On Friendship and Cyclic Parking Functions

Yujia Kang[†], Thomas Selig[‡], Guanyi Yang[†], Yanting Zhang[§], and Haoyue Zhu[‡],

[†]*School of Mathematics and Physics, Xi'an Jiaotong-Livepool University, 111 Ren'ai Road, Suzhou 215123, China*
Email: Yujia.Kang20@student.xjtlu.edu.cn, Guanyi.Yang20@student.xjtlu.edu.cn

[‡]*School of Advanced Technology, Xi'an Jiaotong-Liverpool University, 111 Ren'ai Road, Suzhou 215123, China*
Email: Thomas.Selig@xjtlu.edu.cn, Haoyue.Zhu18@student.xjtlu.edu.cn

[§]*Mathematical Institute, University of Oxford, Woodstock Road, Oxford OX2 6GG, United Kingdom*
Email: Yanting.Zhang@maths.ox.ac.uk

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. In classical parking functions, if a car's preferred spot is already occupied by a previous car, it drives forward and looks for the first unoccupied spot to park.

In this work, we introduce a variant of classical parking functions, called *friendship parking functions*, which imposes additional restrictions on where cars can park. Namely, a car can only end up parking next to cars which are its friends (friendship will correspond to adjacency in an underlying graph). We characterise and enumerate such friendship parking functions according to their outcome permutation, which describes the final configuration when all cars have parked. We apply this to the case where the underlying friendship graph is the cycle graph. Finally, we consider a subset of classical parking functions, called *cyclic parking functions*, where cars end up in an increasing cyclic order. We enumerate these cyclic parking functions and exhibit a bijection to permutation components.

1. Introduction

In this section, we introduce parking functions and friendship parking functions. Throughout the paper, n represents a positive integer, and we denote $[n] := \{1, \ldots, n\}$. Permutations will play a key role in our paper. We denote S_n the set of permutations of the set $[n]$. For $\pi \in S_n$, we write $\pi = \pi_1 \cdots \pi_n$ using standard one-line notation. We also denote π^{-1} the inverse permutation of π. That is, for any $i \in [n]$, π_i^{-1} is the unique $j \in [n]$ such that $\pi_j = i$.

1.1 Parking functions and their variations

A *parking preference* is a vector $p = (p_1, \ldots, p_n) \in [n]^n$. We will regard p_i as the preferred spot of car i in a car park with spots labelled $1, \ldots, n$. Given a parking preference p, the classical parking process for p is described as follows.

Cars enter the car park sequentially in order $1, \ldots, n$, and initially all spots in the car park are unoccupied. When car i enters, it will occupy (park in) the first spot $k \geq p_i$ which is unoccupied at that point. If no such spot exists, car i exits the car park, having failed to park. We say that p is a *classical parking function* if all n cars are able to park through this process. We denote PF_n the set of all classical parking functions with n cars/spots.

The *outcome map* of classical parking functions is the map $\mathcal{O}_{\mathrm{PF}_n} : \mathrm{PF}_n \to S_n$ which associates a permutation $\mathcal{O}_{\mathrm{PF}_n}(p) = \pi = \pi_1 \cdots \pi_n$ of $[n]$ to a classical parking function p as follows. For any $i \in [n]$, we set π_i to be the label of the car ending up in spot i after all cars have parked in the classical parking process for p.

The *displacement vector* of a classical parking function $p \in \mathrm{PF}_n$ is defined by $\mathrm{disp}\,(p) = (d_1, d_2, \ldots, d_n)$, where $d_i \coloneqq \pi_i^{-1} - p_i$ for all $i \in [n]$. Note that for $i \in [n]$, d_i measures the distance that car i has travelled between its preferred spot p_i and the spot π_i^{-1} where it finally parks. We say that d_i is the displacement of car i in the parking function p. The total displacement of p is defined by $|\mathrm{disp}\,(p)| \coloneqq \sum_{i=1}^{n} d_i$.

Example 1.1. *Consider the parking preference $p = (3, 1, 1, 2)$, and let us illustrate the classical parking process for p.*

- *Car 1 has preference $p_1 = 3$. Spot 3 is unoccupied at that point, so car 1 parks in spot 3 (in fact, car 1 will always park in its preferred spot), with displacement $d_1 = 0$.*

- *Car 2 has preference $p_2 = 1$. Spot 1 is unoccupied at that point, so car 2 parks in spot 1, with displacement $d_2 = 0$.*

- *Car 3 has preference $p_3 = 1$. However, spot 1 is occupied at this point (by car 2), so car 3 cannot park there. Instead, it continues to drive forward and finds the first unoccupied spot after its preference, which is spot 2, and so car 3 parks there. The displacement is given by $d_3 = 2 - 1 = 1$.*

- *Lastly, car 4 has preference $p_4 = 2$. Again, spot 2 is occupied (by car 3), as is spot 3 (by car 1). Thus, car 4 will park in spot 4, which is in fact the only unoccupied spot at this point. The displacement is given by $d_4 = 4 - 2 = 2$.*

Finally, the cars are all able to park, so $p = (3, 1, 1, 2)$ is a classical parking function. We also get the outcome $\mathcal{O}_{\mathrm{PF}_n}(p) = 2314$, and the displacement vector $\mathrm{disp}\,(p) = (0, 0, 1, 2)$. The entire classical parking process is illustrated in Figure 1 below.
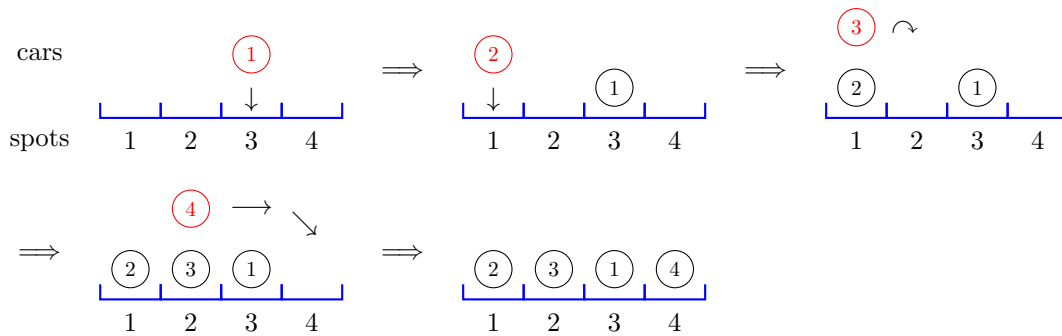

Figure 1: Illustrating the parking process for the classical parking function $p = (3, 1, 1, 2)$.

Parking functions were originally introduced by Konheim and Weiss [16] 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 the Abelian sandpile model [5,11,22,23]. We refer the interested reader to the excellent survey by Yan [26].

We may notice that in classical parking functions, the handling of *collisions* – when a car's preferred spot is already occupied when it enters the car park – appears quite flexible. Indeed, many variations of parking functions with different parking rules have been discussed in the literature. We list some of them briefly here. For a more detailed survey of generalisations and open problems related to parking functions, see [4].

- **Defective parking functions** [3]. This model follows classical parking rules, with m cars and n spots, and k cars being unable to park (k is called the *defect* of the parking function).

- **Naples parking functions** [7,9,25]. This model allows for some backward movement of cars from their preferred spot if it is already occupied.

- **Parking assortments and parking sequences** [1,6,13,14]. These models involve cars of different sizes, with the car park having just enough space for all cars to park.

- **Subset parking functions** [21]. In this model, cars have a preferred ordered subset of spots in which they can park. See also [8] for an in-depth treatment of **interval parking functions**, where each preference subset is a sub-interval of $[n]$, and [2] for recent work on generalisations of these.

- **Vector parking functions**, or **u**-parking functions [20,27]. For a vector $u = (u_1, \ldots, u_n)$, we say that a parking preference $p = (p_1, p_2, \ldots, p_n)$ is a u-parking function if the non-decreasing rearrangement $a = (a_1, \ldots, a_n)$ of p satisfies $a_i \leq u_i$ for all $i \in [n]$. These correspond to classical parking functions when $u = (1, 2, \ldots, n)$ (see e.g. [26, Section 1.1] for this characterisation of classical parking functions).

- **Graphical parking functions** on some graph G, or G-parking functions [17]. In this model, cars park on the vertices of the graph G instead of in a one-way street.

- **Higher-dimensional parking functions**. There are several relevant models in this area, such as the (p, q)-**parking functions** of Cori and Poulalhon [10], or the **tiered parking functions** of Dukes [12]. Both of these models involve cars of different colours or tiers, with extra conditions on where a car can park depending on its tier/colour. Snider and Yan [19,20] then defined a concept of **multidimensional** U-**parking functions**, where U is a collection of multidimensional vectors. These generalise the (p, q)-parking functions of Cori and Poulalhon and the (one-dimensional) vector parking functions, and also connect to the graphical parking functions discussed above.

- **MVP parking functions** [15,18]. In this model, priority is given to cars which enter later in the parking process. That is, when a car enters the car park, if its preferred spot is occupied by a previous car, it will "bump" the earlier car out of that spot, forcing it to drive on and find a new spot to park in.

