ICECA

International Conference
Enumerative Combinatorics and Applications
University of Haifa – Virtual – September 4-6, 2023

MVP parking functions, permutation subgraphs and Motzkin paths

Haoyue Zhu
Joint work with Thomas Selig

Department of Computing, School of Advanced Technology, Xi'an Jiaotong-Livepool University

Haoyue.Zhu18@student.xjtlu.edu.cn

Abstract: In parking problems, a given number of cars enter a one-way street sequentially, and try to park according to a specified preferred spot in the street. Various models are possible depending on the chosen rule for *collisions*, when two cars have the same preferred spot. We study a recent model introduced by Harris, Kamau, Mori, and Tian in recent work, called the *MVP parking problem*. In this model, priority is given to the cars arriving later in the sequence. When a car finds its preferred spot occupied by a previous car, it "bumps" that car out of the spot and parks there. The earlier car then has to drive on, and parks in the first available spot it can find. If all cars manage to park through this procedure, we say that the list of preferences is a (MVP) parking function.

We study the outcome map of MVP parking functions, which describes in what order the cars end up. In particular, we link the fibres of the outcome map to certain subgraphs of the inversion graph of the outcome permutation. This allows us to reinterpret and improve bounds from Harris *et al.* on the fibre sizes. We also focus on a subset of parking functions, called *Motzkin parking functions*, where every spot is preferred by at most two cars. We generalise results from Harris *et al.*, and exhibit rich connections to Motzkin paths.

1. Introduction

In this section we introduce classical and MVP parking functions, and their outcome maps. Throughout the paper, n represents a positive integer, and we denote $[n] := \{1, \cdots, n\}$.

1.1. **Classical and MVP parking functions.** A *parking preference* is a vector $p = (p_1, \cdots, p_n) \in [n]^n$. We think of p_i as denoting the preferred parking spot of car i in a car park with n labelled spots. The car park is one-directional, with cars entering on the left in spot 1 and driving through to spot n (or until they park). Cars enter sequentially, in order $1, \cdots, n$. If the spot p_i is unoccupied when car i enters, it simply parks there. If this is not the case, then a previous car $j < i$ has already occupied spot p_i. We call this a *collision* between cars i and j.

In classical parking functions, such collisions are handled by giving priority to the earlier car j. This means that car i is forced to drive on, and looks for the first unoccupied spot $k > p_i$. If no such spot exists, then car i exits the car park, having failed to find a spot. We say that p is a *parking function* if all cars manage to park. Parking functions were originally introduced by Konheim and Weiss [10] in their study of hashing functions. Since then, they have been a popular research topic in Mathematics and Computer Science, with rich connections to a variety of fields such as graph theory, representation theory, hyperplane arrangements, discrete geometry, and statistical physics [3, 6, 7, 8, 12, 14]. We refer the interested reader to the excellent survey by Yan [15].

One may notice that the collision rule for parking functions has many possible variations, and indeed many variants of parking functions have been studied in the literature (see e.g. [2, 11, 16]). To mention just one variation, we may allow cars to reverse a fixed number of spots before driving on. This rule is called the *Naples parking rule* (see [4, 5]).

In this paper, we are interested in another variant called *MVP parking functions*. In this model, if there is a collision between two cars $j < i$, priority is given to the later car i. In other words, car i will park in its preferred spot p_i. If that spot is already occupied by a previous car j, then car j gets "bumped" out, and has to drive on. It then (re-)parks in the first available spot $k \geq p_i$. Note that bumpings do not propagate: the "bumped" car j does not subsequently bump any other car. If all cars manage to park in this process, we say that p is an MVP parking function.

It is in fact straightforward to check that a parking preference p is an MVP parking function if, and only if, p is a (classical) parking function. Indeed, in both MVP and classical processes, in determining whether all cars can park, the labels of the cars are unimportant: all that matters is which set of spots is occupied at any given time. We denote PF_n or MVP_n the set of parking functions of length n. These sets are the same due to the previous observation, but it will be convenient to use different notation depending on whether we are considering the classical or MVP parking process.

