PARTIAL PARKING FUNCTIONS
RUI DUARTE AND ANTÓNIO GUEDES DE OLIVEIRA

ABSTRACT. We characterise the Pak-Stanley labels of the regions of a family of hyperplane arrangements that interpolate between the Shi arrangement and the Ish arrangement.

1. Introduction
In this paper, we characterise the Pak-Stanley labels of the regions of the recently introduced family of the arrangements of hyperplanes “between Shi and Ish” (Cf. [6]).

In other words, there is a labelling (due to Pak and Stanley [13]) of the regions of the \(n\)-dimensional Shi arrangement (that is, the connected complements of the complement in \(\mathbb{R}^n\) of the union of the hyperplanes of the arrangement) by the \(n\)-dimensional parking functions, and the labelling in this case is a bijection. Remember that the parking functions can be characterised (see Definition 3.3 below) as

\[ a = (a_1, \ldots, a_n) \in [n]^n \] such that there is a permutation \(\pi \in S_n\) with

\[ a_{\pi_i} \leq i, \text{ for every } i \in [n]. \]

By labelling under the same rules the regions of the \(n\)-dimensional Ish arrangement, we obtain a new bijection between these regions and the so-called Ish-parking functions [5] which can be characterised (see Definition 3.5 below) as

\[ a = (a_1, \ldots, a_n) \in [n]^n \] such that there is a permutation \(\pi \in S_n\) with

\[
\begin{cases}
    a_{\pi_i} \leq i, & \text{for every } i \in [a_1] \\
    \pi_{i+1} < \pi_i, & \text{for every } i \in [a_1 - 1].
\end{cases}
\]

In this paper, we show that the sets of labels corresponding to the arrangements \(A_k^n\) (2 ≤ \(k\) ≤ \(n\)) that interpolate between the Shi and the Ish arrangements (which are \(A_2^n\) and \(A_n^n\), respectively) can be characterised (see Definition 3.8 and Theorem 5.1) as

\[ a = (a_1, \ldots, a_n) \in [n]^n \] such that there is a permutation \(\pi \in S_n\) with

\[
\begin{cases}
    a_{\pi_i} \leq i, \text{ for every } i \in [a_1] \text{ and for every } i \in [n] \text{ such that } \pi_i \geq k; \\
    \pi_{i+1} < \pi_i, \text{ for every } i \in [a_1 - 1] \text{ such that } \pi_i < k.
\end{cases}
\]

We call these sets of labels partial parking functions and note that they all have the same number of elements, viz. \((n + 1)^{n-1}\), by [3] Section 2 and Theorem 3.7.
2. Preliminaries

Consider, for an integer $n \geq 3$, hyperplanes of $\mathbb{R}^n$ of the following three types. Let, for $1 \leq i < j \leq n$,

$$C_{ij} = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid x_i = x_j\},$$

$$S_{ij} = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid x_i = x_j + 1\},$$

$$I_{ij} = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid x_1 = x_j + i\}$$

and define, for $2 \leq k < n$,

$$A^k_n := \{C_{ij} \mid 1 \leq i < j \leq n\} \cup \{I_{ij} \mid 1 \leq i < j \leq n \land i < k\} \cup \{S_{ij} \mid k \leq i < j \leq n\}$$

Note that $A^2_n = \text{Shi}_n$, the $n$-dimensional Shi arrangement, and $A^n_n = \text{Ish}_n$, the $n$-dimensional Ish arrangement introduced by Armstrong [1].

2.1. The Pak-Stanley labelling. Similarly to what Pak and Stanley did for the regions of the Shi arrangement (cf. [13]), we may represent a region $\mathcal{R}$ of $A = A^n_k$ as follows. Suppose that $x = (x_1, \ldots, x_n) \in \mathcal{R}$ and $x_{w_1} > \cdots > x_{w_n}$ for a given $w = (w_1, \ldots, w_n) \in \mathcal{S}_n$. Let $\mathcal{H}$ be the set of triples $(i, j, a_{ij})$ such that $i, j, a_{ij} \in \mathbb{N}$, $1 \leq i < j \leq n$, $x_i > x_j$ and $a_{ij} - 1 < x_i - x_j < a_{ij}$. Then, of course,

$$(2.1) \quad \mathcal{R} = \left\{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid x_{w_1} > x_{w_2} > \cdots > x_{w_n}, \ a_{ij} - 1 < x_i - x_j < a_{ij}, \ \forall (i, j, a_{ij}) \in \mathcal{H} \right\}. \tag{2.1}$$