In this paper, we consider a new variation on parking functions, which we call *friendship parking functions*. Informally, cars follow the classical parking process, but can only park next to cars which they are "friends" with (friendship will be determined by adjacency in an underlying graph).

1.2 Friendship parking functions

In this section, we formally define the concept of friendship parking functions. Let G be a graph with vertex set $[n]$. We say that two vertices i and j are *adjacent* in the graph G if $\{i, j\}$ is an edge of G. As in the classical case, we consider a parking preference $p = (p_1, \ldots, p_n) \in [n]^n$. The *friendship parking process* is described as follows.

Cars enter sequentially in order $1, \ldots, n$ into a car park with spots labelled $1, \ldots, n$. All spots are initially unoccupied. For $i \geq 1$, we say that a spot k is *available* for car i if it is unoccupied when car i enters the car park, and each of the two neighbouring spots $k - 1$ and $k + 1$ are either unoccupied, or occupied by a car j (necessarily $j < i$) which is adjacent to i in the graph G. Here we consider spots 0 and $n + 1$ to be unoccupied by convention when handling the cases $k = 1$ or $k = n$. When car i enters the car park, it occupies (parks in) the first spot $k \geq p_i$ which is available for i. If no such spot exists, the car exits the car park, having failed to park.

We say that p is a *friendship parking function* for G, or G-friendship parking function, if all n cars are able to park following this process. We denote $\mathrm{FPF}(G)$ the set of all G-friendship parking functions. As in the classical case, we can define the outcome map $\mathcal{O}_{\mathrm{FPF}(G)} : \mathrm{FPF}(G) \to S_n$. For a G-friendship parking function p, we let $\pi = \pi_1 \cdots \pi_n = \mathcal{O}_{\mathrm{PF}_G}(p)$ be the permutation such that for any $i \in [n]$, π_i is the label of the car ending up in spot i when running the friendship parking process for p.

Note that if G is the complete graph on n vertices (with edges between any pair of distinct vertices), the notions of classical and friendship parking functions are identical. Indeed, in this case, a spot is available for i if, and only if, it is unoccupied when i enters the car park.

Example 1.2. *Let G be the cycle on $n = 4$ vertices, with edges $\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 1\}$. Consider the parking preference $p = (2, 1, 2, 2)$, and let us illustrate the G-friendship parking process for p.*

- *Car 1 has preference $p_1 = 2$. Spot 2 is available for car 1 since both adjacent spots are both unoccupied (this will always be the case for car 1's preferred spot). Therefore car 1 parks in spot 2.*

- *Car 2 has preference $p_2 = 1$. The neighbouring spot 2 is occupied by car 1 at this point, but vertices 1 and 2 are adjacent in the graph G, so spot 1 is available for car 2, allowing it to park there.*

- *Car 3 has preference $p_3 = 2$. However, spot 2 is occupied at this point (by car 1), so is not available for 3. Spot 3 is unoccupied, but the adjacent spot 2 is occupied by car 1, and 1 and 3 are not adjacent in G. Therefore spot 3 is not available for car 3. Finally, spot 4 is also unoccupied, as is its neighbouring spot 3, which implies that it is available, and thus car 3 parks in spot 4.*

- *Lastly, car 4 has preference $p_4 = 2$. Spot 2 is occupied, so car 4 can't park there. But spot 3 is unoccupied, and the cars occupying its neighbouring spots 2 (car 1) and 4 (car 3) are both adjacent to 4 in G, which means spot 3 is available for car 4, so it parks there.*

Finally, the cars are all able to park, so $p = (2, 1, 2, 2)$ is a G-friendship parking function, and we get the outcome $\mathcal{O}_{\mathrm{FPF}(G)}(p) = 2143$. The entire friendship parking process is illustrated in Figure 2 below.
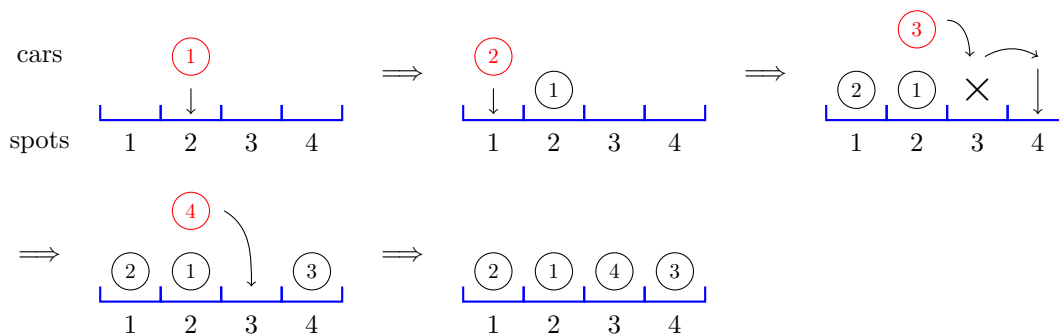
Figure 2: Illustrating the parking process for the friendship parking function $p = (2, 1, 2, 2)$.

Example 1.3. *Let G again be the cycle graph on 4 vertices, as in Example 1.2, and consider the parking preference $p = (4, 2, 2, 1)$. Car 1 parks in spot 4, and car 2 parks in spot 2 (both neighbouring spots are unoccupied). However, car 3 cannot park. Indeed, the preference of car 3 is $p_3 = 2$. At this point, the only unoccupied spot $k \geq 2$ is $k = 3$, but the neighbouring spot 4 is occupied by car 1, which is not adjacent to 3 in G. Therefore spot 3 is not available for car 3, and so there are no available spots on or after car 3's preference for it to park in. Thus p is not a G-friendship parking function. This parking process is illustrated on Figure 3 below.*
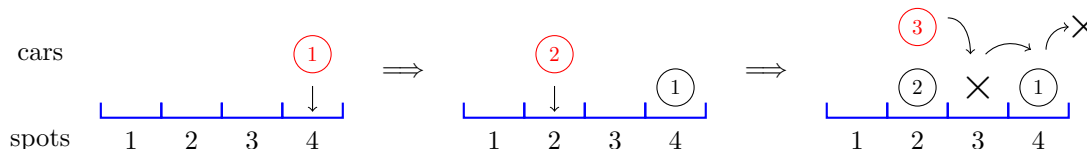

Figure 3: Illustrating the friendship parking process for the parking preference $p = (4, 2, 2, 1)$. Here, car 3 is unable to park, so p is not a friendship parking function.

It is worth noting that in the previous example, $p = (4, 2, 2, 1)$ is a classical parking function (we get the outcome $\mathcal{O}_{\mathrm{PF}_4}(p) = 4231$). This shows that our new notion of friendship parking functions is a meaningful variant of classical parking functions.

1.3 Contents of the paper

Our paper is organised as follows. In Section 2 we study this new friendship parking process. We begin in Section 2.1 by exhibiting a link between G-friendship parking functions and *Hamiltonian paths* of the graph G (Proposition 2.1), and by giving some results connecting friendship and classical parking functions (Propositions 2.2 and 2.3). Section 2.2 then gives a characterisation and enumeration formula for friendship parking functions on general graphs (Theorem 2.1 and Corollary 2.1). In Section 2.3 we apply these results to the case where G is the cycle graph on n vertices, yielding the enumeration formula from Theorem 2.2.

In Section 3 we return to the study of classical parking functions, considering those whose outcome is an increasing cycle of the form $\mathrm{IncCyc}^{i,n} := i(i + 1) \cdots n12 \cdots (i - 1)$ for some $i \in [n]$. We call such parking functions *cyclic parking functions*, and count them (Proposition 3.1 and Corollary 3.1). We then describe a bijection between cyclic parking functions and *permutation components*, which maps the displacement of cars to inversions in the corresponding permutation (Theorem 3.2). Finally, Section 4 gives a brief summary of our results, and outlines some possible future research directions on friendship parking functions.

2. Characterising and counting friendship parking functions

In this section, we give characterisations and enumeration formulas for friendship parking functions on general graphs G. We then apply these to the case where G is the cycle graph on n vertices.

2.1 Friendship parking functions and Hamiltonian paths

We begin by exhibiting a connection between G-friendship parking functions and *Hamiltonian paths* of the graph G. To do this, we first recall some relevant concepts. Given a graph G, with vertex set $[n]$, a *path* of G is a sequence of distinct vertices $\mathcal{P} = (v_1, v_2, \ldots, v_k) \in [n]^k$ such that $\{v_i, v_{i+1}\}$ is an edge of E for each $i \in \{1, \ldots, k-1\}$. Moreover, $\mathcal{P}$ is called a *Hamiltonian* path if $k = n$, i.e. every vertex of G appears in $\mathcal{P}$.

Recall that for a G-friendship parking function $p \in \mathrm{FPF}(G)$, we can define its outcome $\pi = \mathcal{O}_{\mathrm{FPF}(G)}(p) = \pi_1 \cdots \pi_n$, which is a permutation of all the cars (for $i \in [n]$, car π_i is the car that ends up in spot i). In friendship parking functions, the relationship among cars is indicated by a graph G and the permutation reflects that cars (vertices) that can park next to each other must be "friends" (adjacent in the graph G). We can therefore obtain the following fact.

