

Counting k -Naples Parking Functions Through Permutations and the k -Naples Area Statistic

Laura Colmenarejo¹, Pamela E. Harris², Zakiya Jones³, Christo Keller⁴,
Andrés Ramos Rodríguez⁵, Eunice Sukarto⁶ and Andrés R. Vindas-Meléndez⁷

¹*Department of Mathematics and Statistics, UMass Amherst, United States
Email: laura.colmenarejo.hernando@gmail.com*

²*Department of Mathematics and Statistics, Williams College, United States
Email: peh2@williams.edu*

³*Department of Mathematics, Pomona College, United States
Email: zakiyacmjones@gmail.com*

⁴*Department of Mathematics and Statistics, UMass Amherst, United States
Email: thechristokeller@gmail.com*

⁵*Department of Mathematics, Universidad de Puerto Rico, Río Piedras, United States
Email: ramosandres443@gmail.com*

⁶*Department of Mathematics, University of California, Berkeley, United States
Email: eunicesukarto@berkeley.edu*

⁷*Department of Mathematics, University of Kentucky, United States
Email: andres.vindas@uky.edu*

Received: October 15, 2020 **Accepted:** January 22, 2021, **Published:** February 19, 2021
The authors: Released under the CC BY-ND license (International 4.0)

ABSTRACT: The k -Naples parking functions of length n (a generalization of parking functions) are defined by requiring that a car which finds its preferred spot occupied must first back up a spot at a time (up to k spots) before proceeding down the street. Note that the well-studied parking functions are the specialization of k to 0. For fixed $0 \leq k \leq n - 1$, we define a function φ_k which maps a k -Naples parking function to the permutation denoting the order in which its cars park. Summing the sizes of the fibers of φ_k gives a new formula for the number of k -Naples parking functions as a sum over the permutations of length n .

We note that our formula for enumerating k -Naples parking functions is not recursive, in contrast to the previously known formula of Christensen et al [10]. It can be expressed as the product of the lengths of particular subsequences of permutations, and its specialization to $k = 0$ gives another way to describe the number $PF_n = (n + 1)^{n-1}$ of parking functions of length n . We give a formula for the sizes of the fibers of the map φ_0 , and provide a recurrence relation for its logarithmic generating function. Furthermore, we relate the q -analog of our formula to a new statistic that we denote \mathbf{area}_k and call the k -Naples area statistic, the specialization of which to $k = 0$ gives the usual \mathbf{area} statistic on the parking functions.

Keywords: k -Naples parking functions; permutations; q -analogs; generating functions; area statistic

2020 Mathematics Subject Classification: 05A15; 05A19

1. Introduction

Parking functions are combinatorial objects introduced in [11] to study hashing problems. To define these objects, consider a one-way street with n parking spots labeled 1 through n and a line of n cars, c_1, c_2, \dots, c_n waiting to park. A parking preference is a tuple $\alpha = (\alpha_1, \dots, \alpha_n) \in [n]^n$, where α_i is the preferred parking spot of car c_i . The rule for parking cars is that car c_i goes to its preferred spot α_i and if it is empty, parks there. If

the space is occupied, c_i moves forward until it finds the next available space. We say that a parking preference α is a parking function if all the cars can park within the given n spots without cars driving off the road.

Parking functions have received much attention in the combinatorics literature [7]. The study of parking functions has led to connections with fields such as graph theory, representation theory, hyperplane arrangements, and discrete geometry [6, 12, 13, 19]. Moreover, in the last few years many generalizations of parking functions have appeared in the literature [1–4]. For a survey of generalizations and open problems related to parking functions, see [8]. Among those generalizations, we find the *k-Naples parking functions*, introduced by Baumgardner for $k = 1$ and by Christensen et. al. for general k , see [5, 10]. This generalization modifies the parking rule so that car c_i , upon finding its preferred space α_i occupied, first backs up and checks if the spot $\alpha_i - 1$ is occupied. If it is empty it parks there, otherwise it backs up, a spot at a time, at most k spaces attempting to park in the first available space before going forward once again. The set of *k-Naples parking functions* is the set of parking functions that can park under the *k-Naples rule*. In [10], the authors study this generalization and provide a recursive formula for the number of *k-Naples parking functions* and provide some connections to signature Dyck paths.

