability is defined as follows:

Definition 2.2. A sandpile configuration u is stable if for any i, $u - \Delta_i$ is not a sandpile configuration. It is superstable if, for any nonempty subset I of $\{1, 2, \ldots, n-1\}$, the configuration $u - \sum_{i \in I} \Delta_i$ is not a sandpile configuration.

Notice that stability may be defined by the condition $0 \leq u_i < d_i$ for all i. It is more difficult to check whether a configuration is superstable. These are often called parking configurations. The following results have simple statements but complicated proofs, which may be found in [1].

Theorem 2.2. *For any configuration u, there exists a unique superstable configuration $\hat{u}$ such that $u \sim_{\mathcal{L}(G)} \hat{u}$.*

Theorem 2.3. *A configuration u is effective if and only if the superstable configuration $\hat{u}$, which is $\mathcal{L}_G$-equivalent to it, satisfies $\hat{u}_n \geq 0$.*

Terminology Review

Let us recall all notions related to configurations. Let G be a graph with $V(G) = \{x_1, x_2, \ldots, x_n\}$. Let $u = (u_1, \ldots, u_n)$. We define:

- u is a **configuration** if $u_i \in \mathbb{Z}$ for all $1 \leq i \leq n$;
- u is **effective** if $u_i \geq 0$ for all $1 \leq i \leq n$;
- u is a **sandpile configuration** if $u_i \geq 0$ for all $1 \leq i \leq n-1$;
- u is $\mathcal{L}(G)$-**effective** if there exists an effective v such that $u \sim_{\mathcal{L}(G)} v$;
- u is **stable** if $u_i < d_i$ for all $1 \leq i \leq n-1$;
- u is **superstable** if for any non-empty subset $I \subseteq \{1, 2, \ldots, n-1\}$, $u - \sum_{i \in I} \Delta^{(i)}$ is not effective.

2.3 The rank of configurations

For a reference on this subsection, the reader may refer to the paper by Baker and Norine [1].

Definition

5 The notion of rank aims to determine how far a configuration is from being $\mathcal{L}(G)$-effective. This is done by subtracting effective configurations from it and checking whether it remains $\mathcal{L}(G)$-effective.

Definition 2.3. The rank $\rho(u)$ of a configuration u is the integer defined as follows:

- -1 if u is not L_G-effective,
- or, if u is $\mathcal{L}(G)$-effective, the largest integer r such that for any effective configuration λ of degree r, the configuration $u - \lambda$ is $\mathcal{L}(G)$-effective.

Denoting $\mathbb{P}$ as the set of effective configurations and $\mathbb{E}$ as the set of $\mathcal{L}(G)$-effective configurations, this definition can be expressed compactly as follows, which holds in both cases:

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

An immediate consequence of this definition is that if two configurations u and v satisfy $u_i \leq v_i$ for all i, then $\rho(u) \leq \rho(v)$. Moreover, if $u = v - \varepsilon^{(i)}$, then

$$\rho(v) - 1 \leq \rho(u) \leq \rho(v).$$

The following notion will be useful in proving properties of the rank:

Definition 2.4. An effective configuration μ is a *proof* for the rank $\rho(u)$ of an $\mathcal{L}(G)$-effective configuration u if $u - \mu$ is not $\mathcal{L}(G)$-effective and $u - \lambda$ is $\mathcal{L}(G)$-effective for any effective configuration λ such that $\deg(\lambda) < \deg(\mu)$.

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

Proposition 2.4. *If λ is a proof for the rank of u, and μ is an effective configuration such that $\mu_i \leq \lambda_i$ for all $i = 1, 2, \ldots, n$, then $\lambda - \mu$ is a proof for the rank of $v = u - \mu$.*

Proof. The condition $\mu_i \leq \lambda_i$ implies that $\lambda - \mu$ is an effective configuration. Moreover, we have that $v - (\lambda - \mu) = (u - \lambda)$, hence $v - (\lambda - \mu)$ is not $\mathcal{L}(G)$-effective. Now let ν be an effective configuration such that $\deg(\nu) < \deg(\lambda - \mu)$, then $\mu + \nu$ is such that

$$\deg(\mu + \nu) < \deg(\mu) + \deg(\lambda - \mu) = \deg(\lambda),$$

hence $u - \mu - \nu$ is $\mathcal{L}(G)$-effective, which may also be written as $(u - \mu) - \nu$, proving the result. □

The rank of the sum of two configurations

We now consider the question of finding a relationship between the rank of two configurations and the rank of their sum. When both are not $\mathcal{L}(G)$-effective, their sum may also not be $\mathcal{L}(G)$-effective (for example, when their degrees are negative) or it may be $\mathcal{L}(G)$-effective.

Take, for example, $u = (0, 1, 1; -1)$ and $v = (1, 1, 0; -1)$ on the wheel graph W_3. In both cases, $\rho(u+v) > \rho(u)+\rho(v)$. If one is $\mathcal{L}(G)$-effective and the other is not, then the relation does not always hold. For instance, on W_3, for $u = (0, 1, 1; -5)$ and $v = (2, 1, 0; 2)$, we have $\rho(u) = \rho(u + v) = -1$ and $\rho(v) = 2$.

