Séminaire Lotharingien de Combinatoire **XX** (2026)
Article #105, 12 pp.

Proceedings of the 38th Conference on Formal Power Series and Algebraic Combinatorics (Seattle)

Parking functions and chip-firing on hypergraphs

Timothy Blanton[1], Anton Dochtermann[2], Isabelle Hong[3], Suho Oh[2], and Zhan Zhan[4]

[1]*Department of Mathematics, University of California, Davis*
[2]*Department of Mathematics, Texas State University*
[3]*Department of Mathematics, University of California, Los Angeles*
[4]*Department of Mathematics, University of Washington*

Abstract. For a connected graph G with sink vertex q, a G-parking function is a vector of nonnegative integers whose entries are determined by cut-sets in G. Such objects also arise as the superstable configurations in the context of chip-firing. The set of all G-parking functions has various algebraic and combinatorial properties; for instance they relate to evaluations of the Tutte polynomial and in particular are counted by spanning trees of G. We extend these constructions to the setting of hypergraphs, where edges can have multiple vertices. For a hypergraph H with sink q, we define H-parking functions in terms of cuts in H and prove that the maximal such sequences are characterized by certain acyclic orientations of H. We introduce a notion of a q-rooted spanning tree for H, and prove that the set of all such objects are counted by H-parking functions. We also show how H-parking functions can be recovered as the superstable configurations in a version of chip-firing on H, where chips have a choice of where to go when fired. We prove that one can recover such configurations via chip-firing on a family of digraphs associated to H.

Keywords: hypergraphs, chip-firing, parking functions, acyclic orientations

1 Introduction

Suppose $G = (V, E)$ is a connected graph with specified sink vertex $q \in V$ and nonsink vertices $\{1, 2, \ldots, n\}$. A *G-parking function* is a sequence $\vec{c} \in \mathbb{Z}^n_{\geq 0}$ of nonnegative integers whose entries are bounded above by parameters associated to vertex cuts in G. Such objects were studied by Postnikov and Shapiro in [9], and when $G = K_{n+1}$, the complete graph, they recover the classical parking functions that originated in queueing theory and have been the subject of extensive study [12]. In this case, such objects have an easy description: a sequence $\vec{c} \in \mathbb{Z}^n_{\geq 0}$ is a parking function if and only if its nondecreasing rearrangement is coordinatewise bounded by the vector $\vec{v} = (0, 1, \ldots, n-1)$. This naturally leads to a more general notion of a *vector parking function*.

G-parking functions can also be understood in terms of *chip-firing* on the graph G, where the distribution rule is determined by the reduced Laplacian $L_G = L^q_G$. Here we consider configurations $\vec{c} \in \mathbb{Z}^n_{\geq 0}$ of chips on the nonsink vertices of G. If a nonsink

vertex has at least as many chips as its degree, it can fire, passing one chip to each of its neighbors. If no non-sink vertices can be fired, we say that the configuration is *stable*. A configuration $\vec{c}$ is said to be *superstable* if no nonempty set of non-sink vertices can fire, so that subtracting from $\vec{c}$ any nonempty collection of the columns of L_G results in a vector with a least one negative coordinate. For undirected graphs, the set of superstable configurations can be seen to coincide with the collection of G-parking functions.

According to the definitions, to determine whether a given $\vec{c} \in \mathbb{Z}^n_{\geq 0}$ is a G-parking function one must consider all nonempty subsets $T \subset V \setminus q$. However, Dhar's burning algorithm [5] provides a polynomial-time way to check this condition. G-parking functions also have many pleasing combinatorial properties, for instance the collection of all such sequences is in bijection with the set of spanning trees of G. In [4], Chebikin and Pylyavskyy describe a family of such bijections between G-parking functions and spanning trees. They work in the more general setting of digraphs, where the relevant combinatorial objects are q-rooted spanning trees. For the case of undirected graphs, a particular specialization of their construction can be seen to recover Dhar's algorithm.