This paper continues the study of the *k-Naples parking functions* by extending it to a new q -analog. To make our results concrete we begin by describing our process. Let S_n denote the symmetric group on n letters, $PF_{n,k}$ denote the set of *k-Naples parking functions* of length n , and $PF_{n,0} := PF_n$ denote the set of parking functions of length n . For a fixed $0 \leq k \leq n - 1$, we define a map φ_k which maps a *k-Naples parking function* to the permutation denoting the order in which the cars park (see Definition 3.3). Then we count the size of the fibers of this map under each permutation in S_n (see Theorem 3.1), thereby giving a formula for the number of *k-Naples parking functions* as a sum over S_n (see Theorem 3.2). This formula enumerating *k-Naples parking functions* is new and not recursive, in contrast to the one in [10]. Moreover, our formula is given as the product of the lengths of particular subsequences of permutations. In the parking function case, that is when $k = 0$, we also give a recurrence relation for the fibers of the map φ_0 , providing a recursive formula to compute the coefficients of the corresponding generating function. We continue our work by studying a q -statistic of *k-Naples parking functions* of length n . Given a non-negative integer n , we denote its q -analog

$$[n]_q = \frac{1 - q^n}{1 - q} = 1 + q + q^2 + \dots + q^{n-1}.$$

With this q -analog, one may define q -analogs of almost any formula. For instance, the q -factorial or the q -binomial coefficients. In general, we denote by $[\star]_q$ the q -analog of the formula \star by substituting any number appearing in \star by its q -analog. It turns out that the q -analog of our formula for counting parking functions coincides with the *fermionic formula* presented in [12], taking $t = 0$. We describe the relation between the distribution of the *area* statistic and the q -analog of our formula (see Proposition 5.1). We also define the *k-Naples area statistic*, and give the analogous result for the *k-Naples parking functions*.

The paper is organized as follows. In Section 2, we include main definitions and results related to parking functions and *k-Naples parking functions*. In Section 3, we present our main result counting *k-Naples parking functions* through permutations, and we include the result for parking functions to illustrate the intuitive idea of the proof. Then in Section 4 we give a recursive formula to enumerate permutations whose preimage under φ_0 has a given cardinality and present the associated generating function. Finally, in Section 5, we define the *area_k* statistic for *k-Naples parking functions* and relate our enumerative results to it.

2. Background

Let us start by defining the main objects of our study, as well as the notation we use.

Definition 2.1. Let $\alpha = \alpha_1, \dots, \alpha_n$, with $\alpha_i \in [n] := \{1, \dots, n\}$. We say that α is a parking preference sequence meaning that the i^{th} car, henceforth denoted c_i , wants to park in spot α_i . Then, α is a parking function of length n if all the cars park under the following parking rule:

Imagine n cars travel down a one-way street with n parking spots. Each car prefers a spot, which it attempts to park in. If the spot is empty, it parks there and succeeds; otherwise, it continues down the road until it finds an empty spot to park in.

We denote by PP_n the set of parking preferences of length n , and by PF_n the set of parking functions of length n . It is worth noting that the parking functions are the sequences $p = (p_1, p_2, \dots, p_n)$ of positive integers for which the non-decreasing rearrangement (q_1, q_2, \dots, q_n) with $q_1 \leq q_2 \leq \dots \leq q_n$ satisfies $q_i \leq i$.

Example 2.1. Consider the parking preference $(2, 1, 1)$. We have three open spots and c_1 takes the 2nd spot. In comes c_2 and parks in spot 1. Now c_3 drives up to the 1st spot and sees that the parking spot is occupied. Following the parking rule, it proceeds to the 2nd spot, which is also taken. Finally it tries the 3rd spot, where it parks successfully since the other cars have taken other spots. All three cars park, so $(2, 1, 1) \in PF_3$.

The k -Naples parking functions generalize these objects by modifying the parking rule.

Definition 2.2. We say that a parking preference $\alpha = \alpha_1, \dots, \alpha_n$ is a k -Naples parking function, with $0 \leq k \leq n$, if all the cars are able to park under the following k -Naples parking rule:

Imagine n cars travel down a one-way street with n parking spots. Each car prefers a spot, which it attempts to park in. If the spot is empty, it backs up checking up to k spots behind its preferred spot and parks in the first available. If all the k spots preceding its preferred spot are occupied, then the car continues down the street until it finds an empty spot in which to park.

We denote by $PF_{n,k}$ the set of k -Naples parking functions of length n . Moreover, we denote by $PF_{n,k}^d$ the subset of $PF_{n,k}$ given by the decreasing k -Naples parking functions, i.e., those with $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n$.