Proposition 2.1. *Let $G = (V, E)$ be a graph. Then the set of G-friendship parking functions is non-empty if, and only if, G contains a Hamiltonian path.*

Proof. Suppose there exists some $p \in \mathrm{FPF}(G)$, and consider its outcome $\mathcal{O}_{\mathrm{FPF}(G)}(p) = \pi_1 \cdots \pi_n$. We claim that $\mathcal{P} := (\pi_1, \ldots, \pi_n)$ is a Hamiltonian path of G. Indeed, consider a car $i = \pi_k$ which parks in some spot $k \leq n-1$ in the friendship parking process. We wish to show that $j := \pi_{k+1}$ is adjacent to i in G. If $j < i$, then car j has already parked in spot $k+1$ when car i arrives. This means that for i to park in spot k, vertices i and j must be adjacent in G (since spot k must be available for car i when it arrives). The same applies if $j > i$: when car j arrives, car i has already parked in spot k, so j can only park in spot $k+1$ if i and j are adjacent in G. Therefore $\mathcal{P}$ is a Hamiltonian path, as desired.

Conversely, suppose that G contains a Hamiltonian path, say $\mathcal{P} = (v_1, v_2, \ldots, v_n) \in [n]^n$ where the v_i are distinct. By construction, $\pi := v_1 \cdots v_n$ is a permutation of $[n]$. Define a parking preference $p = (p_1, \ldots, p_n)$ by $p_i = \pi_i^{-1}$ for all $i \in [n]$. Then it is straightforward to check that $p \in \mathrm{FPF}(G)$ and $\mathcal{O}_{\mathrm{FPF}(G)}(p) = \pi$. Indeed, in the friendship parking process for p, each car can simply park in its preferred spot (since $\mathcal{P}$ is a Hamiltonian path, these are always available). $\qquad\square$

Proposition 2.1 establishes a connection between the outcomes of G-friendship parking functions and Hamiltonian paths of G. With a slight abuse of notation, we will henceforth consider a Hamiltonian path of G to be a permutation $\pi = \pi_1 \cdots \pi_n$ of $[n]$ such that $\{\pi_i, \pi_{i+1}\}$ is an edge of G for all $i \in [n-1]$. We denote $\mathcal{H}(G)$ the set of Hamiltonian paths of G. We end this section by stating two fairly straightforward propositions that connect classical and friendship parking functions.

Proposition 2.2. *We have $\mathrm{FPF}(G) \subseteq \mathrm{PF}_n$.*

Proof. We first recall a well-known characterisation of classical parking functions. For a parking preference $p = (p_1, \ldots, p_n)$, we denote $\tilde{p} = (\tilde{p}_1, \ldots, \tilde{p}_n)$ the *non-decreasing rearrangement* of p. Then $p \in \mathrm{PF}_n$ if, and only if, we have $\tilde{p}_i \leq i$ for all $i \in [n]$ (see e.g. [26, Section 1.1]).

Now let $p \in \mathrm{FPF}(G)$ be a G-friendship parking function, and denote $\tilde{p}$ its non-decreasing rearrangement. Seeking contradiction, suppose there is some $i \in [n]$ such that $\tilde{p}_i > i$. By construction, this means that $|\{j \in [n]; p_j \leq i\}| \leq i - 1$. But in the friendship (or indeed classical) parking process, each car necessarily ends up parking on or after its preferred spot. In particular, for any $i \in [n]$, there must be at least i cars whose preference is one of the first i spots (since exactly i cars must end up in these spots). This yields the desired contradiction with the above inequality. $\qquad\square$

Proposition 2.3. *Let $p \in \mathrm{PF}_n$ be a parking function such that its outcome $\mathcal{O}_{\mathrm{PF}_n}(p)$ is a Hamiltonian path of the graph G. Then we have $p \in \mathrm{FPF}(G)$, and $\mathcal{O}_{\mathrm{FPF}(G)}(p) = \mathcal{O}_{\mathrm{PF}_n}(p)$.*

Proof. In fact, in this case, the friendship and classical parking processes are identical. Indeed, when a car i parks in some spot k in the classical parking process, it will never move from spot k thereafter. As such, since $\pi := \mathcal{O}_{\mathrm{PF}_n}(p) \in \mathcal{H}(G)$, this implies that the spot k is available at that time for car i in the friendship parking process. Since by definition k is the first value greater than or equal to p_i which is unoccupied when i enters the car park, car i also parks in k for the friendship parking process. This completes the proof. $\qquad\square$

Remark 2.1. *Propositions 2.2 and 2.3 are not quite converse to each other. Indeed, there exist friendship parking functions (which are thus also classical parking functions) whose classical outcome is not a Hamiltonian path. Example 1.2 gives an example of such a parking function given by $p = (2, 1, 2, 2)$. This is a friendship parking function for the 4-cycle, but its classical outcome is $\mathcal{O}_{\mathrm{PF}_4}(p) = 2134$, which is not a Hamiltonian path of the 4-cycle (vertices 1 and 3 are not adjacent). As such, friendship parking functions are not just classical parking functions whose outcome is a Hamiltonian path.*

2.2 Characterising friendship parking functions according to their outcomes

In this section, we characterise and enumerate friendship parking on general graphs according to their outcome permutation. For $\pi \in S_n$, the *outcome fibre* of π is the set $\mathcal{O}_{\mathrm{FPF}(G)}^{-1}(\pi) := \{p \in \mathrm{FPF}(G) ; \mathcal{O}_{\mathrm{FPF}(G)}(p) = \pi\}$ of G-friendship parking functions whose (friendship) outcome is π. We begin by giving the definitions of *blocker* and *blocking sequence*, which will be used in the subsequent results.

Definition 2.1. *Let $\pi \in \mathcal{H}(G)$ be a Hamiltonian path of the graph G, and $i \in [n]$. We say that $j = \pi_k$ is a* blocker *for i in π, or equivalently that i is blocked by j in π, if either of the two following conditions holds:*

1. *We have $j \leq i$; or*

2. *We have $j > i$, but there is a neighbour ℓ of j in π, i.e. $\ell \in \{\pi_{k-1}, \pi_{k+1}\}$ such that $\ell < i$ and ℓ is not adjacent to i in G.*

The blocking sequence *for i in π is the longest sub-sequence of π, ending at i, whose elements are all blockers for i. We denote $B(i, \pi, G)$ the blocking sequence for i, and $b(i, \pi, G) := |B(i, \pi, G)|$ its length.*

Example 2.1. *Let G be the graph in Figure 4, with Hamiltonian path $\pi = 87152463$ (drawn in red). Then $i = 4$ is blocked by: $1, 2, 3, 4$ (all ≤ 4), as well as 5 and 7. For 5 and 7, note that they are both neighbours of $\ell = 1$ in π, which satisfies $\ell = 1 < 4 = i$, and 1 is not adjacent to 4 in G, so Condition (2) of Definition 2.1 applies.*

However, $j = 8$ is not a blocker, since its only neighbour $\ell = 7$ in π is not strictly less than $i = 4$. Neither is $j = 6$ a blocker. Indeed, its neighbours in π are 3 and 4. Among these, the only value which is strictly less than $i = 4$ is $\ell = 3$, and there is an edge between $\ell = 3$ and $i = 4$ in G, so Condition (2) of Definition 2.1 does not apply here. The blocking sequence for 4 is then $B(4, \pi, G) = 71524$, with length $b(4, \pi, G) = 5$.
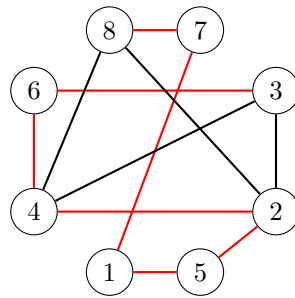

Figure 4: The graph G with Hamiltonian path $\pi = 87152463$ (red)

We now state the main result of this section, which gives a complete characterisation of the outcome fibres of friendship parking functions in terms of blocking sequences.