We represent $\mathcal{R}$ by $w$, decorated with one labelled arc for each hyperplane of $\mathcal{H}$, as follows. Given $(i, j, a_{ij}) \in \mathcal{H}$, the arc connects $i$ with $j$ and is labelled $a_{ij}$, with the following exceptions: if $i \leq j < p \leq m$, $(i, m, a_{im}), (j, p, a_{jp}) \in \mathcal{H}$ and $a_{jp} = a_{im}$, then we omit the arc connecting $j$ with $p$. Note that, given $i \leq j < p \leq m$, forcibly

$$a_{im} > x_i - x_m \geq x_i - x_p \geq x_j - x_p$$

and so $a_{im} \geq a_{jp}$. In the left-hand side of Figure 1 the regions of Ish$_3$ are thus represented.

The Pak-Stanley labelling of these regions may be defined as follows. As usual, let $e_i$ be the $i$th element of the standard basis of $\mathbb{R}^n$, $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$.

**Definition 2.1** (Pak-Stanley labelling [13], ad.). Let $\mathcal{R}_0$ be the region defined by

$$x_{n+1} > x_1 > x_2 > \cdots > x_n$$

(limited by the hyperplanes of equation $x_j = x_{j+1}$ for $1 \leq j < n$ and by the hyperplane of equation $x_1 = x_{n+1}$). Then label $\mathcal{R}_0$ with $\ell(A^n_k, \mathcal{R}_0) := (1, \ldots, 1)$, and, given two regions $\mathcal{R}_1$ and $\mathcal{R}_2$ separated by a unique hyperplane $H$ of $A^n_k$ such that $\mathcal{R}_0$ and $\mathcal{R}_1$ are on the same side of $H$, label the regions $\mathcal{R}_1$ and $\mathcal{R}_2$ so that

$$\ell(A^n_k, \mathcal{R}_2) = \ell(A^n_k, \mathcal{R}_1) + \begin{cases} e_i, & \text{if } H = C_{ij} \text{ for some } 1 \leq i < j \leq n; \\ e_j, & \text{if } H = S_{ij} \text{ or } H = I_{ij} \text{ for some } 1 \leq i < j \leq n. \end{cases}$$

Then it is not difficult to find the label of a given region, directly (cf. Stanley [13] in the case where $A$ is the Shi arrangement). Let again $\mathcal{R}$ be defined as in (2.1) and

- **take** $t = t(w) = (t_1, \ldots, t_n)$ such that $t_{w_i} = \left| \left\{ j \leq i \mid w_j \geq w_i \right\} \right|$.  
- **add** $(a_{ij} - 1)e_{w_j}$ to $t$ for every hyperplane $(i, j, a_{ij}) \in \mathcal{H}$.  

In fact, it is easy to see that \( t(w) \) is the label of the region of the braid arrangement \(^5\) on the Pak-Stanley labelling, which is also the label of the (unique) region of \( A \) contained in \( \mathcal{R}' \) adjacent to the line defined by \( x_1 = \cdots = x_n \). The rest follows easily.

Let us give another example, now with \( A = A_4^1 = \text{Ish}_4 \) (note that our representation in the Ish case, since 1 is the initial point of all arcs, is equivalent to the representation already given by Armstrong and Rhoades \(^2\) and used by Leven, Rhoades and Wilson \(^8\)). Recall that this arrangement is formed by the hyperplanes of equation \( x_i = x_j \) for \( 1 \leq i < j \leq 4 \), and by the hyperplanes of equation \( x_i - x_j = a_{ij} \) for \( 1 \leq a_{ij} < j \leq 4 \).

Now, for example, \( \ell(A_4^1, 1324) = 1211 \). In fact, in this case, \( w = 1324 \) and \( \mathcal{H} = \{(1,3,1),(1,2,1),(1,4,1)\} \). Hence, \( t(w) = 1211 \), since 2 is the only element in \( w \) with a greater element on its left, 3, and this element is unique. At last, \( \sum_{(i,j,a_{ij}) \in \mathcal{H}} (a_{ij}-1)e_{w_j} = (0,0,0,0) \).

For another example, note that \( \ell(A_4^1, 1324) = 1323 \), since \( \mathcal{H} = \{(1,3,2),(1,4,3)\} \).