Varying the choice of sink vertex q will in general lead to a different set of G-parking functions for a given graph G. However, some parameters are invariant. As mentioned above, if G is connected then the set of G-parking functions is in bijection with the set of spanning trees of G. Furthermore, for a graph G with a specified sink q, the *maximal* G-parking functions are in bijection with acyclic orientations of G with a unique source q [2]. From this it follows that all maximal G-parking functions have the same degree. In [7], Merino showed that the *degree sequence* of G-parking functions can be recovered as an evaluation of the Tutte polynomial of G (and hence is independent of the choice of q). This then can be used [8] to establish a conjecture of Stanley regarding the h-vector of matroids for the special case of cographic matroids.

2 Our setting: hypergraphs and H-parking functions

In this work we extend the constructions and results described above to the setting of *hypergraphs*, where edges can now have more than one 'endpoint'. More formally, a *hypergraph* $H = (V, E)$ consists of a vertex set $V = V(H)$ and a set $E = E(H)$ of subsets of V called hyperedges (or just edges if the context is clear). We can represent a hypergraph H in terms of its bipartite incidence graph $B(H) = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V} = E \sqcup V$ and with edges $(e, v) \in \mathcal{E}$ whenever $v \in e$. We refer to Figure 1 for a running example that we will use throughout this paper. If all elements of E have the same cardinality d, we say that H is *d-regular*. The *complete hypergraph* K^d_n is the d-regular hypergraph on vertex set $[n] = \{1, 2, \ldots, n\}$ whose edge set consists of all d-subsets of $[n]$.

The *degree* of a vertex $v \in V(H)$ is given by

$$\deg(v) = |\{e \in E(H) : v \in e\}|.$$

Note that $\deg(v)$ agrees with the degree of v in the (usual) graph $B(H)$.

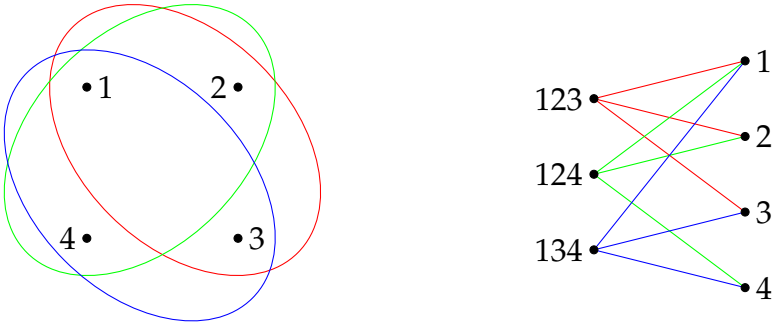

Figure 1: A hypergraph H along with its bipartite representation $B(H)$.

Remark 2.1. *Throughout the paper, we will assume that our hypergraphs H are* connected*, i.e., given any two vertices v_1 and v_2, there exists a sequence of edges $(e_1, e_2, \ldots, e_m)$ such that $v_1 \in e_1$, $v_2 \in e_m$ and $e_i \cap e_{i+1} \neq \emptyset$ for all $i = 1, \ldots m-1$. From this it follows that the graph $B(H)$ is also connected.*

We will also need a notion of degree of a vertex relative to a subset that contains it.

Definition 2.2. *Suppose $H = (V, E)$ is a hypergraph, and let $T \subset V$ be a nonempty set of vertices with $v \in T$. The* T-degree of v *is*

$$\deg_T^H(v) = |\{e \in E : v \in e \text{ and } e \not\subset T\}|.$$

If the underlying hypergraph is clear, we will write $\deg_T(v)$. Also note that if $T = \{v\}$ then we have $\deg_{\{v\}}(v) = \deg(v)$. With this we can provide one of our main definitions.

Definition 2.3. *Suppose $H = (V, E)$ is a hypergraph with sink vertex $q \in V$ and nonsink vertices $[n]$. An* H-parking function *is a sequence $\vec{c} \in \mathbb{Z}_{\geq 0}^n$ such that for all $T \subset [n]$, there exists an $i \in T$ such that $c_i < \deg_T(i)$.*

With the connection to chip-firing in mind, we will often refer to a vector $\vec{c} \in \mathbb{Z}_{\geq 0}^n$ as a *configuration of chips*. If a nonempty subset $T \subset [n]$ has the property that $\deg_T(i) \leq c_i$ for all $i \in T$, we will say that T is *bounded* by $\vec{c}$. Hence a configuration $\vec{c}$ is an H-parking function if no nonempty subset of $[n]$ is bounded by $\vec{c}$.