Theorem 2.1. *Let $\pi \in \mathcal{H}(G)$ be a Hamiltonian path of G, and $p = (p_1, \ldots, p_n) \in [n]^n$ a parking preference. Then we have $p \in \mathcal{O}_{\mathrm{FPF}(G)}^{-1}(\pi)$ if, and only if, for all $i \in [n]$, we have $p_i \in S_i$, where $S_i := \{s \in [n] ; \pi_s \in B(i, \pi, G)\}$ is the set of spots that are occupied in the outcome π by an element in the blocking sequence for i in π.*

Example 2.2. *Consider the graph G from Figure 4 with corresponding Hamiltonian path $\pi = 87152463$ as in Example 2.1. We wish to calculate the fibre set $\mathcal{O}_{\mathrm{FPF}(G)}^{-1}(\pi)$ by applying Theorem 2.1. For this we calculate the blocking sequences $B(i, \pi, G)$ for each element $i \in [8]$, and their corresponding index sets $S_i := \{s \in [n] ; \pi_s \in B(i, \pi, G)\}$.*

- *For $i = 1$, $B(1, \pi, G) = 1 = \pi_3$, so $S_1 = \{3\}$.*

- *For $i = 2$, $B(2, \pi, G) = 7152 = \pi_2\pi_3\pi_4\pi_5$, so $S_2 = \{2, 3, 4, 5\}$.*

- *For $i = 3$, $B(3, \pi, G) = 3 = \pi_8$, so $S_3 = \{8\}$.*

- *For $i = 4$, $B(4, \pi, G) = 71524 = \pi_2 \cdots \pi_6$, so $S_4 = [2, 6]$.*

- *For $i = 5$, $B(5, \pi, G) = 15 = \pi_3\pi_4$, so $S_5 = \{3, 4\}$.*

- *For $i = 6$, $B(6, \pi, G) = 715246 = \pi_2 \cdots \pi_7$, so $S_6 = [2, 7]$.*

- *For $i = 7$, $B(7, \pi, G) = 7 = \pi_2$, so $S_7 = \{2\}$.*

- *For $i = 8$, $B(8, \pi, G) = 8 = \pi_1$, so $S_8 = \{1\}$.*

Finally, we get the fibre set $\mathcal{O}_{\mathrm{FPF}(G)}^{-1}(\pi) = \prod_{i=1}^{8} S_i = \{3\} \times [2,5] \times \cdots \times \{2\} \times \{1\}$, *and the fibre size* $\left|\mathcal{O}_{\mathrm{FPF}(G)}^{-1}(\pi)\right| = \prod_{i=1}^{8} |S_i| = 1 \cdot 4 \cdot 1 \cdot 5 \cdot 2 \cdot 6 \cdot 1 \cdot 1 = 240$.

Theorem 2.1 immediately implies enumeration formulas for the fibres and total number of friendship parking functions.

Corollary 2.1. *For any Hamiltonian path* $\pi \in \mathcal{H}(G)$ *of* G, *we have* $\left|\mathcal{O}_{\mathrm{FPF}(G)}^{-1}(\pi)\right| = \prod_{i \in [n]} b(i, \pi, G)$. *In particular, this implies that the total number of* G*-friendship parking functions is equal to* $\sum_{\pi \in \mathcal{H}(G)} \left(\prod_{i \in [n]} b(i, \pi, G) \right)$.

Proof of Theorem 2.1. Fix $\pi \in \mathcal{H}(G)$ a Hamiltonian path of G. For $i \in [n]$, denote $S_i := \{s \in [n]; \pi_s \in B(i, \pi, G)\}$ the set of spots occupied in π by cars in the blocking sequence of i, and $s_i := \min S_i$ the first such spot. Note that by definition of the blocking sequence, we have $S_i = [s_i, \pi_i^{-1}]$.

We first assume that $p = (p_1, \ldots, p_n) \in \mathcal{O}_{\mathrm{FPF}(G)}^{-1}(\pi)$ is a friendship parking function whose outcome is the Hamiltonian path π. We wish to show that for all $i \in [n]$, we have $p_i \in S_i$. Seeking contradiction, suppose that is not the case. Let $i \in [n]$ be such that $p_i \notin S_i$, and define $j := \pi_i^{-1}$ to be the spot where car i ends up. Since $S_i = [s_i, \pi_i^{-1}] = [s_i, j]$, we have two cases to consider.

1. Case 1: $p_i > j$. That is, car i prefers a spot (strictly) to the right of spot j. But by definition, we have $j = \pi_i$, so car i should end up in spot j, which is impossible given the parking process (cars necessarily end up parking on or to the right of their preference).

2. Case 2: $p_i < s_i$. Consider the spot $k := s_i - 1$, and the corresponding car π_k. Note that we have $p_i \leq k < j$. By definition of the blocking sequence, π_k is not a blocker for i in π. In particular, we must have $\pi_k > i$, so spot k is unoccupied when car i arrives. Moreover, none of the neighbouring spots $k-1, k+1$ can be occupied by cars arriving before i which are not adjacent to i in G. This means exactly that spot k is available for i in the G-friendship parking process for p. But since $p_i \leq k$, this means that car i will end up either in spot k or to its left. In particular, it is impossible for car i to end up in spot j, since $j > k$. This gives the desired contradiction.

We now show the converse. That is, suppose that $p = (p_1, \ldots, p_n)$ is a parking preference such that $p_i \in S_i$ for all $i \in [n]$. We wish to show that $p \in \mathcal{O}_{\mathrm{FPF}(G)}^{-1}(\pi)$. Consider the friendship parking process for p. We show by induction on $i \geq 1$ that car i always ends up in spot π_i^{-1}.

For $i = 1$, we note that 1 has no blockers other than itself, since it is the minimal element of $[n]$, and the condition for $j > i$ to be a blocker (Condition (2) of Definition 2.1) relies on the existence of an element ℓ satisfying $\ell < i$. That is, $B(1, \pi, G) = 1$, so $S_1 = \{\pi_1^{-1}\}$ by definition. Therefore, since $p_1 \in S_1$ by assumption, we have $p_1 = \pi_1^{-1}$. Finally, since car 1 always parks in its preferred spot, we have that car 1 ends up in spot π_1^{-1}, as desired.

Now suppose that, for some $i \geq 2$, we have shown that car j parks in spot π_j^{-1} for all $j \leq i - 1$. Consider how car i parks. By assumption, we have $p_i \in S_i = [s_i, \pi_i^{-1}]$. Consider a spot $k \in S_i \setminus \{\pi_i^{-1}\}$, and $j := \pi_k$ the corresponding car. We wish to show that i cannot park in spot k. By definition, j is a blocker for i, and $j \neq i$ (since $k = \pi_j^{-1} < \pi_i^{-1}$). We distinguish two cases.

1. Case 1: $j < i$. By the induction hypothesis, j occupies spot $\pi_j^{-1} = k$, so car i cannot park in spot k.

2. Case 2: $j > i$. In this case, by definition of blockers, for at least one of the values $\ell \in \{k-1, k+1\}$, we must have that $\pi_\ell < i$ and π_ℓ is not adjacent to i in G. By the induction hypothesis, car π_ℓ occupies spot ℓ, which neighbours spot k in π, and since π_ℓ and i are not adjacent in G, this prevents car i from parking in spot k in the friendship parking process.

As such, car i cannot park in any of the spots $s_i, s_i + 1, \ldots, \pi_i^{-1} - 1$. It therefore remains to show that it is allowed to park in spot π_i^{-1} in the friendship parking process, i.e. that spot π_i^{-1} is available for i. First, note that by the induction hypothesis, cars $1, \ldots, i-1$ occupy spots $\pi_1^{-1}, \ldots, \pi_{i-1}^{-1}$. Therefore, when car i enters the car park, spot π_i^{-1} is unoccupied. Now consider $N(i) := \{\pi_i^{-1} - 1, \pi_i^{-1} + 1\}$ the two neighbouring spots to π_i^{-1}. By definition, if $s \in N(i)$, then π_s is a neighbour of i in the Hamiltonian path π, and therefore is adjacent to i in the graph G. Moreover, by the induction hypothesis, when i enters the car park, such a spot s is either occupied by π_s (if $\pi_s < i$), or unoccupied (if $\pi_s > i$). Finally, we have shown that when car i enters the car park, spot π_i^{-1} is unoccupied (see above), and each of its two neighbouring spots is either unoccupied, or is occupied by a car adjacent to i in G. This means exactly that spot π_i^{-1} is available for i in the friendship parking process, as desired. $\square$

2.3   Application: the cycle graph case

In this section, we are interested in the specific case where G is a cycle graph. For $n \geq 3$, we denote the cycle graph on n vertices by C_n. This is the graph with vertex set $[n]$ and edge set $\{(i, i+1)\}_{i \in [n-1]} \cup \{(n, 1)\}$ (see Figure 5).
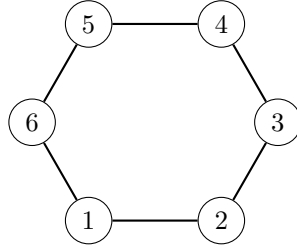

Figure 5: The cycle graph C_6 on six vertices.

We wish to apply Theorem 2.1 and Corollary 2.1 to the cycle graph case. First, note that Hamiltonian paths of the cycle graph are of one of the following two types:

1. *Increasing cycles*, which are of the form $\mathrm{IncCyc}^{i,n} := i(i+1) \cdots n 1 2 \cdots (i-1)$ for some $i \in [n]$;

2. *Decreasing cycles*, which are of the form $\mathrm{DecCyc}^{i,n} := i(i-1) \cdots 1 n(n-1) \cdots (i+1)$ for some $i \in [n]$.

We characterise the blocking sequences for both types of outcome. We begin with the decreasing cycle case. For technical reasons (see Remark 2.2) we assume that $n \geq 4$.