1.2. **The outcome maps.** While the sets of MVP and classical parking functions are the same, these two processes differ in their *outcome map*. This map describes where the cars end up. More precisely, if p is a parking function, its *outcome* is a permutation $\pi = \pi_1 \cdots \pi_n$, where for all $i \in [n]$, π_i is the label of the car occupying spot i when all cars have parked. The classical, resp. MVP, outcome map, denoted $\mathcal{O}_{\mathrm{PF}_n}$, resp. $\mathcal{O}_{\mathrm{MVP}_n}$, is then the map $p \mapsto \pi$ describing the outcome of the classical, resp. MVP, parking process.

Example 1.1. Consider the parking function $p = (3, 1, 1, 2)$. Under the classical parking process, car 1 first parks in spot 3, followed by car 2 parking in spot 1. Then car 3 wishes to park in spot 1 but cannot do so, so it drives on and parks in spot 2 (the first available spot at this point). Finally, car 4 wishes to park in spot 2. However, 2 is occupied, so car 4 drives on: 3 is also occupied (by car 1), so car 4 ends up parking in spot 4. Finally, we get the outcome $\pi = \mathcal{O}_{\mathrm{PF}_4}(p) = 2314$.

Now consider the same parking function p, but for the MVP parking process. Again, cars 1 and 2 park in spots 3 and 1 respectively. Now car 3 arrives, and sees car 2 in its preferred spot (spot 1). It bumps car 2 out of spot 1, forcing it to drive on. Spot 2 is available, so car 2 parks there. Finally car 4 arrives and sees car 2 in its preferred spot (spot 2). It bumps car 2, forcing it to drive on and park in the only remaining spot, which is spot 4. Finally, we get the outcome $\pi = \mathcal{O}_{\mathrm{MVP}_4}(p) = 3412$.

In this paper, we mainly study the *fibres* of the MVP outcome map. That is, for a given, fixed permutation $\pi \in S_n$, we are interested in the set $\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi)$ of parking functions whose outcome is the permutation π. For the sake of brevity we omit proofs here: these can be found in an upcoming companion paper.

2. General case

In this section we study the MVP outcome map in the general setting. We will give an interpretation of the fibres in terms of certain subgraphs of the inversion graph of the outcome permutation.

2.1. **Inversion graphs and subgraphs.** Given a permutation $\pi = \pi_1 \cdots \pi_n \in S_n$, we say that a pair (j, i) is an *inversion* of π if $j < i$ and $\pi_j > \pi_i$. We denote $\mathrm{Inv}(\pi)$ the set of inversions of π. For any $i \in [n]$ we define the set of *left-inversions* at i in π by $\mathrm{LeftInv}_\pi(i) := \{j \in [n];\, (j, i) \in \mathrm{Inv}(\pi)\}$. The *inversion graph* of a permutation π, denoted G_π is the graph with vertex set $[n]$ and edge set $\mathrm{Inv}(\pi)$.

It will be convenient to represent permutations and their inversion graphs graphically in a $n \times n$ grid. We label columns and rows $1, \cdots, n$ from top to bottom and left to right respectively. The graphical representation of a permutation π consists in placing a dot in each row π_i and column i. The edges of the corresponding inversion graph are then pairs of dots where one is above and to the right of the other. We think of edges (j, i) with $j < i$ as directed from j to i (i.e. from left to right), and refer to them as *arcs*.

Example 2.1. Consider the permutation $\pi = 42315$. The inversions are the pairs of indices $(1, 2)$, $(1, 3)$, $(1, 4)$, $(2, 4)$ and $(3, 4)$. Figure 1 shows the graphical representations of π and of its inversion graph.

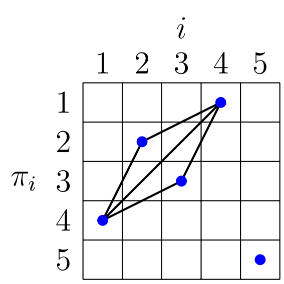

FIGURE 1. The permutation $\pi = 42315$ and its inversion graph G_π.