If both are $\mathcal{L}(G)$-effective, we prove:

Proposition 2.5. *Let u and v be two $\mathcal{L}(G)$-effective configurations. Then their ranks $\rho(u)$ and $\rho(v)$ satisfy the following relation:*

$$\rho(u + v) \geq \rho(u) + \rho(v). \tag{1}$$

When adding an effective configuration λ to a configuration u, we obtain an upper bound for the rank of $\lambda + u$:

Proposition 2.6. *Let u be any configuration and λ be an effective configuration. Then*

$$\rho(u + \lambda) \leq \rho(u) + \deg(\lambda). \tag{2}$$

Baker-Norine Formula

In their seminal paper, Baker and Norine introduced a configuration κ defined by

$$\kappa_i = d_i - 2,$$

for every vertex i. They established a relation that allows computing the rank of u given the rank of $\kappa - u$. More precisely:

Theorem 2.7. *Let u be a configuration on the connected graph G, and let κ be defined as above. Then the ranks $\rho(u)$ and $\rho(\kappa - u)$ satisfy:*

$$\rho(u) = \rho(\kappa - u) + \deg(u) - m + n, \tag{3}$$

where $\deg(u)$ *is the degree of the configuration u, and m and n are the number of edges and vertices of G, respectively.*

We will use this theorem to compute the rank of configurations with high degree. We have:

Corollary 2.8. *If the degree of u is greater than $2(m - n)$, then*

$$\rho(u) = \deg(u) - (m - n + 1).$$

Proof. Since the degree of κ is given by $\deg(\kappa) = \sum_{i=1}^{n}(d_i - 2)$, and using the fact that $\sum_{i=1}^{n} d_i = 2m$, we obtain $\deg(\kappa) = 2m - 2n$.

Thus, if $\kappa - u$ has negative degree, it follows that $\rho(\kappa - u) = -1$. Applying Baker and Norine's theorem, we obtain the result. $\square$

Corollary 2.9. *For any configuration u, the rank $\rho(u)$ satisfies*

$$\rho(u) \geq \deg(u) - (m - n + 1).$$

Proof. Regardless of the degree of u, we always have $\rho(\kappa - u) \geq -1$. The result follows directly from Theorem 2.7. $\square$

3 The wheel graphs and their superstable configurations

The wheel graph W_{n+1} has $n+1$ vertices, denoted by $x_1, x_2, \ldots, x_n, x_0$, and $2n$ edges, consisting of: A cycle of length n: $x_1, x_2, \ldots, x_n$, and n edges connecting x_0 to each x_i for $1 \leq i \leq n$.

A configuration on W_n is written as $[u, p]$, where $u = (u_1, \ldots, u_n)$ corresponds to the values of u on the cycle, and p is the value at x_0.

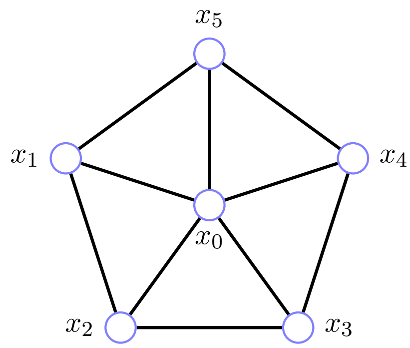

Figure 1: The Wheel Graph W_5

In the wheel graph almost all the Laplacian configurations have all the same structure only four entries are different from 0. Three of them may be considered as consecutive in the circular order $u_1, u_2, \ldots, u_n$ they get the values $-1, 3, -1$ and $u_0 = -1$. The last Laplacian configuration is equal to $[-1, -1, \ldots, -1, n]$.

The vertex x_0 plays a central role in the wheel graph. Hence we select it in order to define sand pile and superstable configurations. For any configuration $[u, p]$ it is easy to find a sandpile configuration $\mathcal{L}(G)$-equivalent to it. Indeed, if $[u, p]$ is not a sand pile configuration, let u_i be the minimal entry of u, then $[u, p] - u_i\Delta^{(0)}$ is a sand pile configuration $\mathcal{L}(G)$-equivalent to $[u, p]$.

We can characterize the superstable configurations on W_{n+1} as follows:

Proposition 3.1. *A sandpile configuration $[u, p]$ of W_{n+1} is superstable if and only if and only if the following conditions are satisfied*

- $0 \leq u_i \leq 2$ *for all* $1 \leq i \leq n$
- *There exists at least one* $i \in \{1, 2, \ldots, n\}$ *such that* $u_i = 0$.

- *If for $i < j$ we have $u_i = u_j = 2$ then there exists $\ell_1 \in \{i+1, \ldots, j-1\}$ and $\ell_2 \in \{1, \ldots, i-1\} \cup \{j+1, \ldots, n\}$ such that $u_{\ell_1} = u_{\ell_2} = 0$. (In other words, between any two consecutive occurrences of the number 2 in the cycle, there is at least one occurrence of the number 0.)*

Proof. We prove first that these conditions are necessary for $[u,p]$ to be superstable. If there exists i such that $u_i > 2$, then $[u,p]$ is not stable. If $u_i > 0$ for all i, then the set $I = \{1, 2, \ldots, n\}$ is such that $[v,q] = [u,p] - \sum_{i \in I} \Delta^{(i)}$ is a sand pile configuration, since $v_j = u_j - 1$ for all $1 \leq j \leq n$. The same argument holds for the third condition using $I_1 = \{i, i+1, \ldots, j\}$ or $I_2 = \{1, \ldots, i\} \cup \{j, \ldots, n\}$ to index the sum of the $\Delta^{(i)}$.

Conversely, consider $[u,p]$ that is not superstable. Then, there exists a subset $I \subset \{1, 2, \ldots, n\}$ such that

$$[v,q] = [u,p] - \sum_{i \in I} \Delta^{(i)}$$

is not a sandpile configuration.