Proposition 2.4. *Decreasing cycle case, $n \geq 4$.*

1. **Case** $i \leq n-2$. *Let $i \leq n-2$. For any $j \in [n]$, we have:*

$$B\left(j, \mathrm{DecCyc}^{i,n}, C_n\right) = \begin{cases} j & \textit{if } j \leq n-2, \\ i(i-1) \cdots 1 n(n-1) & \textit{if } j = n-1, \\ i(i-1) \cdots 1 n & \textit{if } j = n. \end{cases}$$

   In particular, this implies that $\left| \mathcal{O}_{\mathrm{FPF}(C_n)}^{-1} \left(\mathrm{DecCyc}^{i,n} \right) \right| = (i+1)(i+2)$.

2. **Case** $i = n-1$. *For any $j \in [n]$, we have:*

$$B\left(j, \mathrm{DecCyc}^{n-1,n}, C_n\right) = \begin{cases} j & \textit{if } j \leq n-1, \\ (n-1) \cdots 1 n = \mathrm{DecCyc}^{n-1,n} & \textit{if } j = n. \end{cases}$$

   In particular, this implies that $\left| \mathcal{O}_{\mathrm{FPF}(C_n)}^{-1} \left(\mathrm{DecCyc}^{n-1,n} \right) \right| = n$.

3. **Case** $i = n$. *For any $j \in [n]$, we have $B\left(j, \mathrm{DecCyc}^{n,n}, C_n\right) = j$. In particular, this implies that $\left| \mathcal{O}_{\mathrm{FPF}(C_n)}^{-1} \left(\mathrm{DecCyc}^{n,n} \right) \right| = 1$.*

Proof. Fix some $i, j \in [n]$, and consider the blocking sequence for j in the decreasing cycle $\mathrm{DecCyc}^{i,n} = i(i-1) \cdots 1 n(n-1) \cdots (i+1)$. We first note that if $j \leq n-2$, then $\mathrm{DecCyc}^{i,n} = \cdots (j+2)(j+1)j \cdots$ (assuming j doesn't appear in position 1 or 2). This implies that $j+1$ is not a blocker for j, since it is strictly greater than j and both of its neighbours in $\mathrm{DecCyc}^{i,n}$ are greater than or equal to j. This shows that $B\left(j, \mathrm{DecCyc}^{i,n}, C_n\right) = j$, as desired.

  Now consider $j = n-1$. If $i \leq n-2$, then we have $\mathrm{DecCyc}^{i,n} = i \cdots 1 n(n-1) \cdots$. As the vertices 1 and $n-1$ are not adjacent in C_n (since $n \geq 4$), and $1 < n-1$, this implies that n is a blocker for $n-1$. Moreover, all values $i, i-1, \ldots, 1$ are strictly less than $n-1$, so are all blockers. Thus, we have $B\left(n-1, \mathrm{DecCyc}^{i,n}, C_n\right) = i(i-1) \cdots 1 n(n-1)$, as desired. However, this relies on 1 (immediately) preceding n in $\mathrm{DecCyc}^{i,n}$, which only occurs if $i \leq n-2$. Otherwise, $n-1$ appears in first or second position, and is at most preceded by n, which is not a blocker in that case, so $B\left(n-1, \mathrm{DecCyc}^{i,n}, C_n\right) = n-1$ if $i \in \{n-1, n\}$, as desired.

  Finally, consider $j = n$. If $i \leq n-1$, then we have $\mathrm{DecCyc}^{i,n} = i \cdots 1 n \cdots$. All elements in $i, i-1, \ldots, 1$ are strictly less than n, so are blockers in this case. This gives $B\left(n, \mathrm{DecCyc}^{i,n}, C_n\right) = i(i-1) \cdots 1 n$ as desired. If $i = n$, then n appears in first position in $\mathrm{DecCyc}^{n,n} = n(n-1) \cdots 1$, so $B\left(n, \mathrm{DecCyc}^{i,n}, C_n\right) = n$. This completes the proof. □

Remark 2.2. *The above proof in the case $j = n - 1$ shows why we assumed that $n \geq 4$. Otherwise, for $n = 3$, in the outcome 132, the blocking sequence for $j = n - 1 = 2$ is just 2. Indeed, in this case, 3 is not a blocker for 2, since 1 and 2 are adjacent in C_3. We therefore get $\left|\mathcal{O}_{\mathrm{FPF}(C_3)}^{-1}(132)\right| = 1 * 1 * b(3, 132, C_3) = 2$, instead of the value $2 * 3 = 6$ as would be given by Case (1) of Proposition 2.4.*

Proposition 2.5. *Increasing cycle case.*

1. ***Case*** $i \geq 4$. *Let $i \geq 4$. For any $j \in [n]$, we have:*

$$B\left(j, \mathrm{IncCyc}^{i,n}, C_n\right) = \begin{cases} i(i+1)\cdots j & \text{if } j \geq i, \\ n12\cdots j & \text{if } 3 \leq j \leq i - 1, \\ 12 & \text{if } j = 2, \\ 1 & \text{if } j = 1. \end{cases}$$

   In particular, this implies that $\left|\mathcal{O}_{\mathrm{FPF}(C_n)}^{-1}\left(\mathrm{IncCyc}^{i,n}\right)\right| = (n - i + 1)! i! / 3$.

2. ***Case*** $i \leq 3$. *Let $i \leq 3$. For any $j \in [n]$, we have:*

$$B\left(j, \mathrm{IncCyc}^{i,n}, C_n\right) = \begin{cases} i\cdots j & \text{if } j \geq i, \\ 1\cdots j & \text{if } j \leq i - 1. \end{cases}$$

   In particular, this implies that $\left|\mathcal{O}_{\mathrm{FPF}(C_n)}^{-1}\left(\mathrm{IncCyc}^{i,n}\right)\right| = (n + 1 - i)!(i - 1)!$.

Proof. Fix $i, j \in [n]$. If $j \geq i$, we have $B\left(j, \mathrm{IncCyc}^{i,n}, C_n\right) = i(i+1)\cdots j$. Indeed, all elements to the left of j in $\mathrm{IncCyc}^{i,n}$ are less than or equal to j, so they are all blockers for j. Moreover, it is straightforward to check that $B\left(1, \mathrm{IncCyc}^{i,n}, C_n\right) = 1$ (true in fact for any graph G and Hamiltonian path $\pi \in \mathcal{H}(G)$, see proof of Theorem 2.1).

Now consider the case $j = 2$ where $j \leq i - 1$, i.e. 2 is not in first position in $\mathrm{IncCyc}^{i,n}$. In this case, 2 immediately follows 1 in $\mathrm{IncCyc}^{i,n}$, and therefore 1 is in the blocking sequence for 2. Moreover, n is not a blocker for 2, since its neighbours in $\mathrm{IncCyc}^{i,n}$ are $n - 1$ (which is greater than or equal to 2), and 1 (which is adjacent to 2 in C_n). Therefore if $i \geq 3$ we have $B\left(2, \mathrm{IncCyc}^{i,n}, C_n\right) = 12 = 1\cdots j$, as desired.

It therefore remains to consider the case $3 \leq j \leq i - 1$. Necessarily we have $i \geq 4$, so that $\mathrm{IncCyc}^{i,n} = \cdots n12\cdots j\cdots$. By definition, $1, 2, \ldots, j$ are all blockers for j, since they are less than or equal to j. Moreover, since $3 \leq j \leq i - 1 \leq n - 1$, the elements j and 1 are not adjacent in C_n (n and 2 are the vertices adjacent to 1). Since n is followed by 1 in $\mathrm{IncCyc}^{i,n}$, this implies that n is a blocker for j. Finally, $n - 1$ is not a blocker for j, since both its neighbours in $\mathrm{IncCyc}^{i,n}$ are greater than or equal to j. Therefore, we get $B\left(2, \mathrm{IncCyc}^{i,n}, C_n\right) = n12\cdots j$, as desired. This completes the proof. $\square$

Combining Propositions 2.4 and 2.5, we get the following enumeration for the total number of C_n-friendship parking functions.

Theorem 2.2. *Let $n \geq 4$. The total number of friendship parking functions for the cycle graph C_n is given by*

$$\left|\mathrm{FPF}(C_n)\right| = n + 1 + \sum_{i=1}^{n-2}(i+1)(i+2) + \sum_{i=1}^{3}(n+1-i)!(i-1)! + \sum_{i=4}^{n}(n-i+1)!i!/3.$$

For the case $n = 3$, the enumeration is given by the same formula, except that we replace the product $(i+1)(i+2)$ in the left-most sum with just $(i+1)$, as per Remark 2.2. Calculating these numbers for the first few values of $n \geq 3$, we get the sequence $16, 65, 256, 1331, 8710, 67135, \ldots$, which doesn't appear in the OEIS [24].

3. Cyclic parking functions and permutation components

In this part, we consider classical parking functions whose outcome is an increasing cycle, i.e. of the form $\mathrm{IncCyc}^{i,n} := i(i+1)\cdots n12\cdots(i-1)$ for some $i \in [n]$. This is a slight variant on the C_n-friendship parking functions from Section 2.3. We say that $p = (p_1, \ldots, p_n)$ is a *cyclic parking function* if we have $\mathcal{O}_{\mathrm{PF}_n}(p) = \mathrm{IncCyc}^{i,n}$, for some $i \in [n]$. We denote $\mathrm{CycPF}_n := \bigcup_{i \in [n]} \mathcal{O}_{\mathrm{PF}_n}^{-1}\left(\mathrm{IncCyc}^{i,n}\right)$ the set of cyclic parking functions of length n.