We will use certain subgraphs of inversion graphs to represent MVP parking functions. Here, subgraphs are considered to be vertex-spanning, so that a subgraph is simply a subset of edges of the original graph. For a permutation π and corresponding inversion graph G_π, we define $\mathrm{Sub}^1(G_\pi) := \{S \subseteq \mathrm{Inv}(\pi)\,;\ \forall i \in [n], |\{j \in [n], (j,i) \in S\}| \leq 1\}$. In words, this is the set of subgraphs of G_π where the number of incident *left-arcs* at any vertex is at most 1. We refer to elements of $\mathrm{Sub}^1(G_\pi)$ as 1-*subgraphs* of G_π. Figure 2 below shows all four 1-subgraphs of G_{312}.

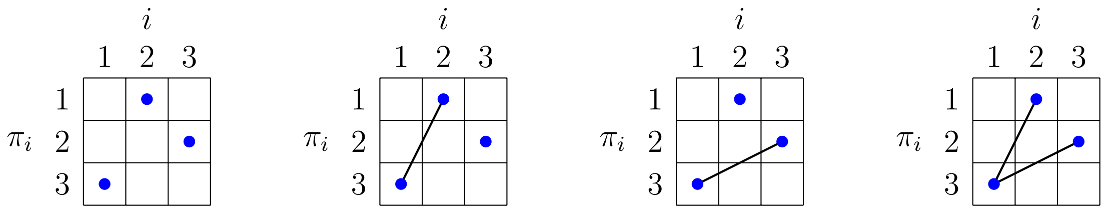

FIGURE 2. The four 1-subgraphs of the inversion graph G_{312}.

2.2. **The MVP outcome map and 1-subgraphs.** In this section, we explain how to represent parking functions in the MVP outcome fibre of a given permutation π via 1-subgraphs of the permutation's inversion graph, and vice versa.

Definition 2.2. Let $\pi \in S_n$ be a permutation. We define a map $\Psi_{\mathrm{PF}\to\mathrm{Sub}} : \mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi) \to \mathrm{Sub}^1(G_\pi)$, $p \mapsto S(p)$ as follows:

$$S(p) := \{(j,i) \in \mathrm{Inv}(\pi)\,;\, p_{\pi_i} = j\}. \tag{1}$$

In words, if the car π_i that ends up in spot i initially preferred some spot $j < i$ in the parking function p (so was eventually bumped to i in the MVP parking process), then we put an edge from j to i in $S(p)$. Note that for this bumping to occur, the car π_j which eventually ends up in spot j must enter the car park after car π_i, which exactly means that (j,i) is an inversion of π. Moreover, since exactly one car ends up in any given spot i, there is at most one left-arc incident to i in $S(p)$ (in the case where $p_{\pi_i} = i$, i.e. the car that ends up in i wanted to park there, we have no incident left-arc), so that $S(p)$ is indeed a 1-subgraph of G_π, as desired. We can then define an inverse for this map from parking functions to sub-graphs, as follows.

Definition 2.3. Let $\pi \in S_n$ be a permutation. We define a map $\Psi_{\mathrm{Sub}\to\mathrm{PF}} : \mathrm{Sub}^1(G_\pi) \to \mathrm{MVP}_n$, $S \mapsto p = p(S)$ as follows:

$$p_{\pi_i} = \begin{cases} i & \text{if} \quad |\{j \in [n]; (j,i) \in S\}| = 0 \\ j & \text{if} \quad j \text{ is the unique } j < i \text{ such that } (j,i) \in S \end{cases}. \tag{2}$$

In words, if there is no left-arc incident to i in S, we set $p_{\pi_i} = i$. Otherwise, since S is a 1-subgraph, there is a unique left-arc (j,i) incident to i in S, and we set $p_{\pi_i} = j$.

Example 2.4. Consider the permutation $\pi = 34125$ and the 1-subgraph $S \in \mathrm{Sub}^1(G_\pi)$ consisting of the arcs $(2,3)$ and $(2,4)$ as in Figure 3. We calculate $p := \Psi_{\mathrm{Sub}\to\mathrm{PF}}(S)$ as follows. First, let us determine p_1 the preference of car 1. Note that $1 = \pi_3$ here, so we are looking at the vertex in row 1, column 3 (labelled 3 in our inversion graph labelling).