By the first two conditions, I has more than one element and is not equal to $\{1, 2, \ldots, n\}$. If $i \in I$ and $u_i = 2$, then i must have at least one neighbor in I. If $u_i = 1$, then i must have at least two neighbors in I.

Thus, there exists a sequence of vertices forming a path, where the first vertex x_i and the last vertex x_j satisfy $u_i = u_j = 2$, and all vertices x_ℓ in the path satisfy $u_\ell > 0$, contradicting the last condition.

□

We can then state that a maximal superstable configuration is the one with the largest sum, and we have the following corollary:

Corollary 3.2. *A sandpile configuration is a maximal superstable configuration if it is superstable, and between any two consecutive occurrences of the number 2 on the cycle, there is exactly one occurrence of the number 0.*

The configurations in the left of Figure 2 is a superstable configurations and is not effective by Theorem 2.3. The second one is not superstable it is $\mathcal{L}(G)$-equivalent to configuration given by $(0, 2, 0, 0, 0, 0)$ and hence effective.

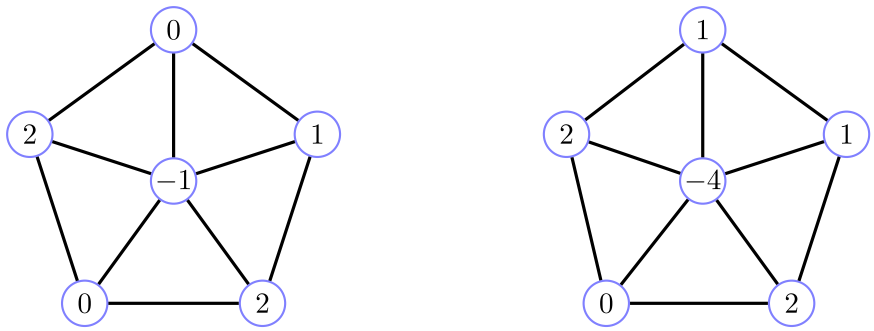

Figure 2: Non effective and effective configurations on W_5

We have the following important result for the wheel graph.

Corollary 3.3. *For any maximal superstable configuration* $[u, p]$ *with* $p \geq 0$*, the rank* $\rho([u, p])$ *is equal to* p*.*

Proof. By Corollary 2.9, we have

$$\rho([u, p]) \geq \deg((u, p)) - (m - n + 1) = k + p - k = p.$$

On the other hand, if we take $\lambda = (p+1)\epsilon_n$, then $\deg(\lambda) = p + 1$, and

$$[u, p] - \lambda = [u, -1].$$

Since $[u, p]$ is already superstable, $[u, -1]$ is not $\mathcal{L}(G)$-effective. This implies that the rank of $[u, p]$ is strictly smaller than $p+1$. Hence, we conclude that $\rho([u, p]) = p$. $\square$

Note that because the definition is independent of u_0, if $[u, p]$ is superstable (resp. maximal superstable), we also say that u is superstable (resp. maximal superstable). Moreover, if u is maximal superstable, then $\sum_{i=1}^{n} u_i = n$.

4 Algorithm

4.1 A greedy algorithm

For a general graph with n vertices, we have seen that a configuration u is $\mathcal{L}(G)$-effective if (and only if) the superstable configuration $\hat{u}$ associated to it satisfies $\hat{u}_n \geq 0$. This value $\hat{u}_n$ will allow to determine the rank. Let us denote it $\pi(u)$.

In order to determine the rank of a configuration one has to determine an effective configuration λ with minimal degree such that $u-\lambda$ is not effective, that is $\pi(u-\lambda) < 0$. This search may be decomposed in a sequence of steps. At each step j one has to obtain an integer i_j such that $\pi(u-\varepsilon^{(i_j)})$ is no more than $\pi(u)$, then u is set equal to $u-\varepsilon^{(i_j)}$. After some steps the value of $\pi(u)$ becomes negative . Let s be the number of steps to obtain a negative value for $\pi(u)$ then taking $\lambda = \sum_{j=1}^{s} \varepsilon^{(i_j)}$ we get that $u - \lambda$ is not $\mathcal{L}(G)$-effective. Hence $s-1$ is an upper bound for the rank.

One can improve the informal algorithm above to have a greedy algorithm obtaining λ such that $u-\lambda$ is not $\mathcal{L}(G)$ effective, the degree of such λ is an upper bound for the rank $\rho(u)$. In some particular families of graphs this algorithm gives exactly $1+\rho(u)$, this is the case for complete graphs for instance (see [7]) .

Definition 4.1. The transformation associating to the configuration $u = [u', \pi(u)]$ the configuration $v = u - \varepsilon^{(i)}$ is called a basic operation and is denoted O_i. Moreover let $[v', p']$ the superstable configuration $\mathcal{L}(G)$-equivalent to v then we will also write $v' = O_i(u')$.

A greedy algorithm giving an upper bound of the rank

Input: A configuration u on G

- $u \leftarrow \hat{u}$;
- $s \leftarrow 0$; $\lambda \leftarrow 0$;
- While $u_n \geq 0$ do
  - Find $i \neq n$ such that $\pi(u - \varepsilon^{(i)})$ is minimal;
  - $u \leftarrow O_i(u)$; $u \leftarrow \hat{u}$
  - $\lambda \leftarrow \lambda + \varepsilon^{(i)}$;
  - $s \leftarrow s + 1$;
- od;

  Output: $s-1, \lambda$.