Finally, note that we have \( 2312 = 2211 + 0101 = \ell(A_4^1, 3124) \) whereas \( 2312 = 2311 + 0001 = \ell(A_4^2, 3142) = \ell(A_4^3, 3142) \). In fact, in both arrangements \( (A_4^2 \text{ and } A_4^3) \) \( \mathcal{H} = \{(1,2,1),(1,4,1)\} \) for the region \( 3142 \), and hence for no \( a \in \mathbb{N} \) is \( (3,4,a) \in \mathcal{H} \), although the hyperplane of equation \( x_3 - x_4 = 1 \) belongs to both arrangements \( ^9 \).

In the right-hand side of Figure 1 the Pak-Stanley labelling of the regions of \( \text{Ish}_3 \) is shown. In dimension \( n \), these labels form the set of \( n \)-dimensional Ish-parking functions, characterized in a previous article \(^6\). The labels of the regions of \( 
\text{Shi}_n \) form the set of

\(^{(*)}\) i.e., the arrangement \( \{C_{ij} \mid 1 \leq i < j \leq n\} \).

\(^{†}\) Note that \( \ell(A_4^1, 3124) = 2313 \).
n-dimensional parking functions, defined below, as proven by Pak and Stanley in their seminal work [12].

Parking functions and Ish-parking functions, as well as the Pak-Stanley labels of $A^k_n$ for $2 < k < n$, are graphical parking functions as introduced by Postnikov and Shapiro [11] and reformulated by Mazin.

3. Graphical parking functions

Definition 3.1 (Ish, ad.). Let $G = (V, A)$ be a (finite) directed loopless connected multigraph, where $V = [n]$ for some natural $n$. Then $a = (a_1, \ldots, a_n) \in \mathbb{N}^n$ is a $G$-parking function if for every non-empty subset $I \subseteq [n]$ there exists a vertex $i \in I$ such that the number of arcs $(i, j) \in A$ with $j \notin I$, counted with multiplicity, is greater than $a_i - 2$.

Given the arrangement $A^k_n$, consider a multigraph $G^k_n$ where for each hyperplane of equation $x_i = x_j$ there is a corresponding arc $(i, j)$, and for each hyperplane of equation $x_i = x_j + a$ with $a \in \mathbb{N}$ there is a corresponding arc $(j, i)$. In Figure 2, the graphs $G^2_4$, $G^3_4$ and $G^4_4$ are shown. Note that $G^2_n$ is the complete digraph $K_n$ on $n$ vertices. We will use the following crucial result.

Theorem 3.2 (Mazin [9], ad.). For every $2 \leq k \leq n$, the set

$$\{ \ell(A^k_n, R) \mid R \text{ is a region of } A^k_n \}$$

is the set of $G^k_n$-parking functions.

3.1. Parking functions.

Definition 3.3. The $n$-tuple $a = (a_1, \ldots, a_n) \in [n]^n$ is an $n$-dimensional parking function if

$$\left| \{ j \in [n] \mid a_j \leq i \} \right| \geq i, \quad \forall i \in [n].$$

Note that parking functions (sometimes called classical parking functions) are indeed $G^2_n$-parking functions, being $G^2_n = K_n$, the complete digraph on $[n]$. In fact, suppose that $a$ is a $K_n$-parking function. Then, given $i \in [n]$, let $I = \{ j \in [n] \mid a_j > i \}$. If

\footnote{With this definition, $1 := (1, \ldots, 1) \in [n]^n$ is a parking function and $0 := (0, \ldots, 0) \in [n]^n$ is not. Parking functions are sometimes defined differently, so as to contain $0$ (and not $1$). In that case, they are the elements of form $b = a - 1$ for $a$ a parking function in the current sense.}
I = \emptyset, then |\{j \in [n] \mid a_j \leq i\}| = n \geq i. If I \neq \emptyset, then there is i \in I such that 
\{(i, j) \in A \mid j \notin I\} \geq a_i - 1 and so 
|\{j \in [n] \mid a_j \leq i\}| = |\{(i, j) \in A \mid j \notin I\}| \geq a_i - 1 \geq i,
the last inequality since i \in I. The other direction is obvious.

Now, consider the following algorithm.

### Parking Algorithm

**Input:** a \in [n]^n
1: street_parking = (0, \ldots, 0) \in \mathbb{Z}^{2n}
2: foreach i \in [n] in descending order do
3: \quad p = a(i)
4: \quad while street_parking(p) \neq 0 do
5: \quad \quad increase p
6: \quad end while
7: \quad parking_place(i) = p.
8: \quad street_parking(p) = i
9: end for

**Output:** street_parking, parking_place

Konheim and Weiss [7] introduced this concept and showed that a is a parking function if and only if the first n coordinates of street_parking form a permutation of [n] (the inverse of parking_place, in this case) at the end of the execution.