Here there is a left-arc incident to this vertex, whose left end-point is in column 2, yielding $p_1 = 2$. Similarly, $p_2 = 2$ also, since there is a left arc incident to the dot in row 2, column 4, whose left-end point is also in column 2. However, the dot in row 3, column 1, has no incident left-arc, and neither does the dot in row 4, column 2, or the dot in row 5, column 5. We therefore set $p_3 = 1$, $p_4 = 2$, and $p_5 = 5$. Finally, we get the preference $p(S) = (p_2, p_2, p_3, p_4, p_5) = (2, 2, 1, 2, 5)$.

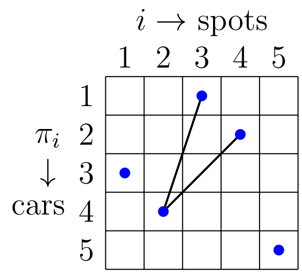

FIGURE 3. A 1-subgraph of G_{34125} whose corresponding parking function is $p := \Psi_{\mathrm{Sub}\to\mathrm{PF}}(S) = (2, 2, 1, 2, 5)$

Our main result of this Section 2 is the following.

Theorem 2.5. *The maps* $\Psi_{\mathrm{PF}\to\mathrm{Sub}} : \mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi) \to \mathrm{Sub}^1(G_\pi)$ *and* $\Psi_{\mathrm{Sub}\to\mathrm{PF}} : \mathrm{Sub}^1(G_\pi) \to \mathrm{MVP}_n$ *are injective, and for any* $\pi \in S_n$ *and* $p \in \mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi)$*, we have* $\Psi_{\mathrm{Sub}\to\mathrm{PF}}(\Psi_{\mathrm{PF}\to\mathrm{Sub}}(p)) = p$.

As such, for a given permutation π, the map $\Psi_{\mathrm{PF}\to\mathrm{Sub}}$ induces a bijection from the fibre $\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi)$ unto its image $\Psi_{\mathrm{PF}\to\mathrm{Sub}}\left(\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi)\right)$. The question of calculating the fibre $\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi)$ then becomes that of calculating the image set, or equivalently calculating the set of 1-subgraphs S of G_π such that $\mathcal{O}_{\mathrm{MVP}_n}(\Psi_{\mathrm{Sub}\to\mathrm{PF}}(S)) = \pi$. Our next results are essentially re-formulations of [9, Theorems 3.1 3.2] in this subgraph context.

Theorem 2.6. *Given any permutation* $\pi \in S_n$*, we have* $\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi) \subseteq \Psi_{\mathrm{Sub}\to\mathrm{PF}}\left(\mathrm{Sub}^1(G_\pi)\right)$*, with equality if and only if* π *avoids the patterns* 321 *and* 3412*, or equivalently if the graph* G_π *is acyclic.*

Corollary 2.7. *For any permutation* $\pi \in S_n$*, we have*

$$\left|\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi)\right| \leq \left|\mathrm{Sub}^1(G_\pi)\right| = \prod_{i\in[n]} \left(1 + |\mathrm{LeftInv}_\pi(i)|\right).$$

Another useful feature of the subgraph representation introduced in this section is that it allows certain statistics of parking functions to be easily read from the corresponding subgraph. Given a parking function p, the *displacement* of car i in p as the number of spots car i ends up from its original preference, i.e. $|p_i - \pi^{-1}_i|$. The displacement of p, denoted $\mathrm{disp}_{\mathrm{MVP}}(p)$, is simply the sum of the displacements of all cars in p. We have the following.

Proposition 2.8. *Let* $p \in \mathrm{MVP}_n$ *be a parking function, and* $S := \Psi_{\mathrm{PF}\to\mathrm{Sub}}(p)$ *the corresponding* 1*-subgraph. Then we have* $\mathrm{disp}_{\mathrm{MVP}}(p) = \sum_{(j,i)\in S} (i - j)$.

2.3. **Improved bounds on the fibre sizes.** In this part, we improve the upper bound from Corollary 2.7, and also give a lower bound for the fibre size. We call a 1-subgraph S $\overrightarrow{P_2}$*-free* if there is no triple $i < j < k$ such that (i, j) and (j, k) are both edges in S.