4.2 A Greedy Algorithm for the Rank on the Wheel Graphs

The greedy algorithm described above for general graphs may be specialized for wheel graphs since it is easy to determine, for a superstable configuration

$[u,p]$, the values of i such that $\pi(u) - \pi(O_i(u))$ is maximal.

On the cycle of the wheel W_{n+1}, we adopt the convention that indices are taken modulo n. This means that x_{n+1} indicates x_1, and more generally, x_{k+l} indicates x_t, where $t = (k+l) \mod n$.

A maximal 0-sequence in a configuration u is given by a sequence of integers $i_0, i_0+1, \ldots, i_0+\ell, i_0+\ell+1$ such that $u_j = 0$ for $i_0+1 \leq j \leq i_0+\ell$ and $u_{i_0} > 0, u_{i_0+\ell+1} > 0$. We define the length of this sequence to be ℓ. By convention, the 0-configuration in W_{n+1} has a maximal 0-sequence of length n. We also write 0^ℓ for a 0-sequence of length ℓ.

Proposition 4.1. *Let $[u,p]$ be a configuration on W_{n+1} and let $[v,q] = O_i([u,p])$. Then, we have:*

- *If $u_i > 0$, then $v_i = u_i - 1$ and $q = p$;*
- *If $u_i = 0$, then $q = p - \ell$, where ℓ is the length of the maximal 0-sequence containing i.*

Proof. The first assertion is clear.

For the second one, we need to obtain the superstable configuration $\mathcal{L}(G)$-equivalent to $[u,p] - \epsilon^{(i)}$. For that, we denote $I = \{i_0+1, \ldots, i_0+\ell\}$ the maximal 0-sequence containing i, and we consider the values $a = u_{i_0}$ and $b = u_{i_0+\ell+1}$, which delimit I. We compute

$$[v,q] = [u,p] - \epsilon_{(i)} - \sum_{j \notin I} \delta_j.$$

We interpret this subtraction as a toppling $\epsilon^{(i)}$ and all vertices not in I. Let us write u as $(*a0^\ell b*)$, where $*$ represents some suitable sequence. Three cases are now possible: $i_0+1 < i < i_0+\ell$, $i = i_0+1$ or $i = i_0+\ell$. By simple computation, we obtain:

- If $i_0+1 < i < i_0+\ell$, then
$$[v,q] = [*(a-1)21^{i-i_0}01^{i_0+\ell-i-1}2(b-1)*, p-\ell].$$
- If $i = i_0+1$, then
$$[v,q] = [*(a-1)1^{\ell-1}2(b-1)*, p-\ell].$$
- If $i = i_0+\ell$, then
$$[v,q] = [*(a-1)21^{\ell-1}(b-1)*, p-\ell].$$

In any case, $[v,q]$ is a superstable configuration, which implies the statement. □

The greedy algorithm

Input: A configuration u on the Wheel graph W_{n+1}

- $u \leftarrow \hat{u}$;
- $s \leftarrow 0$; $\lambda \leftarrow 0$;
- While $u_0 \geq 0$ do
  - Find $i \neq n$ such that $u_i = 0$ such that i is among a 0-sequence of maximum length among all the maximal 0-sequences.
  - $u \leftarrow O_i(u)$; $u \leftarrow \hat{u}$
  - $\lambda \leftarrow \lambda + \varepsilon^{(i)}$;
  - $s \leftarrow s + 1$;
- od; **Output:** $s - 1, \lambda$.

This algorithm does not always provide the correct rank of the configuration; it needs to be improved in certain aspects. If there exist multiple maximal 0-sequences of maximum length, one must select the sequence that satisfies a specific condition. We will analyze this condition in the next section.

Example 4.1. There are 45 possible u such that $[u, p]$ is a superstable configurations in the wheel graph W_4. These may be represented by 11 configurations performing rotations or reversions on them. We give for these the sequences of ranks $r_p = \rho([u, p])$. The last three are superstable configurations.

u	r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
(0,0,0,0)	0	0	0	0	1	2	3	3	4
(0,0,0,1)	0	0	0	0	1	2	3	4	5
(0,0,1,1)	0	0	1	1	2	3	4	5	6
(0,1,0,1)	0	1	1	1	2	3	4	5	6
(0,0,0,2)	0	0	0	1	2	3	4	5	6
(0,1,1,1)	0	1	2	2	3	4	5	6	7
(0,0,1,2)	0	0	1	2	3	4	5	6	7
(0,1,0,2)	0	1	1	2	3	4	5	6	7
(0,1,1,2)	0	1	2	3	4	5	6	7	8
(0,1,2,1)	0	1	2	3	4	5	6	7	8
(0,2,0,2)	0	1	2	3	4	5	6	7	8

and the sequence of $\pi_p(u)$

u	$\pi_0(u)$	$\pi_1(u)$	$\pi_2(u)$	$\pi_3(u)$	$\pi_4(u)$	$\pi_5(u)$
(0,0,0,0)	0	4	5	6	8	9
(0,0,0,1)	0	4	5	6	7	8
(0,0,1,1)	0	2	4	5	6	7
(0,1,0,1)	0	1	4	5	6	7
(0,0,0,2)	0	3	4	5	6	7
(0,1,1,1)	0	1	2	4	5	6
(0,0,1,2)	0	2	3	4	5	6
(0,1,0,2)	0	1	3	4	5	6
(0,1,1,2)	0	1	2	3	4	5
(0,1,2,1)	0	1	2	3	4	5
(0,2,0,2)	0	1	2	3	4	5

4.3 Algorithm

We consider the following problem. Let $[u, p]$ be a superstable configuration. Our goal is to compute the rank $r = \rho([u, p])$.