It turns out that H-parking functions on a hypergraph H can be understood in terms of its underlying bipartite representation $B(H)$. Recall that the vertex set of $B(H)$ is given by $\mathcal{V} = E \sqcup V$. If $q \in V$ is the chosen sink vertex for H, then by convention we take q to be the sink vertex for $B(H)$. Also, if $\vec{c}$ is a configuration on the nonsink vertices of H, we let $\vec{c}_B$ denote the configuration on the nonsink vertices of $B(H)$ given by placing zeros on each vertex $w \in E$. With this we have the following observation.

Proposition 2.4. *Suppose $\vec{c}$ is a configuration on the nonsink vertices of a hypergraph H. Then $\vec{c}$ is an H-parking function for H if and only if $\vec{c}_B$ is a G-parking function for $B(H)$.*

Remark 2.5. *If H is any hypergraph, note that from Proposition 2.4 we can use Dhar's burning algorithm on $B(H)$ to check whether a given configuration on H is an H-parking function.*

2.1 Orientations and maximal parking functions

For a usual graph G with sink q, there is a simple bijection between the set of maximal G-parking functions and the set of *acyclic orientations* of G with a unique source q, as described in [2]. Here we extend this to the setting of hypergraphs. We begin with a definition.

Definition 2.6. *Suppose H is a hypergraph with sink q, and with bipartite incidence graph $B(H)$. An* orientation *of an edge $e \in E(H)$ is a choice of vertex $v_e \in e$. An* orientation of H *is a choice of orientation for each edge, which we denote $\mathcal{O} = \{v_e\}_{e \in E(H)}$.*

An orientation $\mathcal{O}$ of H defines an orientation of the bipartite graph $B(H) = E(H) \sqcup V(H)$ by prescribing the directed edge (v, e) if $v = v_e$ is the orientation of e, and (e, v) otherwise.

Definition 2.7. *An orientation $\mathcal{O}$ of a hypergraph H is* acyclic with unique source q *if the underlying orientation of $B(H)$ has this property.*

Our definition of a hypergraphic orientation is a special case of a more general notion introduced in [1]. An orientation $\mathcal{O}$ of a hypergraph H gives rise to a configuration $\vec{c} = \vec{c}(\mathcal{O})$ on the nonsink vertices $[n] = \{1, 2, \ldots, n\}$ via the induced orientation on $B(H)$: For each $i \in [n]$ we let $\vec{c}_i = \text{indeg}(i) - 1$. See Figure 2 for an example.

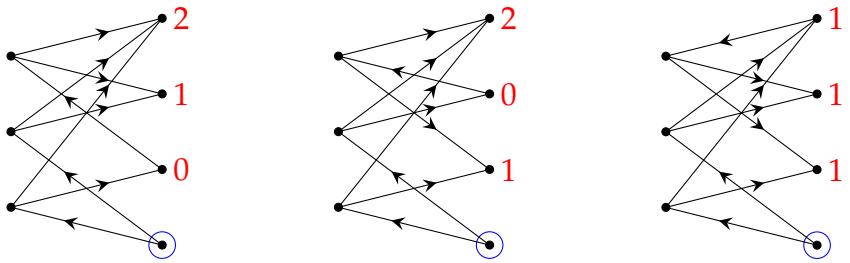

Figure 2: Acyclic orientations with corresponding maximal superstable configurations.

Theorem 2.8. *Suppose H is a hypergraph with sink q. Then the assignment described above defines a bijection between the set of acyclic orientations of H with unique source q and the set of maximal H-parking functions on H.*

From the above result it follows that the degree of a maximal H-parking function equals the number of edges oriented rightwards minus the number of non-sink vertices of H, so it is independent of the acyclic orientation we choose.

Corollary 2.9. *For a hypergraph H with chosen sink q, all maximal H-parking functions have the same degree.*

From Corollary 2.9 we have that the degree sequence of superstable configurations for any hypergraph H is a *pure O-sequence*.

2.2 Complete hypergraphs and vector parking functions

We next consider H-parking functions for the case of $H = K_n^d$, the *complete d-hypergraph on n vertices* consisting of all d-subsets of $[n]$. We show that the set of H-parking functions for such hypergraphs can be described as vector parking functions as studied by Yan [12].