Proposition 2.9. *Let $\pi \in S_n$ be a permutation, and $S \in \mathrm{Sub}^1(G_\pi)$ a 1-subgraph of G_π. If S is such that $p(S) := \Psi_{\mathrm{Sub}\to\mathrm{PF}}(S) \in \mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi)$, then S is $\overrightarrow{P_2}$-free. In particular, we have $\left|\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi)\right| \leq |\{S \in \mathrm{Sub}^1(G_\pi)\,;\, S \text{ is } \overrightarrow{P_2}\text{-free}\}|$.*

We say that a 1-subgraph $S \in \mathrm{Sub}^1(G_\pi)$ is *horizontally separated* if for any pair of arcs (j, i) and (j', i') of S, we either have $i < j'$ or $i' < j$. In words, there is no pair of arcs in S which "overlap horizontally" in the graphical representation, end-points included.

Proposition 2.10. *Let $\pi \in S_n$ be a permutation, and $S \in \mathrm{Sub}^1(G_\pi)$ a 1-subgraph of G_π. If S is horizontally separated, then we have $\Psi_{\mathrm{Sub}\to\mathrm{PF}}(S) \in \mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi)$. In particular, we have $\left|\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi)\right| \geq |\{S \in \mathrm{Sub}^1(G_\pi)\,;\, S \text{ is horizontally separated}\}|$.*

Note that any subgraph consisting of a single arc is horizontally separated, as is the empty subgraph. As such, the above implies in particular that $\left|\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\pi)\right| \geq 1 + |\mathrm{Inv}(\pi)|$.

3. Motzkin parking functions

3.1. **Motzkin parking functions and Motzkin paths.** We consider lattice paths starting from $(0, 0)$ with steps $U = (1, 1)$ (upwards step), $D = (1, -1)$ (downwards step), and $H = (1, 0)$ (horizontal step). A Motzkin path is a lattice path with these steps ending at some point $(n, 0)$ which never goes below the X-axis (see Figure 4). We denote Motz_n the set of Motzkin paths ending at $(n, 0)$ (i.e. with n steps). Motzkin paths are enumerated by the ubiquitous *Motzkin numbers* and are in bijection with a number of different combinatorial objects (see e.g. [1] or [13]).

Given a parking function $p \in \mathrm{PF}_n$, we define a lattice path with n steps $\Phi(p) := \phi_1 \cdots \phi_n$ by:

$$\forall j \in [n] \quad \phi_j = \begin{cases} U & \text{if} \quad \#\{j; p_i = j\} \geq 2; \\ H & \text{if} \quad \#\{j; p_i = j\} = 1; \\ D & \text{if} \quad \#\{j; p_i = j\} = 0. \end{cases} \tag{3}$$

Definition 3.1. Let $p = (p_1, \cdots, p_n) \in \mathrm{PF}_n$. We say that p is a *Motzkin parking function* if $\forall j \in [n]$, $\#\{j; p_i = j\} \leq 2$. We denote MotzPF_n the set of Motzkin paking functions of length n.

In words, a Motzkin parking function is a parking function in which each spot is preferred by at most two cars. The terminology of Motzkin parking function comes from the following result.

Theorem 3.2. *Let p be a parking preference. Then $p \in \mathrm{MotzPF}_n$ if and only if $\Phi(p)$ is a Motzkin path.*

Example 3.3. Consider the Motzkin parking function $p = (2, 2, 1, 4, 3, 6, 4, 6) \in \mathrm{MotzPF}_8$. The corresponding Motzkin path is $\Phi(p) = HUHUDUDD$. Indeed, spots 1 and 3 are preferred by one car, spots 2, 4 and 6 by two cars, and spots 5, 7 and 8 by no cars. We can check that $\Phi(p)$, illustrated on Figure 4, is indeed a Motzkin path.