Let us recall that $r = k$ if there exists a proof λ of degree $k + 1$ such that $[u, p] - \lambda$ is not $\mathcal{L}(G)$-effective, and for all μ of degree k, $[u, p] - \mu$ is $\mathcal{L}(G)$-effective.

We can then state the following scheme:

1. $r = 0$ if $p < \ell_1$, where ℓ_1 is the largest value such that there exists $\lambda_1 = \epsilon^{(i)}$ and $[u, p] - \lambda_1 \sim_{\mathcal{L}(G)} [v, p - \ell_1]$.

2. $r = 1$ if $\ell_1 \leq p < \ell_2$, where ℓ_2 is the largest value such that there exists λ_2 of degree 2 and $[u, p] - \lambda_2 \sim_{\mathcal{L}(G)} [v, p - \ell_2]$.

3. and so on.

4. $r = k$ if $\ell_k \leq p < \ell_{k+1}$, where ℓ_{k+1} is the largest value such that there exists λ_{k+1} of degree $k + 1$ and $[u, p] - \lambda_{k+1} \sim_{\mathcal{L}(G)} [v, p - \ell_{k+1}]$.

Then, our algorithm aims to find all configurations λ_i (and at the same time, all values ℓ_i) to determine $\rho([u, p])$. This algorithm will give the rank of all configurations $[u, p]$ for a given u and for all values of p.

Before describing the idea of our algorithm, let us first present some examples to illustrate it.

Example 4.2. Recall that, when one applies the basic operation on position i of a 0-sequence, one has:

$$[*a0^m b*, p] \xrightarrow{O_i} [*(a-1)1^{m-1}2(b-1), p-m].$$

Let $[u = 2000001200020011, p]$ be a superstable configuration.
We consider $i = 2$ and $j = 9$, and two operations O_2 and O_9:

$$[2000001200020011 - \epsilon^{(2)}, p] \sim_{\mathcal{L}(G)} [1111120200020011, p-5],$$
$$[2000001200020011 - \epsilon^{(9)}, p] \sim_{\mathcal{L}(G)} [2000001111210011, p-3].$$

Hence, to find the largest value ℓ_1 such that there exists $\lambda_1 = \epsilon^{(i)}$ and

$$[u, p] - \lambda_1 \sim_{\mathcal{L}(G)} [v, p - \ell_1],$$

we must choose a 0-sequence of maximum length, ensuring that the index i is inside this sequence.

We also note that this index i can take values $2, 3, 4, 5$, or 6, meaning it can be any position inside this 0-sequence. For simplicity, we choose the first position of the sequence.

We let m be the maximal length of a 0-sequence, and put $M = 0^m$. We call the motif T, the desired sequence of maximizing ℓ. Hence, for ℓ_1, $T = M = 0^m$.

Definition 4.2. Let $[u, p]$ be a superstable configuration on a Wheel graph. Let m be the greatest length of a 0-sequence in u. For any sequence A of the form $A = 0^m 1*$, we define

$$\tilde{A} = 10^{m-1}1*,$$

which is obtained from A by replacing the first 0 in A with 1.

Lemma 4.2. *Let $[u, p]$ be a superstable configuration. Let $A = 0^{m_1}10^{m_2}\dots10^{m_k}$ be a sequence in u, $u = *aAb*$. We define the basic operation O_A as the sequence of basic operations $O_{i_1}, O_{i_2}, \dots, O_{i_k}$, where: for $1 \le j \le k$, i_j is the first position of 0 in A after applying consecutively the operations $O_{i_1}, O_{i_2}, \dots, O_{i_{j-1}}$.*

Then we have:

$$
\begin{aligned}
[u,p] =& [*a0^{m_1}10^{m_2}10^{m_3}\dots 10^{m_k}b*, p] \\
\xrightarrow{O_{i_1}}& [*(a-1)1^{m_1-1}20^{m_2+1}10^{m_3}\dots 10^{m_k}b*, p-m_1] \\
\xrightarrow{O_{i_2}}& [*(a-1)1^{m_1-1}11^{m_2}20^{m_3+1}\dots 10^{m_k}b*, p-(m_1+m_2+1)] \\
\dots& \\
\xrightarrow{O_{i_k}}& [*(a-1)\ 1^{m_1+1+m_2+\ldots+1+m_{k-1}+m_k}\ (b-1)*,\ p-(m_1+m_2+\ldots+m_k+k-1)]. \\
=& [*(a-1)\ 1^{m_1+m_2+\ldots+m_k+k-2}\ (b-1)*,\ p-(m_1+m_2+\ldots+m_k+k-1)].
\end{aligned}
$$

Our main idea is to find the motif T in u, then apply the operation O_T on u, and continue this process iteratively by identifying the motif in the resulting configuration. This procedure is repeated until we obtain a maximal superstable configuration. The method for determining T is illustrated in the following.

Example 4.3. 1. Let $u = 20000012000200112000001000001000002101$.

By the above result, to obtain ℓ_1, one can choose any index i in any 0-sequence of length 5. There are 4 such sequences, beginning at position $2, 18, 24, 34$.

However, finding λ_2 (and ℓ_2) is not straightforward.