This is the reason for the name parking function, for we may see — as Konheim and Weiss did — a as the record of the preferred parking slots of n drivers who want to park in a one-way street with exactly n places, forced to go ahead of their favorite place when the place is already taken. Then a is a parking function exactly when all the cars can thus park on the street.

More precisely, we say that a parks i \in [n] if parking_place(i) \leq n. Parking functions are those which park every element, or, equivalently, if we set

first_free := \min\{i \in [2n] \mid \text{street_parking}(i) = 0\} and

occupied_positions = \text{street_parking}^{-1}([n]),

those for which first_free = n + 1 or those for which occupied_positions = [n]. Note that by Definition 3.3 \mathcal{S}_n acts on the set PF_n of size n parking functions: if w \in \mathcal{S}_n and \text{w}(a) := a \circ w = (a_{\ell_1}, \ldots, a_{\ell_n}), then a \in PF_n if and only if w(a) \in PF_n. In fact, this is a particular case of a more general situation, described in the following result.

**Lemma 3.4.** Given a \in [n]^n and w \in \mathcal{S}_n,

occupied_positions(a) = occupied_positions(a \circ w).

**Proof.** It is sufficient to prove the claim when w is the transposition (i i + 1) for some i \in [n - 1]. Let b := a \circ w = (b_1, \ldots, b_n) = (a_1, \ldots, a_{i-1}, a_{i+1}, a_i, a_{i+2}, \ldots, a_n), \alpha := parking_place(i + 1) \geq a_{i+1} and \beta := parking_place(i) \geq a_i when the Parking Algorithm is applied to a.

Suppose that \beta < \alpha. Then, since a_i \leq \beta, \beta = parking_place(i) and \alpha = parking_place(i + 1) when the algorithm is applied to b. Now, suppose that \alpha < \beta. Hence, if b_{i+1} (= a_i) > \alpha, then \beta = parking_place(i + 1) and \alpha = parking_place(i) when the algorithm is applied to b, and if b_{i+1} \leq \alpha, then \alpha = parking_place(i + 1) and \beta = parking_place(i). \qed
3.2. Ish-parking functions. The labels of the regions of Ish$_n$, the Ish-parking functions, are characterized as follows [6, Proposition 3.12].

**Definition 3.5.** Let $a = (a_1, \ldots, a_n) \in \mathbb{N}^n$ and $1 < m \leq n$. The centre of $a$, $Z(a)$, is the (possibly empty) largest set $X = \{x_1, \ldots, x_m\}$ contained in $[n]$ with $n \geq x_1 > \cdots > x_m \geq 1$ and the property (§) that $a_{x_1} \leq i$ for every $i \in [m]$. The restriction to $a(a_1, \ldots, a_m)$ of all functions that park every element of $[k, n]$ is a particular case of a defective parking function [3, pp. 3], that is Ish-parking functions [3, Proposition 3.12]. Hence, the number $T_k$ of all functions that park every element of $[k, n]$ is $n^{k-1}c(n, n+1-k, 0)$, where $c(n, m, k)$ is the number of $(n, m, k)$-defective parking functions [3, pp. 3], that is

$$T_k = kn^{k-1}(n+1)^{n-k}.$$  

**Remark 3.9.** A function $a \in [n]^n$ parks every element of $[k, n]$ if and only if

$$\left| \{ j \in [k, n] \mid a_j \leq i \} \right| + k - 1 \geq i, \quad \forall i \in [k, n].$$

In fact, since this property does not depend on the first $k-1$ coordinates of $a$ we may replace each one of them by 1. Now, the new function parks every element of $[k, n]$ if and only if it is a parking function.

