Research Repository UCD

Title	Chip-Firing on the Complete Split Graph: Motzkin Words and Tiered Parking Functions
Authors(s)	Dukes, Mark
Publication date	2021-09-10
Publication information	Dukes, Mark. “Chip-Firing on the Complete Split Graph: Motzkin Words and Tiered Parking Functions.” Springer, September 10, 2021. https://doi.org/10.1007/978-3-030-83823-2_70.
Conference details	EuroComb 2021: Online European Conference on Combinatorics, Graph Theory and Applications, Barcelona, Spain, 6-10 September 2021
Publisher	Springer
Item record/more information	http://hdl.handle.net/10197/25961
Publisher's version (DOI)	10.1007/978-3-030-83823-2_70

Downloaded 2026-05-01 23:36:28

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

© Some rights reserved. For more information

Chip-firing on the complete split graph: Motzkin words and tiered parking functions

Mark Dukes[1]

School of Mathematics and Statistics, University College Dublin, Ireland
mark.dukes@ucd.ie

Abstract. We highlight some results from studying chip-firing on the the complete split graph [5]. In this work it is shown that recurrent states can be characterised in terms of Motzkin words and can also be characterised in terms of a new type of parking function that we call a *tiered parking function*. These new parking functions arise by assigning a tier (or colour) to each of the cars, and specifying how many cars of a lower-tier one wishes to have parked before them.

Keywords: chip-firing, recurrent states, complete split graph, Motzkin words, parking functions

1 Introduction

The complete split graph is a bipartite graph consisting of two distinct parts, a clique part in which all distinct pairs of vertices are connected by a single edge, and an independent part in which no two vertices are connected to an edge. There is precisely one edge between every vertex in the clique part and every vertex in the independent part. We denote the complete split graph which has vertices $\{v_1, \ldots, v_m\}$ in the clique part and vertices $\{w_1, \ldots, w_n\}$ in the independent part by $S_{m,n}$. The graph $S_{5,4}$ is illustrated in Figure 1. The graph $S_{m,n}$ contains the complete graph K_m as a subgraph, but is also a bipartite graph in its own right, and it is this dual feature that we find interesting to examine in terms of chip-firing.

Chip-firing has been studied on several classes of graphs, and rich connections to other combinatorial structures have been established in each of the cases. Cori and Rossin [3] showed that the set of recurrent states of chip-firing on the complete graph are in one-to-one correspondence with parking functions of order n. The author in collaboration with others [6,1,8,7] showed that the set of recurrent states of chip-firing on the complete bipartite graph admits a characterization in terms of planar animals called *parallelogram polyominoes*. Cori and Poulalhon in [2] showed that the recurrent states of chip-firing on the complete tri-partite graph $K_{1,p,q}$ admits a description in terms of a parking function for cars of two different colours.

We will highlight some of the results from our paper [5] that characterizes recurrent states of chip-firing on the complete split graph. The characterization is

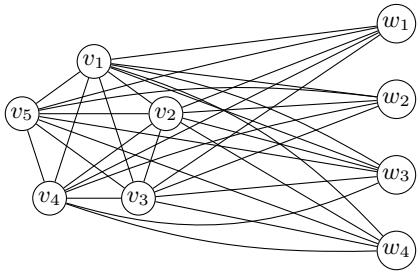


Fig. 1: The complete split graph $S_{5,4}$

in terms of Motzkin words and we give a second characterisation of the recurrent states in terms of a new type of parking function that we call a *tiered parking function*. These parking functions are characterised by assigning a tier (or colour) to each of the cars, and specifying how many cars of a lower-tier one wishes to have parked before them in a one-way street.

2 Recurrent states for chip-firing on a graph

The chip-firing game, also known as the Abelian sandpile model (ASM) in the literature, may be defined on any undirected graph G with a designated vertex s called the *sink*. A *configuration* on G is an assignment

$$c : \text{Vertices}(G)\backslash\{s\} \ \mapsto \ \mathbb{N} = \{0, 1, 2, \ldots\}.$$

The number $c(v)$ is sometimes referred to as the number of *grains* at vertex v.Given a configuration c, a vertex v is said to be *stable* if the number of grains at v is strictly smaller than the threshold of that vertex, which is the degree of v, denoted $\deg(v)$. Otherwise v is *unstable*. A configuration is called *stable* if all non-sink vertices are stable.