Note that the case when $k = 0$ is precisely the case where cars do not back up, which is exactly the definition of a parking function. Hence $PF_{n,0} = PF_n$. Also, if the cars are able to back up and check up to $n - 1$ spots behind their preferred spot, this would allow cars to check the entire length of the street in search for an empty parking spot. Hence, every car can park regardless of their parking preference, which implies that $PF_{n,n-1} = PP_n$. Moreover, whenever $1 \leq k \leq n$ it follows that the containment $PF_{n,k-1} \subseteq PF_{n,k}$ holds. But it is not the case that every k -Naples parking function is a parking function.

Example 2.2. Take the parking preference $(3, 2, 2)$. Let us check how the cars behave trying to park under the parking rule. The first two cars park, but c_3 drives off the road. Following the 1-Naples parking rule, the first two cars park (as before) and c_3 finding the 2nd parking spot occupied, it subsequently checks the $(2 - 1)^{th} = 1^{st}$ spot, where it is able to park. Therefore $(3, 2, 2) \in PF_{3,1}$ even though $(3, 2, 2) \notin PF_{3,0} = PF_3$.

Parking functions are in bijection with many other combinatorial objects. For instance, rooted forests [16], maximal chains of non-crossing partitions [20], and regions in the Shi arrangement [19]. For our purposes, we present their bijection to labeled Dyck paths then a similar result for the k -Naples case.

Definition 2.3. A Dyck path of length $2n$ is a lattice path from $(0, n)$ to $(n, 0)$ consisting of n steps east by $(1, 0)$ and n steps south by $(0, -1)$ all of which have the path staying above the diagonal $y = n - x$. The number of Dyck paths is the n^{th} Catalan number C_n . A labeled Dyck path of length $2n$ is a Dyck path such that the south steps are labeled with numbers $1, \dots, n$ and consecutive south steps have increasing labels.

Theorem 2.1. The parking functions of length n are in bijection with the labeled Dyck paths of length $2n$.

Given a parking function $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \in PF_n$, let u_i denote the number of occurrences of i in α . Consider an $n \times n$ grid, and label the vertical lines from left to right with $1, \dots, n$. Notice the last vertical line on the right is not labeled. Then we define the Dyck path P by drawing u_i south steps in the i^{th} column and we label each south step with the position of the value i in α , making sure that the labels are increasing for consecutive south steps. See [12] for details. We illustrate the bijection of Theorem 2.1 with an example.

Example 2.3. Consider the parking function $\alpha = (3, 3, 1, 4, 2, 2)$. Then, u_1 with label $\{3\}$, $u_2 = 2$ with labels $\{5, 6\}$, $u_3 = 2$ with labels $\{1, 2\}$, $u_4 = 1$ with label $\{4\}$, and $u_5 = u_6 = 0$. See Figure 1 for this Dyck path.

There is no known generalization of this result to all the k -Naples parking functions. However, using k -lattice paths of length $2n$, we have a partial characterization for weakly decreasing k -Naples parking functions.

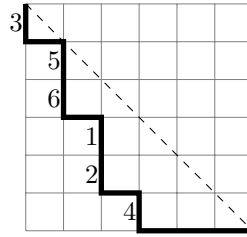


Figure 1: Labeled Dyck path corresponding to the parking function $\alpha = (3, 3, 1, 4, 2, 2)$.

Definition 2.4. Given $n, k \in \mathbb{N}$ with $0 \leq k \leq n - 1$, a k -lattice path of length $2n$ is a lattice path from $(0, n)$ to $(n, 0)$ consisting of n steps east and n steps south, such that the path never goes above the line $y = n - x + k$.

Theorem 2.2 (Theorem 1.3, [10]). Given $n, k \in \mathbb{N}$ with $k \leq n - 1$, the set of decreasing k -Naples parking functions of length n is in bijection with the k -lattice paths of length $2n$ where the first step is south.

Remark 2.1. The definition of k -lattice paths does not require that the first step be south. However, it is a necessary condition for the bijection, as the authors of [10] point out in the proof.

In the most general case, parking preferences are in bijection with lattice paths. This bijection sends a parking preference, $\alpha = (\alpha_1, \dots, \alpha_n) \in PP_n$, to the lattice path with east steps $(i - 1, \alpha_i - 1)$ to $(i, \alpha_i - 1)$, for $i = 1, \dots, n$. In [10, Theorem 1.3], the authors show that this bijection restricts nicely to k -Naples parking functions, establishing Theorem 2.2. We consider the following example to illustrate this bijection.