To recall this notion, for a sequence $(x_1, x_2, \ldots, x_n)$ of real numbers, we let $x(1) \leq x(2) \leq \cdots \leq x(n)$ denote its rearrangement into a nondecreasing order. Now we fix a nondecreasing vector of nonnegative integers $\vec{u} = (u_1, u_2, \ldots, u_n)$. A vector $\vec{c} \in \mathbb{Z}^n_{\geq 0}$ is a $\vec{u}$-parking function if its rearrangement satisfies $0 \leq c(i) < u_i$.

Theorem 2.10. *The H-parking functions of K^d_{n+1} coincide with the set of $\vec{u}$-parking functions, where $\vec{u} = (u_1, \ldots, u_n)$ with:*

- $u_k = \binom{n}{d-1} - \binom{n-k}{d-1}$, *for $k = 1, 2, \ldots, n+1-d$;*
- $u_k = \binom{n}{d-1}$, *for $k = n-d+2, n-d+3, \ldots n$.*

A natural question to ask is if we have a way to determine the number of H-parking functions of a complete hypergraph. From [12], the number of $\vec{u}$-parking functions for any $\vec{u}$ can be determined by computing the determinant of a certain *Steck matrix*. We recall this result next.

Theorem 2.11. *[10] The number PF($\vec{u}$) of $\vec{u}$-parking functions equals $n!\det D$, where D is the $n \times n$ matrix with ij entry*

$$\frac{u_i^{j-i+1}}{(j-i+1)!}$$

if $j - i + 1 \geq 0$ (and 0 otherwise).

The ideas behind this formula go back to work of Steck [11].

Example 2.12. *For $n = 4$ and $d = 3$, we have from Theorem 2.10 that $\vec{u} = (3, 5, 6, 6)$, and hence the H-parking functions for K_5^3 are given by*

$$\{(2,4,5,5), (2,4,5,4), \ldots, (2,4,5,0), (2,4,4,5), (2,4,4,4), \ldots, (0,0,0,0)\}.$$

The corresponding Steck matrix is given by

$$D = \begin{pmatrix} 3 & \frac{9}{2} & \frac{27}{6} & \frac{81}{24} \\ 1 & 5 & \frac{25}{2} & \frac{125}{6} \\ 0 & 1 & 6 & \frac{36}{2} \\ 0 & 0 & 1 & 6 \end{pmatrix}.$$

From this we conclude that the number of H-parking functions of K_5^3 is $4! \det D = 1203$.

This determinantal formula computes the number of H-parking functions for K_{n+1}^d, but it is not always easy to implement in practice. In [12] Yan has obtained closed formulas for $\mathrm{PF}(\vec{u})$ for certain values of $\vec{u}$, but unfortunately they do not apply here. Hence we ask the following.

Question 2.13. *Is there a closed formula for the number of H-parking functions, where $H = K_n^d$ is the complete hypergraph?*

3 Tree like objects and bijections

We next introduce our notion of spanning trees for hypergraphs, and describe a bijection between these objects and the set of H-parking functions. For this, we adapt an algorithm from [4] to the setting of hypergraphs, by applying similar ideas to the underlying bipartite incidence graphs.

For our notion of hypergraph spanning trees, we will work with a certain equivalence relation on (usual) spanning trees of bipartite graphs.

Definition 3.1. *Suppose B is a connected bipartite graph with vertex set $V(B) = L \sqcup R$, and with distinguished sink vertex $q \in R$. Spanning trees T and T' of B are said to be* burning equivalent *if, when orienting the edges of T and T' towards the sink, the set of edges directed from R to L is the same.*

See Figure 3 for an example of burning equivalent trees. With this we can define our notion of a spanning tree of a hypergraph (with choice of sink vertex q).