Consider $\mu = \epsilon^{(2)} + \epsilon^{(18)}$ and $\lambda = \epsilon^{(18)} + \epsilon^{(23)}$. We have:

$$
\begin{aligned}
[u,p] &= [20000012000200112000001000001000002101, p] \\
&\xrightarrow{O_2} [11112020002001120000010000010000002101, p-5] \\
&\xrightarrow{O_{18}} [1111202000200111111120000000100000100002101, p-10].
\end{aligned}
$$

$$
\begin{aligned}
[u,p] &= [20000012000200112000001000001000002101, p] \\
&\xrightarrow{O_{18}} [2000001200020011111112000000100000100002101, p-5]. \\
&\xrightarrow{O_2} [200000120020011111111111111111120000002101, p-11].
\end{aligned}
$$

We observe that to obtain a maximal ℓ_1, it suffices to choose a 0^m-sequence. However, to obtain the maximal ℓ_2, simply choosing two arbitrary 0^m-sequences is not sufficient. Instead, it is better to select a sequence of the form 0^m10^m. Hence, $T = 0^m10^m$.

The reasoning is as follows: if we apply the basic operation in i, the first position of the first 0^m-sequence, we create a 0^{m+1}-sequence, as shown below:

$$[a * 0^m 10^m b*, p] \xrightarrow{O_i} [*(a-1)1^{m-1}20^{m+1}b*, p-m].$$

On the other hand, the value

$$\ell_2 = m + (m+1)$$

is actually the maximal value for ℓ_2. This is because if we choose any $\lambda = \epsilon^{(i)} + \epsilon^{(j)}$, then O_i will give a maximal value of m and O_j will provide a maximal value of $m+1$.

Similarly, to obtain a maximal ℓ_3, it is preferable to take a sequence of the form $0^m 10^m 10^m$, which results in:

$$\ell_3 = m + (m+1) + (m+1).$$

Thus, to construct the maximal value of ℓ_i for all levels i, we choose the greatest k such that there exists a sequence of the form:

$$T = \max_k 0^m (10^m)^k,$$

and

$$\ell_{i+1} = m + (m+1) \cdot i, \quad \text{for all } i = 0, 1, 2, \ldots, k.$$

2. Let $u = 20^5 120^3 20^2 1120^5 10^4 10^4 10^5 2101$.

   Here, $M = 0^5, \tilde{M} = 10^4$. There is no $M1M$, so we cannot apply the above idea. Moreover, we can easily check that in any case, the maximal ℓ_2 is equal to $m + m$, because for any choice of i, the basic operation O_i cannot create a sequence 0^{m+1}.

   Similarly, for ℓ_3, for any choice of i_1 and i_2, the basic operation O_{i_1} followed by O_{i_2} cannot create a sequence 0^{m+1}.

   Nevertheless, it is still not simple to choose λ_4. Put $A = 0^5 10^4 10^4 10^5$, and compare $[u,p] \xrightarrow{O_A}$ then $\xrightarrow{O_2}$ with $[u,p] \xrightarrow{O_2}$ then $\xrightarrow{O_A}$.

$$\begin{aligned}
[u,p] &= [20^5120^320^21120^510^410^410^52101, p] \\
&\xrightarrow{O_{18}} [20^5120^320^21111^4200^410^410^52101, p-5] \\
&\xrightarrow{O_{23}} [20^5120^320^21111^411^4200^410^52101, p-10] \\
&\xrightarrow{O_{28}} [20^5120^320^21111^411^411^4200^52101, p-15] \\
&\xrightarrow{O_{33}} [20^5120^320^21111^411^411^411^521101, p-21] \\
&\xrightarrow{O_2} [11^42020^320^21111^411^411^411^521101, p-26].
\end{aligned}$$

$$\begin{aligned}
[u,p] &= [20^5120^320^21120^510^410^410^52101, p] \\
&\xrightarrow{O_2} [11^42020^320^21120^510^410^410^52101, p-5] \\
&\xrightarrow{O_{18}} [11^42020^320^21111^4200^410^410^52101, p-10] \\
&\xrightarrow{O_{23}} [11^42020^320^21111^411^4200^410^52101, p-15] \\
&\xrightarrow{O_{28}} [11^42020^320^21111^411^411^4200^52101, p-20] \\
&\xrightarrow{O_{33}} [11^42020^320^21111^411^411^411^521101, p-26].
\end{aligned}$$

So we see that, to obtain a maximal ℓ_4, it is better to apply O_A first. This sequence results in a decreasing sequence of p by $5, 5, 5, 6$. Any other different sequence can at best yield only $5, 5, 5, 5$.

Thus, the key idea is that if we do not have a consecutive $0^5 10^5$, we should search for a configuration of the form $0^5(10^4)^k 10^5$. To obtain a decrease of 6 as soon as possible, we choose k to be minimal.

Hence, the optimal structure is:

$$T = \min_k 0^m(10^{m-1})^k 10^m = \min_k M(\tilde{M})^k 1M.$$

From the idea raised by the above example, we construct the following motif T, and search the first occurrence of T in u.