Indeed, $k$-partial parking functions are exactly the $G^n_k$-parking functions. But to prove it we still need a different tool.\(^{(*)}\)  

\(^{(*)}\)Note that if this property holds for both $X, Y \subseteq [n]$ then it holds for $X \cup Y$, and so this concept is well-defined (Cf. [4, 5, 6]). The centre was previously called the reverse centre [6].
4. The DFS-Burning Algorithm

We want to characterise the $G_k^n$-parking functions for every $k, n \in \mathbb{N}$ such that $2 \leq k \leq n$. Similarly to what we did for the characterisation of the Ish-parking functions [10] (the case $k = n$), our main tool is the DFS-Burning Algorithm of Perkinson, Yang and Yu [10] (Cf. Figure 3). Recall that this algorithm, given $a \in [n]^n$ and a multiple digraph $G$, determines whether $a$ is a $G$-parking function by constructing in the positive case an oriented spanning subtree $T$ of $G$ that is in bijection with $a$ [10] [6]. The Tree to Parking Function Algorithm (Cf. Figure 3 on the right) builds $a$ out of $T$ (and $G$), thus defining the inverse bijection.

Recall that the algorithm is not directly applied to the multidigraph $G$. Indeed, it is applied to another digraph, $\overline{G}$, with one more vertex, 0, and set of arcs $\overline{A}$ defined so that:

- For every vertex $v \in [n]$, $(0, v) \in \overline{A}$;
- For every unique arc $(v, w) \in A$, $(w, v) \in \overline{A}$;
- For every multiple arc $(v, w) \in A$ that occurs $\ell$ times ($\ell > 1$), $(w, v + j \ell) \in \overline{A}$, for $j \in [\ell]$.

We use the following result, which is an extension to directed multigraphs of the work of Perkinson, Yang and Yu [10].

**Proposition 4.1** ([10, Proposition 3.2]). Given a directed multigraph $G$ on $[n]$ and a function $a : [n] \to \mathbb{N}_0$, $a$ is a $G$-parking function if and only if the list burnt_vertices at the end of the execution of the DFS-Burning Algorithm applied to $\overline{G}$ includes all the vertices in $V = \{0\} \cup [n]$.

Note that although the order of the vertices in neighbours is not relevant in the context of Proposition 4.1 it is indeed relevant in other contexts, like that of Lemma 4.3 (Cf. [6, Remark 3.4]). We define the order in $G_k^n$ so that:

1. neighbours$(0)$ if formed by the arcs of form $(0, i)$ for every $i \in [n]$; we sort neighbours$(0)$ based on the value of $i$, in descending order.
2. There is an arc of form $(1, i+m\ell)$ for every $i > 1$ and every $0 \leq m \leq \min\{i, k\} - 2$. We sort neighbours$(1)$ by the value of $i$ in descending order, breaking ties by the value of $m$, again in descending order. For example, in $A_k^3$ (Cf. Figure 4),
   \[
   \text{neighbours}(1) = \langle 8, 4, 7, 3, 2 \rangle.
   \]
3. For every $1 \leq m < i$ there is a unique arc $(i, m)$. Exactly when $i \geq k$, there is also an arc of form $(i, m)$ for every $i < m \leq n$. In all cases, we sort neighbours$(i)$ by the value of $m$ in descending order.

**Example 4.2.** We apply the DFS-Burning Algorithm to $a = 4213 \in [4]^4$ with the three different graphs associated with $n = 4$. Actually, $a$ is a parking function —that is, a label of a region of $A_4^2 = \text{Shi}_4$— but neither a label of a region of $A_4^3$ nor of $A_4^4$ = Ish$_4$. In fact, in the first case, where neighbours = $\langle \langle 4, 3, 2, 1 \rangle, \langle 4, 3, 2 \rangle, \langle 4, 3, 1 \rangle, \langle 4, 2, 1 \rangle, \langle 3, 2, 1 \rangle \rangle$ (Cf. the left table in the bottom of Figure 3), when the algorithm is applied with $G = \overline{G}_k^n$ to $a$, it calls $\text{DFS\_FROM}(i)$ with $i = 0$, assigns $j = 4$ and then, since $a(j) \neq 1$, $(0, 4)$ is joined to dampened_edges. This is represented on the left-hand table below with the inclusion of $0_1$ in the top box of column 4. Next assignment, $j = 3$. Since now $a(3) = 1$, $(0, 3)$ is joined to tree_edges and $\text{DFS\_FROM}$ is called with $i = 3$. Then, $0_2$ is written in the only box of column 3. At the end, burnt_vertices = $\langle 0, 3, 2, 4, 1 \rangle$, which proves that 4213 is a $G_2^2$-parking function, that is, a standard parking function.
DFS-Burning Algorithm (ad.)