Definition 3.2. *Suppose H is a hypergraph with sink vertex q, and let $B(H) = (\mathcal{V}, \mathcal{E})$ denote its bipartite incidence graph. A* (q-rooted) spanning tree *of H is a burning equivalence class $[T]$ of spanning trees of $B(H)$.*

Remark 3.3. *Note that the data of a spanning tree $[T]$ of H is given by a subset $T_{\leftarrow} \subset \mathcal{E}$, namely the set of edges oriented R to L when orienting edges toward the sink q. This set $T_{\leftarrow}$ has the property that every non-sink vertex $V(H) \setminus q$ is incident to some element of $T_{\leftarrow}$. Also note that for any $T \in [T]$, the degrees of vertices on the left (from $E(H)$) will always be the same.*

Remark 3.4. *One can check that Definition 3.2 generalizes the usual notion of a spanning tree of a graph, in the sense that if G is a graph, then the q-spanning trees of $B(G)$ recover the usual spanning trees of G.*

3.1 Bijections

Suppose B is a connected bipartite graph with vertex set $\mathcal{V} = L \sqcup R$ and specified sink vertex $q \in R$. Given the data of a G-parking function on B, we will be interested in a burning type algorithm that produces a spanning tree of B that can be thought of as a canonical representative of the underlying burning equivalence class.

In [4, Theorem 2.1] the authors describe a bijection between G-parking functions and rooted spanning trees of a digraph G, based on a breadth-first search order. Our construction is a generalization of the inverse map described in [4, Theorem 2.1], which we call the *inverse BCP algorithm*. In the case of a bipartite graph, the inverse BCP map that produces a spanning tree from a G-parking function $(b_1, \dots, b_n)$ can be understood in fewer steps. Instead of constructing the tree one vertex at a time, we can proceed in batches, moving from one side of the bipartite graph to the other.

In particular, at step m of the process, instead of choosing a single vertex p_m and edge e_{p_m} to add to our set, we choose all vertices in $v \in V_m$ that have the same distance from q as p_m. We then add all such vertices and edges e_v to increment. We call this the *Bipartite Breadth-first Burning (BBB) algorithm*. Details can be found in [3]. We then have the following observation.

Lemma 3.5. *Suppose $\vec{c} \neq \vec{c}'$ are H-parking functions for H, and let T and T' be the corresponding spanning trees of $B(H)$ obtained by running the BBB algorithm on $\vec{c}_B$ and $\vec{c}'_B$. Then T and T' are not burning equivalent (i.e. $T_{\leftarrow} \neq T'_{\leftarrow}$).*

This then leads us to our desired bijection.

Theorem 3.6. *For a hypergraph H with root vertex q, there exists a bijection between the set of H-parking functions and the set of q-rooted spanning trees of H.*

In work of Kálmán and Postnikov [6], the notion of a *hypertree* associated to a hypergraph is introduced. To recall this notion, suppose H is a hypergraph with bipartite representation $B(H)$. A *hypertree* in H is a function (vector) $f : E \to \mathbb{N} = \{0, 1, \dots\}$ with the property that there exists a spanning tree of $B(H)$ that has degree $f(e) + 1$ at each $e \in E$. In [6] the authors develop a notion of *internal activity* for these hypertrees that gives rise to a notion of a Tutte polynomial for polymatroids.

Proposition 3.7. *For any hypergraph H with sink q, two burning equivalent spanning trees give rise to the same hypertree.*

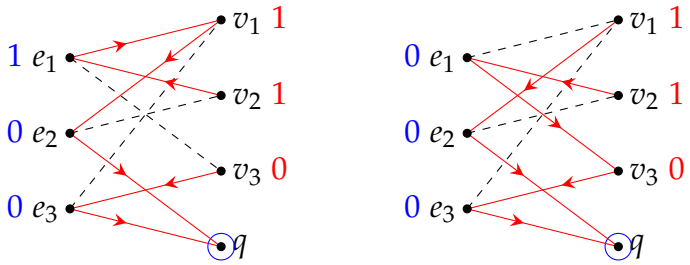

Figure 3: Burning equivalent spanning trees T and T' of $B(H)$, along with the corresponding G-parking functions of $B(H)$ given by our bijection.

For example, the spanning trees depicted in Figure 3 each correspond to the hypertree $(1,1,1)$. An important application of the bijection between spanning trees and superstable configurations is preservation of the external activity/degree. It would be interesting if we can extend the definition of external activity for spanning trees on hypergraphs and superstable configurations to our model so that it is preserved under the bijection described above.