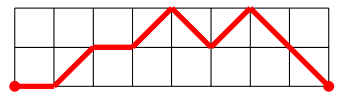

FIGURE 4. The Motzkin path $\Phi(p) = HUHUDUDD$ corresponding to the Motzkin parking function $p = (2, 2, 1, 4, 3, 6, 4, 6)$

The definition of the map Φ in Equation (3) only depends on the number of cars which prefer each spot, and not on the labels of the cars in question. It is therefore natural to define an equivalence relationship on MotzPF_n by $p \sim p'$ if p' is obtained by permuting the preferences in p. For example, the parking functions $(2, 1, 1, 4)$ and $(1, 4, 1, 2)$ are equivalent. We write ${}^{\mathrm{MotzPF}_n}\!/\!_{\sim}$ for the set of equivalence classes of Motzkin parking functions. The above observation implies that Φ is constant on the equivalence classes of $\sim$, so that with slight abuse of notation, we can consider Φ to be defined on the set ${}^{\mathrm{MotzPF}_n}\!/\!_{\sim}$. We then have the following.

Theorem 3.4. *The map* $\Phi : {}^{\mathrm{MotzPF}_n}\!/\!_{\sim} \to \mathrm{Motz}_n$ *is a bijection.*

This theorem can be viewed as a generalisation of a bijection between Motzkin paths and parking functions whose MVP outcome is the *decreasing permutation* $\mathrm{dec}_n := n(n-1)\cdots 1$, which was established by Harris *et al.* [9, Theorem 4.2], as follows.

Theorem 3.5. *For any* $p \in \mathrm{MotzPF}_n$*, there exists a unique* $p' \in \mathrm{MotzPF}_n$ *such that* $p \sim p'$*, and* $\mathcal{O}_{\mathrm{MVP}_n}(p') = \mathrm{dec}_n$*. In particular,* Φ *induces a bijection from the decreasing fibre* $\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\mathrm{dec}_n)$ *to the set* Motz_n *of Motzkin paths of length* n*.*

3.2. **Non-crossing arc diagrams.** Theorem 3.5 implies in particular that the decreasing fibres $\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\mathrm{dec}_n)$ are enumerated by the Motzkin numbers. In this part we give a new bijective explanation of this fact by using our subgraph representation from Section 2. Note that the inversion graph of the decreasing permutation dec_n is just the complete graph K_n on n vertices, since all pairs are inversions in dec_n.

Instead of subgraphs of K_n, we will consider *arc diagrams*. An arc diagram is simply a set of pairs $(j, i) \in [n]^2$ with $j < i$. For $n \geq 1$, we define $\mathrm{ArcDiag}^1_n$ to be the set of arc diagrams of $[n]$ such that for any $i \in [n]$ there is at most one arc (j, i) with $j < i$. We call these 1-*left* arc diagrams.

There is a clear one-to-one correspondence between $\mathrm{ArcDiag}^1_n$ and $\mathrm{Sub}^1(K_n)$, so with slight abuse of notation we identify the two sets. In particular, we consider that the map $\Psi_{\mathrm{Sub}\to\mathrm{PF}}$ from Definition 2.3, mapping a 1-subgraph in $\mathrm{Sub}^1(K_n)$ to a parking function is defined on $\mathrm{ArcDiag}^1_n$. We say that an arc diagram D is a *non-crossing matching* if it satisfies the two following conditions.

(1) **Matching condition**: for every vertex $i \in [n]$, there is at most one arc incident to i in D.
(2) **Non-crossing condition**: no two arcs of D "cross", that is there are no four vertices $i < j < k < \ell$ such that (i, k) and (j, ℓ) are both arcs in D.

We denote $\mathrm{NonCross}_n$ the set of non-crossing arc diagrams on $[n]$. It is well-known that non-crossing arc diagrams are enumerated by Motzkin numbers. Our main result of this section is the following, which gives an alternate proof of the enumerative consequence of Theorem 3.5.

Theorem 3.6. *Let $n \geq 1$. The map $\Psi_{\mathrm{Sub}\to\mathrm{PF}}$ is a bijection from the set of non-crossing matchings $\mathrm{NonCross}_n$ to the decreasing fibre $\mathcal{O}^{-1}_{\mathrm{MVP}_n}(\mathrm{dec}_n)$.*

Example 3.7. Consider the non-crossing matching D in Figure 5 below. We wish to compute the corresponding MVP parking function $p := \Psi_{\mathrm{Sub}\to\mathrm{PF}}(D)$. To get the parking preference p_i of car i, we look at the vertex $n+1-i$ (equivalently, the i-th vertex from the right). If there is a left-arc incident to that vertex, we set p_i to be the label of the left end-point j of that arc. Otherwise we set $p_i = n+1-i$. We get the parking function $p = (11, 7, 8, 8, 7, 1, 3, 4, 3, 2, 1)$. One can check that we do indeed have $\mathcal{O}_{\mathrm{MVP}_{11}}(p) = \mathrm{dec}_{11}$, as desired.