If a vertex is unstable then it may *topple*, which means the vertex donates one grain to each of its neighbors. The sink vertex has no height associated with it and it only absorbs grains, thereby modelling grains exiting the system. Given this, it is possible to show that starting from any configuration c and toppling unstable vertices, one eventually reaches a stable configuration c'. Moreover, c' does not depend on the order in which vertices are toppled in this sequence. We call c' the *stabilisation* of c. We use the notation $\mathsf{ASM}(G, s)$ to indicate that we are considering the Abelian sandpile model as described here on the graph G with sink s.

Starting from the empty configuration, one may indefinitely add any number of grains to any vertices in G and topple vertices should they become unstable. Certain stable configurations will appear again and again, that is, they *recur*, while other stable configurations will never appear again. These *recurrent configurations* are the ones that appear in the long term limit of the system. In [4, Sec.

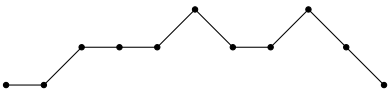
6], Dhar describes the so-called *burning algorithm*, which establishes whether a given stable configuration is recurrent:

Proposition 1 ([4], Section 6.1). *Let G be a graph with sink s. A stable configuration c on G is recurrent if and only if there exists an ordering $v_0 = s, v_1, \ldots, v_n$ of the vertices of G such that, starting from c, for any $i \geq 1$, toppling the vertices $v_0, \ldots, v_{i-1}$ causes the vertex v_i to become unstable.*

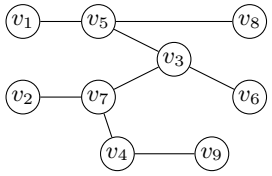
A configuration on $\mathsf{ASM}(S_{m,n}, s)$ is a vector $c = (c_1, \ldots, c_{i-1}, -, c_{i+1}, \ldots, c_{m+n-1})$ whereby c_i represents the number of chips/grains at the ith vertex in the sequence $(v_1, \ldots, v_m, w_1, \ldots, w_n)$. The dash indicates the location corresponding to the sink. We will use a semi-colon in the configurations to distinguish between the clique and independent parts, e.g. $c = (a_1, \ldots, a_{m-1}, -; b_1, \ldots, b_n)$. Let $\mathsf{Rec}(\mathsf{ASM}(S_{m,n}, s))$ be the set of recurrent states of the model. We will restrict our analysis to those configurations that are weakly decreasing with respect to vertex labels. This restriction will allow us to focus our analysis to consider characterizing all those 'different' configurations, and from which we can generate all configurations through permutations.

3 Motzkin words and recurrent states for a clique-sink

We will use Motzkin words for the characterisation of recurrent states. A *Motzkin path P of length p* is a lattice path in the plane from $(0, 0)$ to $(p, 0)$ which never passes below the x-axis and whose permitted steps are the up step $U = (1, 1)$, the down step $D = (1, -1)$, and the horizontal step $H = (1, 0)$. An example of a Motzkin path is given in Figure 2a. The Motzkin word of a path is a listing of the p steps of the path in the order they appear from left to right, e.g. the Motzkin word of Figure 2a is $\alpha = HUHHUDHUDD$.



(a) Example of a Motzkin path of length 10



(b) A spanning tree of the complete split graph $S_{5,4}$

Fig. 2

Definition 1. *A* Motzkin word *is a word α consisting of the letters U, D, H with the properties (i) in counting α from left to right the U count is always greater than or equal to the D count, and (ii) the total U count is equal to the D count. Let $\mathsf{Motzkin}_{m,n}$ be the set of Motzkin words with m occurrences of U and n occurrences of H.*

The deletion of the H letters in a Motzkin word gives a Dyck word and, conversely, a Motzkin word with m occurrences of U and n occurrences of H is obtained by taking a Dyck word with m occurrences of U and inserting n occurrences of H in any positions in this word. This decomposition provides an enumeration of such words: $|\mathsf{Motzkin}_{m,n}| = \frac{1}{m+1}\binom{2m}{m}\binom{2m+n}{n}$.

Definition 2. *To any Motzkin word $\alpha \in \mathsf{Motzkin}_{m-1,n}$ we associate a configuration $f(\alpha) = (a_1, \ldots, a_{m-1}, -; b_1, b_2, \ldots, b_n)$ on $S_{m,n}$ as follows:*

- *a_i is such that $a_i + 1$ is equal to the number of occurrences of the letter D plus the number of occurrences of the letter H appearing in α after the i-th occurrence of the letter U.*
- *b_i is equal to the number of occurrences of the letter D appearing after the i-th occurrence of the letter H in α.*

Observe that, by construction, the sequences $(a_1, \ldots, a_{m-1})$ and $(b_1, \ldots, b_n)$ are weakly decreasing. For the Motzkin word example above, the associated configuration on $\mathsf{ASM}(S_{4,4}, v_4)$ is $f(\alpha) = (5, 3, 1, -; 3, 3, 3, 2)$.