Example 2.4. We draw the 2-lattice path corresponding to the 2-Naples parking function $\alpha = (6, 6, 4, 4, 2, 2)$.

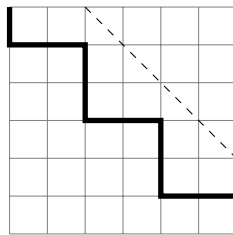


Figure 2: 2-lattice path corresponding to $\alpha = (6, 6, 4, 4, 2, 2)$.

Note that α_i corresponds to the i^{th} east step. Moreover, the facts that the first step is south and the path cannot pass the line $n - x + k$ implies that for all i , $\alpha_i \leq \min(n, n - i + k + 1)$.

This line of thought gives us the following variant of Theorem 2.2:

Theorem 2.3. Let $\alpha = (\alpha_1, \dots, \alpha_n) \in PP_n$ be a decreasing parking preference. Then, $\alpha \in PF_{n,k}^d$ if and only if $\alpha_i \leq \min(n, n - i + 1 + k)$ for all $1 \leq i \leq n$.

Formulas enumerating the number of parking functions and k -Naples parking functions are presented below.

Theorem 2.4 (See [11, 14]). For $n \geq 1$,

$$|PF_n| = (n + 1)^{n-1}.$$

In the next section we give a new formula to enumerate the number of k -Naples parking function. Prior to the present work, the only result enumerating k -Naples parking function was the following recursive formula.

Theorem 2.5 (Theorem 1.1, [10]). For $n \geq 0$ and $0 \leq k \leq n$,

$$|PF_{n+1,k}| = \sum_{i=0}^n \binom{n}{i} \min(i + 1 + k, n + 1) |PF_{i,k}| (n - i + 1)^{n-i-1}.$$

3. Counting parking functions through permutations

In this section, we provide a non-recursive formula for $|PF_{n,k}|$ as a sum over permutations. We begin by establishing the formula for the case $k = 0$, as it helps build intuition to generalize the result to $k > 0$.

3.1 The parking function case

Given a parking function, we consider the permutation, written in *one-line notation*, resulting from recording the position in which each car parks. The fibers of this map will allow us to count the parking functions.

Definition 3.1. *Given a parking function α , consider the function $\varphi : PF_n \rightarrow S_n$ which maps $\alpha = (\alpha_1, \dots, \alpha_n) \in PF_n$ to $\varphi(\alpha) = \sigma_1 \cdots \sigma_n$, where parking spot i is occupied by the σ_i^{th} car.*

We remark that the map φ is the inverse of the *outcome map* defined in [9] and slightly different from the definition which appears in *FindStat St001346* [15]. However, our results follow more naturally by defining φ as in Definition 3.1. In order for a parking preference to be a parking function, all the cars have to park and there cannot be two cars in the same parking space, therefore φ is well-defined. Moreover, every permutation is a parking function. In fact, φ maps each permutation to itself, so φ is surjective.

Example 3.1. *Let $\alpha = (4, 2, 2, 4, 1) \in PF_5$, so the cars c_1, c_2, c_3, c_4, c_5 park in the following way*

<i>spot 1</i>	<i>spot 2</i>	<i>spot 3</i>	<i>spot 4</i>	<i>spot 5</i>
c_5	c_2	c_3	c_1	c_4

Therefore, $\varphi(\alpha) = 52314$.

Now that we have a well-defined map from parking functions to permutations, a natural question to ask is how many parking functions map to a given permutation. In other words, given $\sigma \in S_n$, we would like to determine the size of the fiber $|\varphi^{-1}(\sigma)|$. For that, we need to introduce one more concept.

Definition 3.2. *Given a permutation $\sigma = \sigma_1 \cdots \sigma_n \in S_n$, we define $\ell(i; \sigma)$ as the length of the longest subsequence $\sigma_j \cdots \sigma_i$ of σ such that $\sigma_t \leq \sigma_i$ for all $j \leq t \leq i$.*

Let us see one example of how to compute $\ell(i; \sigma)$.