Input: a: [\frac{n}{2}] \rightarrow \mathbb{N}
1: burnt_vertices = \{0\}
2: dampened_edges = \{
3: tree_edges = \{
4: execute \text{DFS_FROM}(0)

Output: burnt_vertices, tree_edges and dampened_edges

AUXILIARY FUNCTION
5: function \text{DFS_FROM}(i)
6: foreach j in neighbours(i) do
7:     \text{if} \ j_n \notin burnt_vertices \text{then}
8:         \text{if} \ a(j_n) = 1 \text{then}
9:             append (i, j) to tree_edges
10:        append j_n to burnt_vertices
11:    execute \text{DFS_FROM}(j_n)
12: else
13:    append (i, j) to dampened_edges
14:    a(j_n) = a(j_n) + 1
15: end if
16: end if
17: end for
18: end function

Tree to Parking Function Algorithm (ad.)

Input: Spanning tree T rooted
1: at r with edges directed away from root.
2: burnt_vertices = \{r\}
3: dampened_edges = \{
4: a = \{1, \ldots, 1\}
5: execute \text{TREE_FROM}(r)

Output: a: \mathbb{V} \setminus \{r\} \rightarrow \mathbb{N}

AUXILIARY FUNCTION
6: function \text{TREE_FROM}(i)
7: foreach j in neighbours(i) do
8:     j_n = \text{Mod}(j, n)
9:     if j_n \notin burnt_vertices then
10:        if (i, j) is an edge of T
11:            append j_n to burnt_vertices
12:        execute \text{TREE_FROM}(j_n)
13: else
14:    a(j_n) = a(j_n) + 1
15: append (i, j) to dampened_edges
16: end if
17: end if
18: end for
19: end function

Figure 3. DFS-Burning Algorithm and inverse

in dimension 4. The respective spanning tree may be defined by the collection of arcs, tree_edges = \{(0, 3), (0, 2), (2, 4), (0, 1)\}.

| i | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| j | 0 | 1 | 2 | 3 | 4 |
| 0 | 4 | 4 | 4 | 4 | - |
| 1 | 3 | 3 | 3 | 3 | - |
| 2 | 2 | 2 | 2 | 2 | - |
| 3 | 1 | 1 | 1 | 1 | - |

$\overline{\mathcal{G}}_4^r = K_4$

Figure 4. Lists of neighbours and execution of the DFS-Burning Algorithm

We believe that now the content of the tables is self-explanatory. Just note that the entry $i_k$ in column $j$ means that arc $(i, j)$ is the $k$th arc to be inserted. Note also that the elements of the bottom row are those in $a^{-1}(1)$, and thus represent elements from tree_edges, whereas the remaining entries represent elements from dampened_edges.

Finally, note that the algorithm runs in the second graph by choosing the same arcs up to the seventh arc, which is not $(2, 4)$ since $4 \notin \text{neighbours}(2)$ in this graph. Since burnt_vertices $\neq \{0, 1, \ldots, 4\}$ at the end of the execution for the two last graphs, we
verify that 4213 is neither a label of the regions of \( A_k^n \) nor an Ish-parking function (in fact, \( 1 \notin Z(4213) = \{2, 3\} \)).

**Lemma 4.3.** Let \( a \in [n]^n \) be the input of the DFS-Burning Algorithm applied to \( \overline{G}_k \) (\( 2 \leq k \leq n \)) as defined above, and suppose that, at the end of the execution, the list of burnt vertices is \( \text{burnt vertices} = \{0=i_0, i_1, \ldots, i_m\} \). Let \( i_p = \min\{i_1, \ldots, i_m\} \) and suppose that \( i_p < k \). Then either \( i_p = 1 \) or \( p = m \). In any case,

\[
Z(\tilde{a}_k) = \{i_1, \ldots, i_p\}.
\]

**Proof.** Let \( M = \text{dampened edges} \cup \text{tree edges} \) (as sets), and note that when \((j, m)\) is added to \( M \), in Line 10 or Line 14, then \( j \in \text{burnt vertices} \) and \( m \notin \text{burnt vertices} \).