4 Chip-firing on hypergraphs

We next investigate how H-parking functions on a hypergraph $H = (V, E)$ can be understood in terms of a notion of chip-firing. For this we fix a sink vertex $q \in V$, let $[n]$ denote the set of non-sink vertices, and consider a configuration $\vec{c} \in \mathbb{Z}^n_{\geq 0}$ of chips. Recall that for a usual graph, when we fire a vertex $v \in [n]$, one chip is sent to each incident edge and then passed to the other vertex incident to that edge. For hypergraphs, we do something similar: When we fire a vertex v, it sends a chip to each of its incident hyperedges. But now within each of these hyperedges, we must make a *choice* of which vertex receives the chip that comes in. We make this precise with the following definition.

Definition 4.1. *Suppose H is a hypergraph and $v \in [n]$ is a vertex with incident edges $e_1, \ldots, e_k$. A* firing choice *at v consists of a choice of a vertex $v_i \in e_i$ where $v_i \neq v$, for each $i = 1, \ldots, k$. We use $C_v = (e_1 : v_1, \ldots, e_k : v_k)$ to denote a firing choice.*

Given a configuration $\vec{c}$, when we fire v, we think of sending a chip to each e_i, and then using the firing choice C_v to pass that chip to v_i. Note that if H is a (usual) graph, there is a canonical firing choice for each v (namely, the other endpoint of each edge incident to v).

Example 4.2. *Let $\vec{c} = (1, 2, 0)$ be the configuration on the hypergraph H depicted in Figure 4. We consider two different firing choices for the vertex v_2. If we fire v_2 with the firing choice*

$(e_1 : v_3, e_2 : v_4)$ *we end up with the configuration* $(1,0,1)$*, whereas the choice* $(e_1 : v_1, e_2 : v_1)$ *gives us* $(3,0,0)$*.*

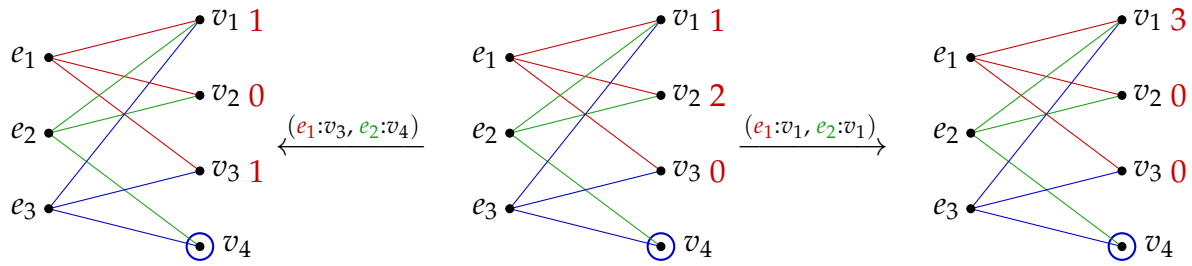

Figure 4: Example of different firing choices.

We use $C_{v,H}$ to denote the set of all possible firing choices of the vertex v in H. We write C_v if the underlying hypergraph H is clear. By considering all possible firing choices, we can recover the notion of when a vertex is ready to fire.

As was the case for usual graphs, we will also want to fire a set $T \subset V \setminus q$ of nonsink vertices. For this, we specify a firing choice for each vertex $v \in T$, but now we need an additional condition. Let $T \subset V \setminus q$ and suppose that $C_T = \{C_{v_i} : v_i \in T\}$ is a collection of firing choices for each element of T. Now suppose that there exists a hyperedge e satisfying $e \subset T$. We say that C_T is *cancellative at* e if the firing choice C_T restricted to e results in a fixed point free permutation of the elements of e. Equivalently,

$$e = \{v_j \mid C_{v_i} = (\cdots, e : v_j, \cdots)\}_{v_i \in e}.$$

This generalizes set-firing in the classical graph case, where if two adjacent vertices are fired, a chip gets passed back and forth along that edge. Hence, the default firing choice for any chosen set to be fired is cancellative at every contained edge.

Definition 4.3. *Suppose* H *is a hypergraph, and let* $T \subset V(H) \setminus q$*. A firing choice* C_T *is* cancellative *(for* T*) if it is cancellative for all* $e \subset T$*.*

Definition 4.4. *Suppose* H *is a hypergraph and let* $\vec{c}$ *be a configuration on the non-sink vertices* $V \setminus q$*. A nonempty subset* $T \subset V \setminus q$ *is* ready to fire *if every cancellative firing choice for* T *results in a nonnegative configuration.*

With this we can define our notion of superstability.