Example 3.2. *Let $\sigma = 23514$. We want to count the number of parking functions α such that $\varphi(\alpha) = \sigma$ by determining all possible entries of α given the entries of σ . To begin, we let entries of $\sigma = \sigma_1\sigma_2\sigma_3\sigma_4\sigma_5$ and consider each σ_i for $1 \leq i \leq 5$ individually:*

- *Since $\sigma_1 = 2$, we know that c_2 parked in spot 1. This implies that c_2 preferred the parking spot 1. We have $\ell(1; 23514) = 1$, which is the length of the subsequence 2 in σ .*
- *Since $\sigma_2 = 3$, we know that c_3 parked in spot 2. This implies that c_3 preferred the parking spots 1 or 2. We have $\ell(2; 23514) = 2$, which is the length of the subsequence 23 in σ .*
- *Since $\sigma_3 = 5$, we know that c_5 parked in spot 3. This implies that c_5 preferred the parking spots 1, 2, or 3. We have $\ell(3; 23514) = 3$, which is the length of the subsequence 235 in σ .*
- *Since $\sigma_4 = 1$, we know that c_1 parked in spot 4. This implies that c_1 preferred the parking spot 4. We have $\ell(4; 23514) = 1$, which is the length of the subsequence 1 in σ .*
- *Finally, since $\sigma_5 = 4$, we know that c_4 parked in spot 5. This implies that c_4 preferred the spot 4 or 5. We have $\ell(5; 23514) = 2$, which is the length of the subsequence 14 in σ .*

The following result answers the aforementioned question.

Proposition 3.1. *Let $\sigma = \sigma_1 \cdots \sigma_n \in S_n$ be a permutation. Then*

$$|\varphi^{-1}(\sigma)| = \prod_{i=1}^n \ell(i; \sigma).$$

Example 3.3. Continuing with Example 3.2, for $\sigma = 23514$, we have that multiplying all values of $\ell(i; 23514)$ for $1 \leq i \leq 5$, we have 12 distinct parking functions of length 5 mapping to $\sigma = 23514$ via the map φ . We list these 12 parking functions below:

$$\begin{array}{cccccc} 41141, & 41142, & 41143, & 41151, & 41152, & 41153, \\ 41241, & 41242, & 41243, & 41251, & 41252, & 41253. \end{array}$$

Proof of Proposition 3.1. Let $\alpha = (\alpha_1, \dots, \alpha_n) \in \varphi^{-1}(\sigma)$, $\sigma = \sigma_1\sigma_2 \cdots \sigma_n \in S_n$, and define $\pi \in S_n$ in such a way that $\sigma_{\pi(i)} = i$ for all $i \in [n]$. That is, c_i parks in the parking spot $\pi(i)$. Note that this corresponds to the outcome map [9]. Now, since π is a permutation in S_n , we have that

$$\prod_{i=1}^n \ell(\pi(i); \sigma) = \prod_{i=1}^n \ell(i; \sigma)$$

as the product of the lengths only changes the order of the terms. Thus, it suffices to show

$$|\varphi^{-1}(\sigma)| = \prod_{i=1}^n \ell(\pi(i); \sigma). \tag{1}$$

To do so, for $i \in [n]$ we claim that $\ell(\pi(i); \sigma)$ is the number of possible preferred parking spots α_i of c_i satisfying $\varphi(\alpha) = \sigma$. Proving this then implies that (1) follows by taking the product over all possibilities for $\alpha_1, \dots, \alpha_n$.

Let us prove our claim:

Given c_i , $\ell(\pi(i); \sigma)$ is the number of possible preferred parking spots α_i such that $\varphi(\alpha) = \sigma$.

To establish this claim we prove the following:

Fact 1: Every possible α_i is an index of the longest subsequence $\sigma_j \cdots \sigma_{\pi(i)}$ of σ such that $\sigma_t \leq \sigma_{\pi(i)}$ for all $j \leq t \leq \pi(i)$.

Fact 2: Every index of the longest subsequence $\sigma_j \cdots \sigma_{\pi(i)}$ of σ such that $\sigma_t \leq \sigma_{\pi(i)}$ for all $j \leq t \leq \pi(i)$ gives a possible α_i .