If \( j = 0 = i_0 \) and \( m = i_1 \) then \( a(m) = 1 \). In general, if \( j = i_p \) and \( m = i_{p+1} \) then \( a_m \leq a_j + 1 \) and if \( 1 < j < k \) then \( m < j \) since \( m \in \text{neighbours}(j) \). Hence,

\[
\{i_1, \ldots, i_p\} \subseteq Z(\tilde{a}_k).
\]

For the converse, note that, by definition of \( \overline{G}_k \), if \( j \in \text{neighbours}(p) \) for some \( p > 1 \) and \( p \neq m < j \), then also \( m \in \text{neighbours}(p) \). Thus, if at the end of the execution \( j \in \text{burnt vertices} \), \( m < j \) and \( a_m \leq a_j + 1 \), then also \( m \in \text{burnt vertices} \). \( \square \)

5. **Main Theorem**

**Theorem 5.1.** The \( G_k^n \)-parking functions are exactly the \( k \)-partial parking functions. Their number is

\[
(n + 1)^{n-1}.
\]

We know that there are \((n+1)^{n-1}\) regions in the \( A_k^n \) arrangement of hyperplanes, which are bijectively labeled by the \( G_k^n \)-parking functions [6, Theorem 3.7].

Hence, all we have to prove is the first sentence. This is an immediate consequence of the following Lemma [5.2] and of the fact that the \( G \)-parking functions are those functions for which the DFS-Burning Algorithm burns all vertices during the whole execution.

**Lemma 5.2.** Let \( a \in [n]^n \) be the input of the DFS-Burning Algorithm applied to \( \overline{G}_k \) (\( 2 \leq k \leq n \)) as defined above, and consider \( \text{burnt vertices} = \{0=i_0, i_1, \ldots, i_m\} \) at the end of the execution. Then the following statements are equivalent:

1. \( a \) parks every element of \([k, n]\) and for some \( 1 \leq p \leq m \), \( i_p = 1 \);
2. \( a \) is a \( k \)-partial parking function;
3. as a set, \( \text{burnt vertices} = \{0\} \cup [n] \) or, equivalently, \( m = n \).

**Proof.**

(5.2.1) \( \Rightarrow \) (5.2.2). Since \( a \) parks all the elements of \([k, n]\), it is sufficient to show that \( 1 \in Z(\tilde{a}_k) \), which follows from Lemma [4.3].

(5.2.2) \( \Rightarrow \) (5.2.3). Suppose that \( 1 \) belongs to the centre of \( \tilde{a}_k \) but there is a greatest element \( j \in [n] \) which is not in \( \text{burnt vertices} \) at the end of the execution. Suppose first that \( j < k \). Then, during the execution of the algorithm (more precisely, during the execution of Line 14) the value of \( a(j) \) has decreased once for \( i = 0 \) (that is, as a neighbour of 0), once for each value of \( i > j \) (in a total of \( n - j \)), since \( i \in \text{burnt vertices} \) by definition of \( j \), and \( j - 1 \) times for \( i = 1 \), and is still greater than zero. Hence \( a(j) > n \), which is absurd.
Now, suppose that \( j \geq k \), and let 
\[
\alpha = \min \{ a_i \mid i \notin \text{burnt\_vertices} \}
\]
\[
p = \min \{ q \in [k, n] \mid a_q = \alpha \}
\]
and
\[
A = \{ q \in [k, n] \mid a_q < \alpha \},
\]
so that
\[
\begin{cases} 
|A| \geq \alpha - k \quad \text{(since a parks \( \alpha \))}; \\
A \subseteq \text{burnt\_vertices}.
\end{cases}
\]

Again, during the execution of Line 14 the value of \( \alpha \) has decreased once for \( i = 0 \), once for each value of \( i \neq p \) in \( \text{burnt\_vertices} \cap [k, n] \supseteq A \), and \( k - 1 \) times for \( i = 1 \), and is still greater than zero. This means that \( \alpha - 1 - (\alpha - k) - (k - 1) > 0 \), which is not possible.

\((5.2.3) \implies (5.2.1)\). Contrary to our hypothesis, we admit that all the elements of \([n]\) belong to \( \text{burnt\_vertices} \) at the end of the execution, but that for some \( j \in [k, n] \)
\[
A_j = \{ q \in [k, n] \mid a_q > j \}
\]
verifies
\[
|A_j| \geq n - j + 1.
\]
Remember that \( \text{burnt\_vertices} = \langle 0=i_0, i_1, \ldots, i_n \rangle \) is the ordered list of burnt vertices at the end of the execution and let
\[
p = \min \{ q \in [k, n] \mid i_q \in A_j \},
\]
so that \( \alpha = a_p > j \). Since \( p \in \text{burnt\_vertices} \), the number of elements of the set \( \text{dampened\_edges} \cup \text{tree\_edges} \) of form \((i, p)\), \( \alpha \), must be greater than \( j \). But when \( p \) was burned, at most \((n - k + 1) - (n - j + 1) = j - k \) elements of \([k, n]\) different from \( p \) were already burned, and even if 0 and 1 were also burned, the number of edges could not be greater than \((j - k) + 1 + (k - 1) = j \), a contradiction. \( \square \)

Acknowledgements. This work was partially supported by CMUP (UID/MAT/00144/2013) and CIDMA (UID/MAT/04106/2013), which are funded by FCT (Portugal) with national (ME) and European structural funds through the programs FEDER, under the partnership agreement PT2020.