Definition 4.5. *For a hypergraph* H*, a configuration* $\vec{c}$ *on the nonsink vertices is* superstable *if no nonempty subset* $T \subset V \setminus q$ *is ready to fire.*

Although varying the cancellative firing choice of a set T could lead to different distributions of chips, detecting whether T is ready to fire can be done using degree.

Lemma 4.6. *Suppose $\vec{c}$ is a configuration on H and $T \subset V \setminus q$. Then T is ready to fire if and only if $\deg_T(i) \leq c_i$ for all $i \in T$.*

Corollary 4.7. *Suppose H is a hypergraph with sink vertex q. Then a configuration $\vec{c}$ is superstable if and only if $\vec{c}$ is an H-parking function.*

Proof. Recall that $\vec{c}$ is superstable if and only if no nonempty subset of the nonsink vertices is ready to fire. The result then follows from Lemma 4.6 and the definition of an H-parking function. □

4.1 Using digraphs to fire hypergraphs

As we have seen, firing a vertex v in a hypergraph H requires specifying a firing choice for each edge incident to v. One way to make this choice (for all vertices at once) is to cyclically order the elements of each hyperedge, giving rise to an underlying digraph on the vertices of H. In this section, we will show how chip-firing on this (collection of) digraphs can be used to recover superstable configurations on H itself.

Definition 4.8. *Let H be a hypergraph. A* cycling *$\mathcal{C}$ of H consists of a choice of cyclic orderings on the elements in each edge $e \in E(H)$.*

A cycling $\mathcal{C}$ of H defines a digraph $D_{\mathcal{C}}(H)$ in the following way. The vertex set of $D_{\mathcal{C}}(H)$ is taken to be $V(H)$, and we include the directed edge (v_i, v_j) if $v_i \prec_{\mathcal{C}} v_j$ in some hyperedge e of H. Here $x \prec_{\mathcal{C}} y$ indicates that x immediately precedes y in the prescribed cyclic order on e. See Figure 5 for an example.

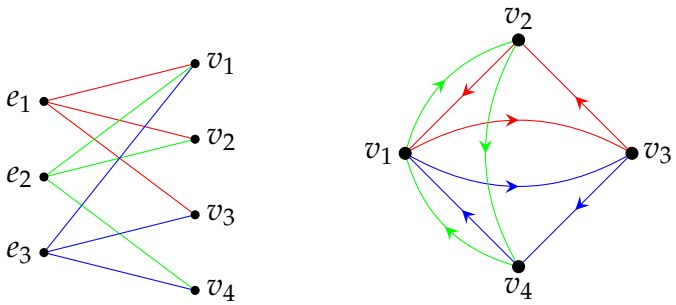

Figure 5: A hypergraph H and the digraph $D_{\mathcal{C}}(H)$ associated to the cycling $\mathcal{C} = \{(v_1v_3v_2), (v_1v_2v_4), (v_1v_3v_4)\}$.

The digraphs we obtain from cyclings on H lead to good notions of chip-firing, according to the following observation.

Lemma 4.9. *For any choice of cycling $\mathcal{C}$ on a hypergraph H, the digraph $D_{\mathcal{C}}(H)$ is Eulerian.*

The collection of digraphs obtained from varying the choice of cyclings of H provide a collection of superstable configurations. An important observation for us will be the following.

Lemma 4.10. *Suppose H is a hypergraph with nonsink vertices $[n]$ and suppose $\vec{c} \in \mathbb{Z}^n_{\geq 0}$ is a configuration of chips. If a nonempty subset $A \subset [n]$ is ready to fire in H, then for any cycling $\mathcal{C}$ of H, A is ready to fire in $D_{\mathcal{C}}(H)$.*

In what follows, we will consider cyclings $\mathcal{C}$ on H that are induced by an ordering of the vertex set $V(H)$. Note an ordering $v_1 < v_2 < \cdots < v_n$ of $V(H)$ induces a linear order on the elements of each edge $e_i = v_{i_1} < v_{i_2} < \cdots < v_{i_p}$, which can be completed to the cycle $(v_{i_1} v_{i_2} \dots v_{i_p})$. In this case, we say that $\mathcal{C}$ is a *vertex induced cycling*.