Theorem 1. *A weakly decreasing stable configuration $c = (a_1, \ldots, a_{m-1}, -; b_1, \ldots, b_n)$ on $\mathsf{ASM}(S_{m,n}, v_m)$ is recurrent iff it corresponds, via the construction f, to a unique Motzkin word α in $\mathsf{Motzkin}_{m-1,n}$.*

Since the number of such decreasing recurrent configurations is equal to the number $|\mathsf{Motzkin}_{m-1,n}|$, we have the number of weakly decreasing recurrent configurations on $S_{m,n}$ with sink v_m is $\frac{1}{m}\binom{2m-2}{m-1}\binom{2m-2+n}{n}$.

3.1 Prüfer code decomposition for spanning trees of $S_{m,n}$

It is a well established fact the the number of recurrent states of the Abelian sandpile model on a graph is equal to the number of spanning trees of that graph. We will enumerate the set of recurrent states by enumerating the spanning trees of the complete split graph. We will do this by presenting a bijective proof that uses the Prüfer code of the spanning trees.

To do this we use a different labelling of the vertices of the complete split graph. Let $S_{m,n}$ have clique vertices $\{v_1, v_2, \ldots, v_m\}$ and independent vertices $\{v_{m+1}, \ldots, v_{m+n}\}$. Suppose these vertex labels are totally ordered:

$$v_1 < v_2 < \cdots < v_m < v_{m+1} < \cdots < v_{m+n}.$$

Any spanning tree T of $S_{m,n}$ is a tree with vertices labelled $v_1, \ldots, v_{m+n}$ such that for any edge (v_i, v_j) of T, one has $i \leq m$ or $j \leq m$.

The Prüfer code of this tree is obtained by successively deleting the leaves of T with minimal label and recording the vertex to which they were attached, until T has only one edge. The unique edge that is obtained at the end of the procedure is (v_{m+n}, v_i) where $i \leq m$. Two words f, g may be built using the alphabet consisting of the labels of the vertices. Initially both f and g are empty, and at each step of the procedure a letter is added either to f or to g. It is added

to f if the leaf deleted is a clique vertex and added to g if it is an independent vertex. The letter added is the label of the vertex neighbour of the deleted leaf.

For the tree given in Figure 2b, the vertices are deleted in the following order

$$v_1, v_2, v_6, v_8, v_5, v_3, v_7.$$

The Prüfer code of the tree is $v_5, v_7, v_3, v_5, v_3, v_7, v_4$ and the construction of the two words f and g gives

$$\mathsf{f} = v_5, v_7, v_3, v_7 \qquad\qquad \mathsf{g} = v_3, v_5, v_4.$$

Proposition 2. *There is a bijection between spanning trees of $S_{m,n}$ and pairs of words (f, g) such that f has length $m-1$ and has letters in the alphabet $\{v_1, \ldots, v_{m+n}\}$, while g has $n-1$ letters in the alphabet $\{v_1, \ldots, v_m\}$.*

Corollary 1. *The number of spanning trees of $S_{m,n}$ is $(m+n)^{m-1}m^{n-1}$ and, consequently, this is the number of recurrent configurations of the Abelian sandpile model on the complete split graph $S_{m,n}$.*

4 Tiered parking functions

We may also offer the following definition of recurrent configurations as a new type of parking function that we will call a *tiered parking function*. Parking functions were mentioned earlier in the paper in relation to the recurrent states of the sandpile model on the complete graph. Moreover, the G-parking functions of Postnikov and Shapiro [9] provide a useful language in which an alternative description of recurrent states of the sandpile model on a general graph may be given.

The application of G-parking functions to the complete split graph is different to what we present in this section. Our aim is to provide a new 'type' of parking function, and provide a setting in which recurrence is quite easily established in this new context. We refer the interested reader to Yan [11] for a discussion of G-parking functions and their relation to the ASM. Our definition is inspired by Cori and Poulalhon's [2] concept of (p,q)-parking functions.