REFERENCES

[1] D. Armstrong, Hyperplane arrangements and diagonal harmonics, \textit{J. Combinatorics} \textbf{4} (2013) pp.157–190. URL: \url{http://dx.doi.org/10.4310/JOC.2013.v4.n2.a2}

[2] D. Armstrong and B. Rhoades, The Shi arrangement and the Ish arrangement, \textit{Trans. Amer. Math. Soc.} \textbf{364} (2012) pp. 1509–1528. URL: \url{https://doi.org/10.1090/S0002-9947-2011-05521-2}

[3] P.J. Cameron, D. Johannsen, T. Prellberg, and P. Schweitzer, Counting defective parking functions, \textit{Electron. J. Combin.} \textbf{15} (2008), Research Paper 92.

[4] R. Duarte, A. Guedes de Oliveira, The braid and the Shi arrangements and the Pak-Stanley labelling, \textit{European J. Combin.} \textbf{50} (2015) pp. 72–86. URL: \url{http://dx.doi.org/10.1016/j.ejc.2015.03.017}

[5] R. Duarte, A. Guedes de Oliveira, The number of parking functions with center of a given length, submitted. URL: \url{https://arxiv.org/abs/1611.03707}

[6] R. Duarte, A. Guedes de Oliveira, Between Shi and Ish, \textit{Disc. Math.} \textbf{341}(2) (2018) pp.388–399. URL: \url{http://dx.doi.org/10.1016/j.disc.2017.09.008}
[7] A.G. Konheim and B. Weiss, An occupancy discipline and applications, *SIAM J. Appl. Math.* **14** (1966) pp. 1266–1274. URL: [https://doi.org/10.1137/0114101](https://doi.org/10.1137/0114101)

[8] E. Leven, B. Rhoades and A. T. Wilson, Bijectons for the Shi and Ish arrangements, *European J. Combin.* **39** (2014) pp. 1–23. URL: [http://dx.doi.org/10.1016/j.ejc.2013.12.001](http://dx.doi.org/10.1016/j.ejc.2013.12.001)

[9] M. Mazin, Multigraph hyperplane arrangements and parking functions. *Ann. Comb.* **21** (2017) pp. 653–661. URL: [https://doi.org/10.1007/s00026-017-0368-7](https://doi.org/10.1007/s00026-017-0368-7)

[10] D. Perkinson, Q. Yang and K. Yu, G-parking functions and tree inversions, *Combinatorica* **37** (2017) pp. 269–282. URL: [http://dx.doi.org/10.1007/s00493-015-3191-y](http://dx.doi.org/10.1007/s00493-015-3191-y)

[11] A. Postnikov and B. Shapiro, Trees, parking functions, syzygies, and deformations of monomial ideals, *Trans. Amer. Math. Soc.* **356** (2004) pp. 3109–3142. URL: [https://doi.org/10.1090/S0002-9947-04-03547-0](https://doi.org/10.1090/S0002-9947-04-03547-0)

[12] R. P. Stanley, Hyperplane arrangements, interval orders and trees, *Proc. Nat. Acad. Sci.* **93** (1996) pp. 2620–2625.

[13] ————, An introduction to hyperplane arrangements, in: E. Miller, V. Reiner, B. Sturmfels (Eds.), *Geometric Combinatorics*, in: IAS/Park City Mathematics Series **13** (2007), A.M.S. pp. 389–496.

[14] C. Yan, Parking Functions, in *Handbook of Enumerative Combinatorics*, M. Bóna (ed.), Discrete Math. and Its Appl., CRC Press, Boca Raton-London-New York (2015) pp. 589–678.

CIDMA and Department of Mathematics, University of Aveiro, 3810-193 Aveiro, Portugal

*E-mail address:* rduarte@ua.pt

CMUP and Department of Mathematics, Faculty of Sciences, University of Porto, 4169-007 Porto, Portugal

*E-mail address:* agoliv@fc.up.pt