Input : A superstable sandpile configuration u
Output: Motif T
$m :=$ the greatest length of a 0-sequence in u;
$M := 0^m$; $T := M$; $b :=$ True;
while b **do**
    if *there exists more than one occurrence of* T **then**
        if *there exists* $T1T$ **then**
            $T := \max_k T(1T)^k$;
        end
        else if *there exists* $T(\tilde{T})^k 1T$ **then**
            $T := \min_l T(\tilde{T})^k 1T$;
        end
        else
            $b :=$ False;
        end
    end
end

Algorithm 1: Computing the motif T

Remark 4.1. When searching for the motif T, we consider whether applying the basic operation transforms a sequence of type 10^t into 0^{t+1} (if $b = 1$). However, it is also possible to transform a sequence of type 0^t1 into 0^{t+1} (if $a = 1$). This does not create an issue, since T is symmetric. In fact, if a potential sequence appears on the left of the current position, it means there exists a motif T in a position on the left. However, since T is defined as the first motif to appear, this scenario is not possible. For instance, suppose we have a sequence 0^4 at position i, given as $*a0^4b*$, and we are searching at i the motif $T = 0^4 10^3 10^3 10^4$. This implies that $b*$ contains $10^3 10^3 10^4*$. Now, suppose that on the left symetrically, a similar sequence is created, meaning that $*a = *0^4 10^3 10^3 1$. In this case, at position $i - 13$, we already have the motif $T = 0^4 10^3 10^3 10^4$. However, since we are searching for the first occurrence of the motif T, this assumption leads to a contradiction. Therefore, the hypothesis that the motif appears on the left is not possible.

Thus, it is sufficient to apply the algorithm to find the motif T by considering only the right side and not worrying about the sequence on the left.

With this construction of the motif, we can state our main result.

Theorem 4.3. *Let $[u, p]$ be a superstable configuration. We apply the following procedure repeatedly: (- Find the motif T; - Apply the operation O_T*

on u;). Continue this process until we obtain a maximum superstable configuration.

Let $i_1, i_2, \ldots, i_k$ be the sequence of positions where the above basic operations are applied consecutively:

$$[u,p] \xrightarrow{O_{i_1}} [u_1, p-\ell_1] \xrightarrow{O_{i_2}} [u_2, p-\ell_1-\ell_2] \ldots \xrightarrow{O_{i_k}} [u_k, p-\ell_1-\ell_2-\ldots-\ell_k].$$

Then, the rank of $[u,p]$ is given by the integer r such that:

$$\ell_1 + \ldots + \ell_r \leq p < \ell_1 + \ldots + \ell_{r+1}.$$

In other words, if $\rho([u,p]) = r$, then the configuration $\lambda_r = \epsilon^{(i_1)} + \ldots + \epsilon^{(i_{r+1})}$ is a proof.

Proof. We recall the idea behind the scheme of the algorithm. Our goal is to find λ_i such that

$$[u,p] - \lambda_i \sim_{\mathcal{L}(G)} [v, p-\ell_i]$$

where ℓ_i is maximized.

It is clear that $\lambda_1 = \epsilon^{(i)}$, where i belongs to any $T = M$, and we have $\ell_1 = m$. However, from $i \geq 2$ onward, ℓ_i can be the sum of several m or $m+1$, because a basic operation on $M1M = 0^m10^m$ can be transformed into $*0^{m+1}$.

The key idea is that the sooner we encounter the sequence 10^m, the better we optimize ℓ. Thus, whenever we have $M1M$, we take it. More generally, we select $M(1M)^k$ with the maximal possible k.

When there are no consecutive $M1M$ but rather a structure $A = M1\tilde{M}$, it is equivalent to either taking A first and then a single M, or taking M first and then A. In both cases, the value of ℓ remains $\ell = m + m$.

However, if there exists a structure $A = M1\tilde{M}1M$, it is preferable to take A before taking a single M, because after two basic operations on A, we "meet" a 0^{m+1}. Therefore, we choose the smallest k of $M(1\tilde{M})^k1M$ in order to encounter $1M$ as early as possible.

We continue this procedure iteratively to obtain the maximal possible T.

Thus, the motif T is constructed in such a way that at each step, there is no better choice to achieve a longer length ℓ.

This proves the statement. □

Remark 4.2. In our algorithm, the basic operation can be applied to a sequence of the form $*a0^mb*$, transforming it into $*(a-1)1^{m-1}2(b-1)*$, where $a, b > 0$. However, if a and b correspond to the same position in the cycle, the operation cannot be performed as usual.

We consider three special cases, which we treat separately:

- If $u = 20^{n-1}$, then:

$$[u, p] \xrightarrow{O_2} [01^{n-2}2, p - (n-1)],$$

  which is a maximal superstable configuration. Hence, we can stop the procedure.

- If $u = 10^{n-1}$, instead of applying the basic operation O_2, we proceed as follows:

$$[u, p] - \epsilon^{(2)} \xrightarrow{\Delta^{(0)}} [01^{n-2}2, p - n],$$

  which is a maximal superstable configuration, allowing us to terminate the process.

- If $u = 0^n$, instead of applying the basic operation O_1, we use:

$$[u, p] - \epsilon^{(1)} \xrightarrow{\Delta^{(0)}} [01^{m+1}, p - n],$$

  and then, we apply the algorithm for the superstable configuration $[01^{m+1}, p - n]$.

```
Input  : A configuration [u, p]
Output: The rank ρ([u, p])
u ← û; s ← 0; λ ← 0;
if u = a0^{n-1} where a ∈ {0, 1, 2} then
  | Apply the procedure as in Remark 4.2. Stop.
end
while p ≥ 0 do
  | Compute the motif T;
  | Find the first occurrence of T in u;
  | while p ≥ 0 and not finish T do
  |   | Apply the basic operation on T consecutively:
  |   |   [u, p] --O_i--> [u, p];
  |   | λ ← λ + ε^(i);
  |   | s ← s + 1;
  | end
end
Return: s − 1, λ;
```

Algorithm 2: Computing the Rank

Moreover, for a given sandpile configuration u, we can construct the tables $\Lambda = [\lambda_i]$ and $D = [d_i]$ to compute the rank of $[u, p]$ for all p by the formula: $\rho([u,p]) = r$ for $d_r \leq p < d_{r+1}$. The value λ_{r+1} serves as a proof for $[u, p]$.