Definition 3 (k-tiered parking function). *Let $m_1, \ldots, m_k$ be a sequence of positive integers with $m_1 + \ldots + m_k = M$. Suppose that there are m_i cars of colour/tier i and there are M parking spaces. We will call a sequence $P = (m_1; P_2, \ldots, P_k)$ of sequences $P_i = (p_1^{(i)}, \ldots, p_{m_i}^{(i)})$ a k-tiered parking function of order $(m_1, \ldots, m_k)$ if there exists a parking configuration of the M cars that satisfies the following preferences for all drivers:*

> *the driver of the jth car having colour $i > 1$ asks that there be at least $p_j^{(i)}$ cars of colours $\{1, \ldots, i-1\}$ parked before him.*

Drivers of cars having colour 1 have no preferences with regard to other coloured cars, which is why we only list their number m_1 in P.

Example 1. The sequence $P = (4; (2,1,0,4,2), (8,2,1,2), (4,10,8))$ is a 4-tiered parking function of order $(4,5,4,3)$. This is realised by the following parking configuration where the leftmost entry represents the first parking spot, and the colour of the parked car is indicated.

$\longrightarrow$ direction of traffic $\longrightarrow$

2	3	1	3	4	3	2	1	2	4	2	1	4	1	3	2

Example 2. The sequence $P = (9; (8,2,9,4,6,9,7,8,2,7,9,4), (18,2,14,6,21,13,7,13,3))$ is a 3-tiered parking function of order $(9,12,9)$. This is realised by the following parking configuration:

$\longrightarrow$ direction of traffic $\longrightarrow$

1	2	3	2	3	2	1	2	3	2	3	2	1	1	2	1	2	3	3	2	3	1	2	2	1	3	2	2	2	3

We now connect these tired parking functions to the recurrent states of the sandpile model for the complete split graph.

Theorem 2. *A configuration* $c = (a_1, \ldots, a_{m-1}, -; b_1, \ldots, b_n)$ *on* $\mathsf{ASM}(S_{m,n}, v_m)$ *is recurrent iff there exists a 3-tiered parking function of order* $(m-1, n, m-1)$ *with* $P = ((b_1, \ldots, b_n), (a_1+1, \ldots, a_{m-1}+1))$.

References

1. J.-C. Aval, M. D'Adderio, M. Dukes & Y.Le Borgne. Two operators on sandpile configurations, the sandpile model on the complete bipartite graph, and a Cyclic Lemma. *Adv. App. Math.* **73**:59–98, 2016.
2. R. Cori & D. Poulalhon. Enumeration of (*p*,*q*)-parking functions. *Disc. Math.* **256**:609–623, 2002.
3. R. Cori & D. Rossin. On the Sandpile Group of Dual Graphs. *Eur. J. Comb.* **21** (2000), 447–459.
4. D. Dhar. Theoretical studies of self-organized criticality. *Physica A: Stat. Mech. App.*, **369**(1):29–70, 2006.
5. M. Dukes. The sandpile model on the complete split graph, Motzkin paths, and tiered parking functions. *J. Comb. Theory Ser. A* **180** (2021). doi: 10.1016/j.jcta.2021.105418
6. M. Dukes & Y. Le Borgne. Parallelogram polyominoes, the sandpile model on a complete bipartite graph, and a q,t-Narayana polynomial. *J. Comb. Theory Ser. A* **120**(4):816–842, 2013.
7. M. Dukes, T. Selig, J.P. Smith & E. Steingrímsson. The Abelian sandpile model on Ferrers graphs – A classification of recurrent configurations. *Eur. J. Comb.* **81**:221–241, 2019.
8. M. Dukes, T. Selig, J.P. Smith & E. Steingrímsson. Permutation graphs and the Abelian sandpile model, tiered trees and non-ambiguous binary trees. *Elec. J. Comb.* **26**(3):P3.29, 2019.
9. A. Postnikov & B. Shapiro. Trees, parking functions, syzygies, and deformations of monomial ideals. *Trans. AMS* **356**(8):3109–3142, 2004.
10. T. Selig, J.P. Smith & E. Steingrímsson. EW-tableaux, Le-tableaux, tree-like tableaux and the Abelian sandpile model. *Elec. J. Comb.* **25**(3):P3.14, 2018.
11. C.H. Yan. Parking functions. In *Handbook of Enumerative Combinatorics*, Discrete Math. Appl. (Boca Raton), pages 835–893. CRC Press, Boca Raton, FL, 2015.