Proposition 3.1. *For any $i \in [n]$, we have $\left|\mathcal{O}_{\mathrm{PF}_n}^{-1}\left(\mathrm{IncCyc}^{i,n}\right)\right| = (n + 1 - i)!(i - 1)!$.*

Proof. The proof is similar to that of Theorem 2.1 and Proposition 2.5, so we allow ourselves to be briefer here. Let $i \in [n]$, and consider how a parking preference $p = (p_1, \ldots, p_n)$ can yield the outcome permutation $\mathrm{IncCyc}^{i,n} = i(i+1) \cdots n 12 \cdots (i-1)$ (for the classical parking process). Car 1 ends up in spot $j := n + 2 - i$ (with the convention that $j = 1$ if $i = 1$). Since car 1 always ends up in its preferred spot in the classical parking process, we must have $p_1 = j$. Then car 2 ends up in spot $j + 1$. This occurs if, and only if, we have $p_2 \in \{j, j+1\}$ (since car 1 occupies j).

We repeat this reasoning on the first $i - 1$ cars, and see that each car $k \in \{1, \ldots, i-1\}$ has exactly k possibilities for its parking preference. This gives the $(i-1)!$ factor in the formula. The $(n+1-i)!$ factor comes from an analogous reasoning on cars $i, i+1, \ldots, n$. Car i ends up in spot 1, so must prefer that spot. Car $(i+1)$ ends up in spot 2, so must prefer spot 1 (occupied by car i) or spot 2, and so on. This completes the proof. $\qquad\square$

Remark 3.1. *The formula in Proposition 3.1 is quite similar to that of friendship parking functions on the cycle graph C_n whose outcome is an increasing cycle (see Proposition 2.5). The key difference lies in the role of the car/vertex n. In C_n-friendship parking functions, this vertex plays the role of a blocker for cars $j \geq 3$ which park to its right, whereas in classical parking functions it is never a "blocker" (if car j wanted to park in the corresponding spot, it could).*

As an example, consider the parking function $p = (2, 3, 1, 1)$. In the friendship parking process for C_4, car 3 cannot park in spot 1, since spot 2 is occupied by car 1, and there is no edge $(1, 3)$ in C_4. Therefore we get the outcome $\mathcal{O}_{\mathrm{FPF}(4)}(p) = 4123 = \mathrm{IncCyc}^{4,4}$, and p is a C_4-friendship parking function. However, in the classical parking process, car 3 can park in spot 1, and so car 4 will end up in spot 4, yielding the outcome $\mathcal{O}_{\mathrm{PF}_4}(p) = 3124$, so p is not a cyclic parking function in the sense of this section.

Summing the right-hand side of the enumeration in Proposition 3.1 over values of i from 1 to n, combined with the change of variable $i \to n + 1 - i$, immediately yields the following.

Corollary 3.1. *We have* $|\mathrm{CycPF}_n| = \sum_{i=0}^{n-1} i!(n-i)!$.

Calculating these numbers for the first few values of n, we get the sequence $1, 3, 10, 40, 192, \ldots$, which corresponds to Sequence A136128 in the OEIS [24]. The entry for this sequence states "Number of components in all permutations of $[n]$", and the goal of this section is to give a bijective proof of this fact by exhibiting a bijection between cyclic parking functions and permutation components. As such, cyclic parking functions provide a new combinatorial interpretation for this Sequence A136128. Before defining our bijection, we first need to introduce some terminology relating to permutations.

Given a permutation $\pi = \pi_1 \cdots \pi_n$, a *component* of π is a sub-sequence $\pi_i \pi_{i+1} \cdots \pi_j$ for some $1 \leq i \leq j \leq n$ such that $\{\pi_i, \pi_{i+1}, \ldots, \pi_j\} = \{i, i+1, \ldots, j\}$, which is minimal for inclusion (i.e. it does not contain any smaller such sub-sequence). It is straightforward to see that every permutation can be uniquely decomposed into its components. For example, the permutation 31248657 has three components: 312, 4, 8657. We will use a slash to separate the components of a permutation, for example $312/4/8657$. The set of components of a permutation π is denoted $\mathrm{comp}(\pi)$.

Note that the same sub-sequence can be a component of more than one permutation. For example, 1 is a component of both $1/32$ or $1/2/3$. In this section, we consider the set of all components of permutations of length n, $\mathrm{Comp}_n := \bigsqcup_{\pi \in S_n} \mathrm{comp}(\pi)$, where $\bigsqcup$ denotes the disjoint union. That is, we distinguish between the component 1 in the permutation $1/32$ and the component 1 in the permutation $1/2/3$. For $c \in \mathrm{Comp}_n$, we will denote $\mathrm{Perm}(c) \in S_n$ the underlying permutation, i.e. $c \in \mathrm{comp}(\mathrm{Perm}(c))$.

In our bijection between cyclic parking functions and permutation components, permutation inversions will play a key role. Given a permutation $\pi = \pi_1 \cdots \pi_n \in S_n$, we say that an ordered pair (i, j) is an *inversion* of π if $i > j$ and i appears before j in the permutation π. The *inversion number* of i in π, denoted $\mathrm{inv}_i(\pi)$, is the number of $j < i$ such that (i, j) is an inversion of π. In words, the inversion number of i counts how many letters less than i appear after i in π. For example, the permutation $\pi = 3152476$ has the following inversions: $(3, 1), (3, 2), (5, 2), (5, 4), (7, 6)$. We have $\mathrm{inv}_1(\pi) = \mathrm{inv}_2(\pi) = \mathrm{inv}_4(\pi) = \mathrm{inv}_6(\pi) = 0$, $\mathrm{inv}_3(\pi) = \mathrm{inv}_5(\pi) = 2$, and $\mathrm{inv}_7(\pi) = 1$. Note that any inversion of π is necessarily contained in a component of π.

Definition 3.1. *Let $a = (a_1, \ldots, a_n)$ be a sequence of non-negative integers. We say that a is an* inversion sequence *if $a_i < i$ for all $i \in [n]$.*

The terminology *inversion sequence* comes from the following classical result, of which we give a brief proof for completeness.

Theorem 3.1. *The map* $\mathrm{InvSeq} : \pi = \pi_1 \cdots \pi_n \mapsto \mathrm{InvSeq}(\pi) := (\mathrm{inv}_1(\pi), \ldots, \mathrm{inv}_n(\pi))$ *is a bijection from the set of n-permutations to the set of inversion sequences of length n.*

Proof. Let $\pi = \pi_1 \cdots \pi_n \in S_n$, and $\mathrm{InvSeq}(\pi) := (\mathrm{inv}_1(\pi), \ldots, \mathrm{inv}_n(\pi))$. By definition, for any $i \in [n]$, $\mathrm{inv}_i(\pi)$ is the number of $j < i$ such that (i, j) is an inversion of π, which immediately implies that $\mathrm{inv}_i(\pi) < i$, i.e. that $\mathrm{InvSeq}(\pi)$ is an inversion sequence. Moreover, it is straightforward to see that there are $n!$ inversion sequences of length n. Indeed, we have 1 choice (0) for the entry a_1, 2 choices (0 and 1) for the entry a_2, and so on.

To show that the map InvSeq is a bijection, it is therefore sufficient to show that it is surjective. Let $a = (a_1, \ldots, a_n)$ be an inversion sequence. We construct a permutation π as follows. Start with $i = 0$ and the empty word $\pi^0 = \emptyset$. Suppose that after i steps we have constructed a word $\pi^i = \pi_1^i \cdots \pi_i^i \in S_i$ such that $\mathrm{inv}_j(\pi^i) = a_j$ for all $j \in [i]$. Then at step $i + 1$, we define $\pi^{i+1} = \pi_1^{i+1} \cdots \pi_{i+1}^{i+1}$ by simply inserting $(i + 1)$ in the permutation π^i at index $i + 1 - a_{i+1}$ (i.e. a_{i+1} spots starting from the right). By construction, we get a permutation $\pi^{i+1} \in S_{i+1}$ such that $\mathrm{inv}_j(\pi^{i+1}) = a_j$ for all $j \in [i+1]$. Therefore, the final permutation $\pi^n \in S_n$ obtained satisfies $\mathrm{InvSeq}(\pi^n) = a$, as desired. $\square$

Example 3.1. *Consider the inversion sequence $a = (0, 0, 1, 3, 2)$. We start with the empty word $\pi^0 = \emptyset$. At step 1, we insert the letter 1 in the empty word so that there are $a_1 = 0$ letters to its right, i.e. we get $\pi^1 = 1$ (we will always have this step). At step 2, we have $a_2 = 0$, so we insert 2 into the permutation $\pi^1 = 1$, so that there are 0 letters to its right, yielding $\pi^2 = 12$. Then at step 3, we have $a_3 = 1$. As such, we insert 3 into the permutation 12 so that it has 1 letter to its right, yielding $\pi^3 = 132$. At step 4, we insert 4 into 132 so that it has $a_4 = 3$ letters to its right, yielding $\pi^4 = 4132$. Finally, at step 5, we insert 5 so that it has $a_5 = 2$ letters to its right, yielding the permutation $\pi = \pi^5 = 41532$. We can then check that $\mathrm{InvSeq}(\pi) = (0, 0, 1, 3, 2) = a$, as desired.*

Inversion sequences are linked to the displacement vector of parking functions by the following result.