Note that the table terminates when we reach a maximal superstable configuration v at some step k, which means that

$$[u,p] - \lambda_k \sim_{\mathcal{L}(G)} [v, p - d_k],$$

where $\rho([v, p - d_k]) = p - d_k$ and $\rho([u,p]) = k + p - d_k$ for $p \geq d_k$.

Furthermore, we do not need to apply the operation O_T; instead, we only analyze the 0-sequence inside T.

Input : A sandpile configuration u
Output: Proof table Λ and Rank table D
$u \leftarrow \hat{u}$; $s \leftarrow 1$; $\lambda_0 \leftarrow 0$; $d_0 \leftarrow 0$;
if $u = a0^{n-1}$ *where* $a \in \{0,1,2\}$ **then**
    Apply the procedure as in Remark 4.2; Stop.
end
while *u is not maximal superstable* **do**
    Compute the motif $T = 0^{m_1}10^{m_2}\dots10^{m_k}$ with $u = *aTb*$;
    Find the position i of the first occurrence of T in u;
    $m_1 \leftarrow m_1 - 1$;
    for $j \leftarrow 1$ **to** k **do**
        $\lambda_s \leftarrow \lambda_{s-1} + \epsilon^{(i)}$;
        $d_s \leftarrow d_{s-1} + (1 + m_j)$;
        $i \leftarrow i + 1 + m_j$;
        $s \leftarrow s + 1$;
    end
    replace aTb in u by $(a-1)1^*2(b-1)$;
end
Return: D, Λ

Algorithm 3: Computing Proof table and Rank table

Input : A sandpile configuration $[u, p]$
Output: Rank $\rho([u, p])$ and proof λ
Compute the Proof table Λ and the Rank table D;
if *$p > d_k$ for the maximal d_k* **then**
  $\rho([u, p]) \leftarrow k + p - d_k$;
  $\lambda \leftarrow [\lambda_k, p - d_k + 1]$; Stop;
end
Find the index r such that $d_r \leq p < d_{r+1}$;
$\rho([u, p]) \leftarrow r$;
$\lambda \leftarrow \lambda_{r+1}$;
return $\rho([u, p]), \lambda$;

Algorithm 4: Computing the rank from Tables

4.4 Complexity

We analyze the algorithm for computing the total rank $\rho([u, p])$ for a given u with all values of p. The most time-consuming step is the computation of the motif T. To achieve this, we first perform an $O(n)$ pass to find the integer m, which represents the maximal length of the 0-sequence 0^m. Then, additional passes are made to identify the motif T. The largest number of passes is $O(x)$, where x is the size of T, so the total time required to find T is $O(xn)$.

The operation O_T takes $O(T)$ time, and after its execution, almost all entries in T become 1. We then take another pass to find a new T in the configuration U but "forget" the newly appearing 1-sequence. At this stage, the time required is now T_{n-x}, and the process repeats. Thus, we have $T_n = O(xn) + T_{n-x}$, leading to $T_n = O(n^2)$.

Note that this complexity estimation is rather coarse; if one refines the procedure for finding T with a more precise analysis, it may be possible to achieve a better complexity.

References

[1] M. Baker and S. Norine. Riemann-roch and abel-jacobi theory on a finite graph. *Advances in Mathematics*, 215:766–788, 2007.

[2] Matthew Baker and Farbod Shokrieh. Chip-firing games, potential theory on graphs, and spanning trees. *Journal of Combinatorial Theory, Series A*, 120(1):164–182, 2013.

[3] A. Björner and L. Lovász. Chip firing games on directed graphs. *J. Algebraic Combin.*, 1:305–328, 1992.

[4] A. Björner, L. Lovász, and W. Shor. Chip-firing games on graphs. *European J. Combin.*, 12:283–291, 1991.

[5] Kristóf Bérczi, Hung P. Hoang, and Lilla Tóthmérész. On approximating the rank of graph divisors. *Discrete Mathematics*, 346(9):113528, September 2023.

[6] Yichao Chen, Jonathan L. Gross, and Toufik Mansour. On the genus distributions of wheels and of related graphs. *Discrete Mathematics*, 341(4):934–945, April 2018.

[7] R. Cori and Y. Le Borgne. On the computation of baker and norine's rank on the complete graph. *Electronic Journal of Combinatorics*, 30:P1.31, 2016.

[8] D. Dhar. Self-organized critical state of sandpile automaton models. *Phys. Rev. Lett.*, 64:1613–1616, 1990.

[9] Tomasz Dzido and Andrzej Jastrzebski. Turán numbers for odd wheels. *Discrete Mathematics*, 341(4):1150–1154, April 2018.

[10] V. Kiss and L. Tóthméresz. Chip-firing games on eulerian digraphs and np-hardness of computing the rank of a divisor on a graph. *Discrete Applied Mathematics*, 193:48–56, 2015.

[11] Thilo Krill. Universal graphs with forbidden wheel minors. *Journal of Graph Theory*, September 2024.

[12] D. Rossin. *Propriétés combinatoires de certaines familles d'automates cellulaires.* PhD thesis, PhD Thesis, 2000. Available online at `https://pastel.archives-ouvertes.fr/tel-00011297/document`.

[13] Long-Tu Yuan. Extremal graphs for odd wheels. *Journal of Graph Theory*, August 2021.