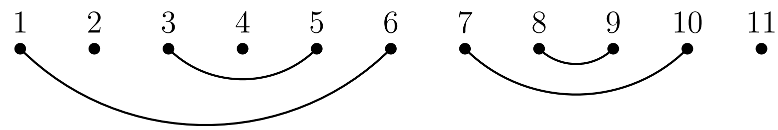

FIGURE 5. An example of a non-crossing matching D on 11 vertices. The corresponding parking function is $p := \Psi_{\mathrm{Sub}\to\mathrm{PF}}(D) = (11, 7, 8, 8, 7, 1, 3, 4, 3, 2, 1)$.

REFERENCES

[1] E. Barcucci, R. Pinzani, and R. Sprugnoli. The Motzkin family. *PU.M.A., Pure Math. Appl., Ser. A*, 2(3-4):249–279, 1991.

[2] Peter Jephson Cameron, Daniel Johannsen, Thomas Prellberg, and Pascal Schweitzer. Counting defective parking functions. *Electron. J. Comb.*, 15(1):research paper R92, 378–388, 2008.

[3] Denis Chebikin and Pavlo Pylyavskyy. A family of bijections between G-parking functions and spanning trees. *J. Comb. Theory, Ser. A*, 110(1):31–41, 2005.

[4] Alex Christensen, Pamela E. Harris, Zakiya Jones, Marissa Loving, Andrés Ramos Rodríguez, Joseph Rennie, and Gordon Rojas Kirby. A generalization of parking functions allowing backward movement. *Electron. J. Comb.*, 27(1):research paper p1.33, 18, 2020.

[5] Laura Colmenarejo, Pamela E. Harris, Zakiya Jones, and Christo Keller. Counting k-naples parking functions through permutations and the k-naples area statistic. *Enumer. Comb. Appl.*, 1(2):Paper No. S2R11, 2021.

[6] Robert Cori and Dominique Poulalhon. Enumeration of (p,q)-parking functions. *Discrete Math.*, 256(3):609–623, 2002. LaCIM 2000 Conference on Combinatorics, Computer Science and Appl ications.

[7] Robert Cori and Dominique Rossin. On the sandpile group of dual graphs. *Eur. J. Comb.*, 21(4):447–459, 2000.

[8] Mark Dukes. The sandpile model on the complete split graph, motzkin words, and tiered parking functions. *J. Comb. Theory, Ser. A*, 180:105418, 2021.

[9] Pamela E. Harris, Brian M. Kamau, J. Carlos Martínez Mori, and Roger Tian. On the outcome map of mvp parking functions: Permutations avoiding 321 and 3412, and motzkin paths. *Enumer. Comb. Appl.*, 3(2):Paper No. S2R11, 16, 2023.

[10] Alan G. Konheim and Benjamin Weiss. An occupancy discipline and applications. *SIAM J. Appl. Math.*, 14(6):1266–1274, 1966.

[11] Alexander Postnikov and Boris Shapiro. Trees, parking functions, syzygies, and deformations of monomial ideals. *T Am Math Soc*, 356:3109–3142, 2004.

[12] Richard P. Stanley. Hyperplane arrangements, interval orders, and trees. *Proc. Natl. Acad. Sci. USA*, 93(6):2620–2625, 1996.

[13] Richard P. Stanley and Sergey Fomin. *Enumerative Combinatorics*, volume 2 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1999.

[14] Richard P. Stanley and Jim Pitman. A polytope related to empirical distributions, plane trees, parking functions, and the associahedron. *Discrete Comput. Geom.*, 27(4):603–634, 2002.

[15] Catherine H. Yan. *Parking Functions*, chapter 13. CRC Press, 2015.

[16] Catalin Zara. Parking functions, stack-sortable permutations, and spaces of paths in the johnson graph. *Electron. J. Comb.*, 9(2):research paper R11, 2003.