Proposition 3.2. *Let $p \in \mathrm{PF}_n$ be a parking function. Then its displacement vector $\mathrm{disp}(p)$ is an inversion sequence.*

Proof. Let $p \in \mathrm{PF}_n$, and write $\mathrm{disp}(p) = (d_1, \ldots, d_n)$ for its displacement vector. Fix $i \in [n]$. By definition, d_i is the distance car i travels from its preferred spot to its final parking spot. In the classical parking process, this means that spots $p_i, p_i + 1, \ldots, p_i + d_i - 1$ must all be occupied when car i enters the car park. Since there are $i - 1$ cars entering before car i, and each of these occupies at most one of those spots, this implies that $d_i \leq i - 1 < i$, as desired. $\square$

We are now equipped to define the bijection from cyclic parking functions to permutation components. Given a cyclic parking function p, whose outcome is $\mathrm{IncCyc}^{i,n}$ for some $i \in [n]$, we define $\Psi(p)$ to be the component of π containing i, where $\pi \in S_n$ is the unique permutation such that $\mathrm{InvSeq}(\pi) = \mathrm{disp}(p)$.

Example 3.2. *Consider the parking function $p = (4, 4, 6, 6, 7, 9, 7, 1, 2, 1)$. By running the parking process, we get $\mathcal{O}_{\mathrm{PF}_{10}}(p) = 89(10)1234567 = \mathrm{IncCyc}^{8,10}$. The displacement vector of p is $\mathrm{disp}(p) = (0, 1, 0, 1, 1, 0, 3, 0, 0, 2)$. The corresponding permutation π such that $\mathrm{disp}(p) = \mathrm{InvSeq}(\pi)$ is $\pi = 21/47536/(10)89$, and we map this to the component containing car $i = 8$, so $\Psi(p) = (10)89$.*

Example 3.3. *Here we fix $n = 3$, and consider the images of all cyclic parking functions of length n under the map Ψ. These are illustrated in Table 1 below. We have partitioned cyclic parking functions (listed in Column 2) according to their outcome (in Column 1). Column 3 then gives the displacement vector of each parking function, with Column 4 listing the permutation whose inversion sequence is the displacement vector. Finally, in the component column (Column 5), we have copied in this permutation for clarity, with the corresponding component $\Psi(p)$ in **bold**.*

Outcome	Parking function p	$\mathrm{disp}(p)$	Permutation π	Component $c = \Psi(p)$
123	$(1, 1, 1)$	$(0, 1, 2)$	321	**321**
	$(1, 1, 2)$	$(0, 1, 1)$	231	**231**
	$(1, 1, 3)$	$(0, 1, 0)$	213	**21**/3
	$(1, 2, 1)$	$(0, 0, 2)$	312	**312**
	$(1, 2, 2)$	$(0, 0, 1)$	132	**1**/32
	$(1, 2, 3)$	$(0, 0, 0)$	123	**1**/2/3
231	$(3, 1, 1)$	$(0, 0, 1)$	132	1/**32**
	$(3, 1, 2)$	$(0, 0, 0)$	123	1/**2**/3
312	$(2, 2, 1)$	$(0, 1, 0)$	213	21/**3**
	$(2, 3, 1)$	$(0, 0, 0)$	123	1/2/**3**

Table 1: All cyclic parking functions of length $n = 3$, with their corresponding permutation components.

Theorem 3.2. *The map $\Psi : \mathrm{CycPF}_n \to \mathrm{Comp}_n$ is a bijection from cyclic parking functions to permutation components. Moreover, for any cyclic parking function p and any $i \in [n]$, the displacement of car i in p is equal to the inversion number of i in the underlying permutation $\mathrm{Perm}\,(\Psi(p))$.*

Lemma 3.1. *Let $p \in \mathrm{CycPF}_n$ be a cyclic parking function with outcome $\mathcal{O}_{\mathrm{FPF}(C_n)}(p) = \mathrm{IncCyc}^{i,n}$, i.e. car i ends up in spot 1. Then we have $i = \min \Psi(p)$.*

Proof. Let $p \in \mathrm{CycPF}_n$ be a cyclic parking function, and $\mathrm{disp}\,(p) = (d_1, \ldots, d_n)$ its displacement vector. Define $\pi \in S_n$ to be the unique permutation such that $\mathrm{InvSeq}\,(\pi) = \mathrm{disp}\,(p)$. Denote $c := \Psi(p)$ the component of the permutation π corresponding to the parking function p, and let $i' := \min c$ and $k = |c|$ be its minimal element and size respectively. By definition, the component c is a permutation of the set $\{i', \ldots, i' + k - 1\}$. Moreover, by construction we have $d_j = \mathrm{inv}_j\,(\pi) = \mathrm{inv}_j\,(c)$ for any $j \in c$, since all inversions of the permutation π are necessarily contained in its components.

Seeking contradiction, suppose that $i' < i$. We consider cars $i', i'+1, \ldots, i-1$ in the parking process, which are the $i - i'$ smallest elements of the component c. By the remarks above, we have $d_{i'+j-1} < j$ for all $j \in [i - i']$. Now consider cars $i, i+1, \ldots, i' + k - 1$. By definition of i, these occupy spots $1, \ldots, k + i' - i$. In particular, we must have $\mathrm{disp}\,(i + j - 1) < j$ for all $j \in [k + i' - i]$. These inequalities can be summarised as below.

cars:	i'	$\cdots$	$i-1$	i	$\cdots$	$i'+k-1$
	$d_{i'}$	$\cdots$	d_{i-1}	d_i	$\cdots$	$d_{i'+k-1}$
displacement:	$\wedge$	$\cdots$	$\wedge$	$\wedge$	$\cdots$	$\wedge$
	1	$\cdots$	$i-i'$	1	$\cdots$	$k+i'-i$

Now recall that $\mathrm{disp}\,(p) = \mathrm{InvSeq}\,(\pi)$, and that c is a sub-sequence in π which is a permutation of the set $\{i' \ldots, i' + k - 1\}$. It is straightforward, using the inverse construction from Theorem 3.1, to check that the inequalities above imply that c can be written as $c = c^1 \cdot c^2$, where c^1, resp. c^2, is a permutation of the set $\{i', \ldots, i-1\}$, resp. of the set $\{i, \ldots, i'+k-1\}$. This contradicts the minimality of the component c. $\qquad\square$

Proof of Theorem 3.2. We first show that Ψ is injective. For this, let p, p' be two cyclic parking functions such that $c := \Psi(p) = \Psi(p')$. Let $i := \min c$. By Lemma 3.1, we have that $\mathcal{O}_{\mathrm{FPF}(n)}(p) = \mathcal{O}_{\mathrm{FPF}(n)}(p') = \mathrm{IncCyc}^{i,n}$. Moreover, by construction, we have $\mathrm{disp}\,(p) = \mathrm{disp}\,(p') = \mathrm{Perm}\,(c)$. We claim that two parking functions which share the same outcome and displacement vector are necessarily equal. Indeed, the preference of any car is equal to its index in the outcome permutation minus its displacement (this is the definition of the displacement vector).

It remains to show that Ψ is surjective, which we do by constructing its inverse. Let $c \in \mathrm{comp}\,(\pi)$ be a component of some permutation $\pi \in S_n$. Define $i := \min c$ to be its minimal element. Let $s = (s_1, \ldots, s_n) := \mathrm{InvSeq}\,(\pi)$ be the inversion sequence of π. For $j \in [n]$, denote $\bar{j}$ the index of j in the cyclic permutation $\mathrm{IncCyc}^{i,n}$. That is, $\bar{j} = n + j + 1 - i$ if $j \le i - 1$, and $\bar{j} = j + 1 - i$ if $j \ge i$.

We claim that $s_j < \bar{j}$ for all $j \in [n]$. If $j \le i - 1$, this simply follows from the fact that s is an inversion sequence: we have $\bar{j} \ge j + 1$ and $s_j < j$. Now note that since i is the minimal element in a component of the permutation π, this means in particular that π can be decomposed as $\pi^{<i} \cdot \pi^{\ge i}$, where $\pi^{<i}$, resp. $\pi^{\ge i}$, is a permutation of the set $\{1, \ldots, i-1\}$, resp. $\{i, \ldots, n\}$. In particular, if $j \ge i$, all inversions (j, j') of π must be in the right-hand permutation $\pi^{\ge i}$. This implies in particular that $s_j < j - (i - 1) = \bar{j}$, as desired.