It turns out that one can obtain all superstable configurations for H by considering vertex induced cyclings. More precisely we have the following result.

Theorem 4.11. *Let H be a hypergraph with fixed sink vertex q. A configuration $\vec{c}$ is superstable for H if and only if it is superstable for $D_{\mathcal{C}}(H)$ for some choice of vertex induced cycling $\mathcal{C}$.*

Example 4.12. *Consider the hypergraph H and digraph $D = D_{\mathcal{C}}(H)$ depicted in Figure 5, where the cycling is induced by the vertex ordering $v_1 < v_3 < v_2 < v_4$. The corresponding directed Laplacian is given by*

$$L_D = \begin{pmatrix} 3 & -1 & -2 \\ -1 & 2 & 0 \\ 0 & -1 & 2 \end{pmatrix}$$

One can check that the superstable configurations (which coincide with the set of G-parking functions) for D are given by

$$\{000, 100, 010, 001, 200, 110, 101, 201\}.$$

This set provides a (proper) subset of the set of superstable configurations for the hypergraph H, which we have seen in Section 3 to consist of

$$\{000, 100, 010, 001, 110, 101, 011, 200, 210, 201, 111\}.$$

Acknowledgements

The work presented here was conducted as part of an REU at Texas State University in the summer of 2023, sponsored by NSF grant #1757233. We thank NSF and Texas State for the support and the stimulating work environment. We are also grateful to Alex Constantinescu, Sophie Rehberg, and Ben Smith for helpful discussions. Our definition of chip-firing on hypergraphs grew out of conversations that the second author had with Ben in early stages of the project. Dochtermann was partially supported by Simons Foundation Grant #964659.

References

[1] C. Benedetti and N. Bergeron. "The antipode of linearized Hopf monoids". *Algebr. Comb.* **2**.5 (2019), pp. 903–935. DOI.

[2] B. Benson, D. Chakrabarty, and P. Tetali. "G-parking functions, acyclic orientations and spanning trees". *Discrete Math.* **310**.8 (2010), pp. 1340–1353. DOI.

[3] T. Blanton, A. Dochtermann, I. Hong, S. Oh, and Z. Zhan. "Parking functions and chip-firing on hypergraphs". 2025. arXiv:2508.09720. Link.

[4] D. Chebikin and P. Pylyavskyy. "A family of bijections between G-parking functions and spanning trees". *J. Combin. Theory Ser. A* **110**.1 (2005), pp. 31–41. DOI.

[5] D. Dhar. "Self-organized critical state of sandpile automaton models". *Phys. Rev. Lett.* **64** (14 Apr. 1990), pp. 1613–1616. DOI.

[6] T. Kálmán and A. Postnikov. "Root polytopes, Tutte polynomials, and a duality theorem for bipartite graphs". *Proc. Lond. Math. Soc. (3)* **114**.3 (2017), pp. 561–588. DOI.

[7] C. Merino. "Chip firing and the Tutte polynomial". *Ann. Comb.* **1**.3 (1997), pp. 253–259. DOI.

[8] C. Merino. "The chip firing game and matroid complexes". *Discrete models: combinatorics, computation, and geometry (Paris, 2001)*. Discrete Math. Theor. Comput. Sci. Proc., AA. Maison Inform. Math. Discrèt. (MIMD), Paris, 2001, pp. 245–255.

[9] A. Postnikov and B. Shapiro. "Trees, parking functions, syzygies, and deformations of monomial ideals". *Trans. Amer. Math. Soc.* **356**.8 (2004), pp. 3109–3142.

[10] R. P. Stanley and J. Pitman. "A polytope related to empirical distributions, plane trees, parking functions, and the associahedron". *Discrete Comput. Geom.* **27**.4 (2002), pp. 603–634. DOI.

[11] G. P. Steck. "The Smirnov two sample tests as rank tests". *Ann. Math. Statist.* **40** (1969), pp. 1449–1466. DOI.

[12] C. Yan. *Handbook of Enumerative Combinatorics: Parking Functions*. Ed. by M. Bona. 2015. DOI.