We can then define $p = (p_1, \ldots, p_n)$ by setting $p_j = \bar{j} - s_j$ for all $j \in [n]$. By the above claim, we have $p_j > 0$ for all j, so p is a parking preference. Moreover, by construction, we have $\mathcal{O}_{\mathrm{FPF}(n)}(p) = \mathrm{IncCyc}^{i,n}$, and $\Psi(p) = c$. This shows that Ψ is surjective, and thus a bijection, as desired. The fact that Ψ maps the displacement of a car to its inversion number in the underlying permutation is by construction: the permutation $\mathrm{Perm}\,(\Psi(p))$ is defined so that $\mathrm{InvSeq}\,(\mathrm{Perm}\,(\Psi(\mathrm{p}))) = \mathrm{disp}\,(p)$. $\qquad\square$

Example 3.4. *We show an example of the construction of Ψ^{-1} in the previous proof. Consider the permutation $\pi = 3412/\mathbf{7865}/9$, with component $c = 7865$. We first calculate the inversion sequence of π, yielding $\mathrm{InvSeq}\,(\pi) = (0, 0, 2, 2; 0, 1, 2, 2; 0)$ where we have used a semi-colon to separate the permutation components for clarity. The minimal element of c is $i = 5$. This indicates that car 5 must end up in spot 1, cars 6 to 9 end up in spots 2 to 5, and cars 1 to 4 end up in spots 6 to 9. For each car i, we determine its parking preference p_i by subtracting its inversion number (which will be equal to its displacement) from its final spot. For example, car 8 ends up in spot 4, and 8 has inversion number 2 in π, corresponding to the inversions $(8, 6)$ and $(8, 5)$, so we set $p_8 = 4 - 2 = 2$. We do this for each car, and finally get the parking preference $p = (6, 7, 6, 7, 1, 1, 1, 2, 5)$. We can check that $\mathcal{O}_{\mathrm{FPF}(9)}(p) = 567891234$, and that $\mathrm{disp}\,(p) = \mathrm{InvSeq}\,(\pi)$, as desired.*

Remark 3.2. *Theorem 3.2 implies a converse for Proposition 3.2. Namely, if $a = (a_1, \ldots, a_n)$ is an inversion sequence, then there exists a cyclic parking function p such that $\mathrm{disp}\,(p) = a$. In fact, the number of such cyclic parking functions is equal to the number of components of the permutation π such that $a = \mathrm{InvSeq}\,(\pi)$. Indeed, given an inversion sequence a with corresponding permutation π (i.e. $\mathrm{InvSeq}\,(\pi) = a$), then the set of cyclic parking functions p satisfying $\mathrm{disp}\,(p) = a$ is exactly the set $\Psi^{-1}\,(\mathrm{comp}\,(\pi))$.*

4. Conclusion and future work

In this paper, we introduced a new variation on classical parking functions, called *friendship parking functions*. These essentially follow classical parking rules, with the added restriction that a car can only park next to its "friends" (adjacent vertices in an underlying graph G). We gave a full characterisation of G-friendship parking functions according to their outcome, or equivalently Hamiltonian paths of G (Theorem 2.1). We then applied this result and its enumerative consequence to the case where G is the cycle graph on n vertices (Section 2.3). Finally, in Section 3 we returned to classical parking functions, considering *cyclic parking functions* whose (classical) outcome is an increasing cycle of the form $i(i + 1) \cdots n12 \cdots (i - 1)$. We exhibited (Theorem 3.2) a bijection between cyclic parking functions of *permutation components*, which maps the *displacement* of each car to its *inversion* number in the corresponding permutation.

One natural question from this last work is to ask whether other families of permutations have natural combinatorial interpretations of their outcome fibres (whether for classical parking functions or one of the variations thereon). For example, in [18] it was shown that the outcome fibre of a permutation π can be interpreted in terms of certain subgraphs of the corresponding permutation graph G_π, and various families of permutations were studied under that representation. It would be interesting to ask similar questions of classical parking functions.

Regarding friendship parking functions, a first question to consider would be to pursue the study of graph families. In this paper we studied the case of cycle graphs. We could also investigate other families such as cycle graphs with different labelling, wheel graphs, complete graphs with one edge removed, and so on. We should look for both closed enumeration formulas and combinatorial interpretations of friendship parking functions in those cases.

Finally, we could also consider the case where the underlying graph G does not have a Hamiltonian path. In this case, there will be no friendship parking functions allowing all cars to park, but we could consider *defective* parking functions, as in the classical case [3]. In other words, for some fixed $k > 0$, how many friendship parking functions are there where exactly k cars fail to park? What are the minimal/maximal values of such a k? Of particular interest could be the case where G is a disjoint union of cliques, i.e. where the friendship relationship is transitive. In other words, we partition the set $[n]$ into subsets $F_1, \ldots, F_j$ where any pair of elements in F_ℓ are friends, but there is no friendship between elements of distinct F_ℓ and $F_{\ell'}$. What can be said about friendship parking functions under this model?

Acknowledgments

References

[1] A. Adeniran and C. Yan, *On increasing and invariant parking sequences*, Australas. J. Comb. 79 (2021), 167–182.

[2] T. Aguilar-Fraga, J. Elder, R. E. Garcia, K. P. Hadaway, P. E. Harris, K. J. Harry, I. B. Hogan, J. Johnson, J. Kretschmann, K. Lawson-Chavanu, J. C. M. Mori, C. D. Monroe, D. Quiñonez, D. Tolston III, and D. A. Williams II, *Interval and ℓ-interval rational parking functions*, ArXiv preprint (2023), arXiv:2311.14055.

[3] P. Cameron, D. Johannsen, T. Prellberg, and P. Schweitzer, *Counting defective parking functions*, Electron. J. Comb. 15(1) (2008), 378–388.

[4] J. Carlson, A. Christensen, P. E. Harris, Z. Jones, and A. Ramos Rodríguez, *Parking functions: choose your own adventure*, Coll. Math. J. 52(4) (2021), 254–264.

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

[6] D. Chen, P. E. Harris, J. C. M. Mori, E. Pabón-Cancel, and G. Sargent, *Permutation invariant parking assortments*, Enumer. Comb. Appl. 4(1) (2021), Article #S2R4.

[7] A. Christensen, P. E. Harris, Z. Jones, M. Loving, A. Ramos Rodríguez, J. Rennie, and G. Rojas Kirby, *A generalization of parking functions allowing backward movement*, Electron. J. Comb. 27(1) (2020), #P1.33.

[8] E. Colaric, R. DeMuse, J. L. Martin, and M. Yin, *Interval parking functions*, Adv. Appl. Math. 123 (2021), #P102129.

[9] L. Colmenarejo, P. E. Harris, Z. Jones, and C. Keller, *Counting k-Naples parking functions through permutations and the k-Naples area statistic*, Enumer. Comb. Appl. 1(2) (2021), Article #S2R11.

[10] R. Cori and D. Poulalhon, *Enumeration of (p,q)-parking functions*, Discrete Math. 256(3) (2002), 609–623.

[11] R. Cori and D. Rossin, *On the sandpile group of dual graphs*, Eur. J. Comb. 21(4) (2000), 447–459.

[12] M. Dukes, *The sandpile model on the complete split graph, Motzkin words, and tiered parking functions*, J. Comb. Theory, Ser. A 180 (2021), #P105418, 2021.

[13] R. Ehrenborg and A. Happ, *Parking cars of different sizes*, Am. Math. Mon. 123(10) (2016), 1045–1048.

[14] S. Franks, P. E. Harris, K. Harry, J. Kretschmann, and M. Vance, *Counting parking sequences and parking assortments through permutations*, Enumer. Comb. Appl. 4(1) (2021), Article #S2R2.

[15] P. E. Harris, B. M. Kamau, J. C. M. Mori, and R. Tian, *On the outcome map of MVP parking functions: permutations avoiding 321 and 3412, and Motzkin paths*, Enumer. Comb. Appl. 3(2) (2023), Article #S2R11.

[16] A. G. Konheim and B. Weiss, *An occupancy discipline and applications*, SIAM J. Appl. Math. 14(6) (1966), 1266–1274.

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

[18] T. Selig and H. Zhu, *New combinatorial perspectives on MVP parking functions and their outcome map*, ArXiv preprint (2023), arXiv:2309.11788.

[19] L. Snider and C. Yan, *U-parking functions and (p,q)-parking functions*, Adv. Appl. Math. 134 (2022), #P102309.

[20] L. Snider and C. Yan, *($\mathfrak{S}_p \times \mathfrak{S}_q$)-invariant graphical parking functions*, ArXiv preprint (2023), arXiv:2305.03651.

[21] S. Spiro, *Subset parking functions*, J. Integer Seq. 22(7) (2019), Article #19.7.3.

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

[23] 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), 603–634.

[24] N. J. A. Sloane et al., *The On-Line Encyclopedia of Integer Sequences*, 2024. Available at `https://oeis.org`.

[25] R. Tian, *Connecting k-Naples parking functions and obstructed parking functions via involutions*, Electron. J. Comb. 29(3) (2022), #P3.8.

[26] C. H. Yan, *Parking Functions*, Chapter 13, CRC Press, 2015.

[27] M. Yin, *Parking functions, multi-shuffle, and asymptotic phenomena*, Matematica 2(2) (2023), 258–282.