We begin by establishing Fact 1. By definition, c_i parks in the spot numbered $\pi(i)$. Otherwise, c_i would not be able to park in spot $\pi(i)$. More explicitly, we must have either

- (i) c_i parks in its preferred spot, and so $\alpha_i = \pi(i)$, or
- (ii) c_i tries to park in a previous spot $\alpha_i < \pi(i)$ but the parking spots $\alpha_i, \dots, \pi(i) - 1$ are all occupied. Thus $\{\alpha_i, \dots, \pi(i) - 1\} \subset \{\pi(1), \dots, \pi(i-1)\}$, so $\{\sigma_{\alpha_i}, \dots, \sigma_{\pi(i)-1}\} \subset \{\sigma_{\pi(1)}, \dots, \sigma_{\pi(i-1)}\} = \{1, \dots, i-1\}$. This means that for any choice of t satisfying $\alpha_i \leq t < \pi(i)$, there exists $r \in \{1, \dots, i-1\}$ such that $\sigma_t = \sigma_{\pi(r)} = r < i = \sigma_{\pi(i)}$.

Conditions (i) and (ii) imply that for all t satisfying $\alpha_i \leq t \leq \pi(i)$, we have $\sigma_t \leq \sigma_{\pi(i)}$, where the equality arises from condition (i). Hence, we have a necessary condition for α_i , which completes the proof of Fact 1.

Now we prove Fact 2. Consider $\sigma_j \cdots \sigma_{\pi(i)}$ such that $\sigma_t \leq \sigma_{\pi(i)}$, for all $j \leq t \leq \pi(i)$. This means spots $j, \dots, \pi(i) - 1$ have been occupied before c_i attempts to park. If $\alpha_i = \pi(i)$, then c_i parks in spot $\pi(i)$ and we are done. If $j \leq \alpha_i < \pi(i)$, as we know the spots $j, \dots, \pi(i) - 1$ are occupied, so c_i will find α_i occupied and by the parking rule will move forward and park at the first available spot which is spot $\pi(i)$. This establishes Fact 2 as the number of possibilities for α_i is precisely the length of the longest subsequence $\sigma_j \cdots \sigma_{\pi(i)}$ such that $\sigma_t \leq \sigma_{\pi(i)}$, for all $j \leq t \leq \pi(i)$. \square

Summing over the fiber for each permutation gives the number of parking functions.

Corollary 3.1 (Exercise 5.49(d,e) [21]).

$$\sum_{\sigma \in S_n} \left(\prod_{i=1}^n \ell(i; \sigma) \right) = |PF_n| = (n+1)^{n-1},$$

where for each $\sigma = \sigma_1 \cdots \sigma_n$, $\ell(i; \sigma)$ is the length of the longest subsequence $\sigma_j \cdots \sigma_i$ with $\sigma_t \leq i$ for all $j \leq t \leq i$.

Remark 3.1. Recently, Sanyal and Drohla [17] gave a different formula for counting parking functions, which they then related to binary trees.

3.2 k -Naples parking functions

We now generalize Corollary 3.1 to k -Naples parking functions. To start, we generalize the map φ .

Definition 3.3. Given a k -Naples parking function, consider the function $\varphi_k : PF_{n,k} \rightarrow S_n$ given by mapping a k -Naples parking function $\alpha \in PF_{n,k}$ to the permutation denoting the position in which the cars park under α using the k -Naples parking rule. That is, given $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \in PF_n$, $\varphi(\alpha) = \sigma_1 \sigma_2 \cdots \sigma_n$, where the spot i is occupied by the σ_i^{th} car.

As before, φ_k is the identity on $S_n \subset PF_{n,k}$ for all k . Moreover, from Definition 3.1 we note that $\varphi = \varphi_0$. However, φ_k is not an extension of φ for $k > 0$. Moreover, we note that the order in which cars park under φ_k might change when allowing the cars to back up one additional spot, as they might find that spot available rather than moving forward, in which case φ_{k+1} would be different than φ_k .

Example 3.4. Consider $\alpha = (4, 2, 2, 4, 1)$. By Example 3.1, we know that α is a parking function. In fact, α is also a 1-Naples parking function. Under the 1-Naples parking rule, the cars park in the following order:

spot 1	spot 2	spot 3	spot 4	spot 5
c_3	c_2	c_4	c_1	c_5

Therefore, $\varphi_1(\alpha) = 32415$. From Example 3.1, we note that $\varphi_1(\alpha) \neq \varphi(\alpha)$.

Just as $\ell(i; \sigma)$ was defined in terms of the subsequences to the left (see Proposition 3.1), we must now extend its definition and consider also subsequences to the right. This allows us to account for the backwards movement of the cars arising from the k -Naples parking rule.

Definition 3.4. Let $\sigma = \sigma_1 \cdots \sigma_n \in S_n$ be a permutation. For each $1 \leq i \leq n$, let $\text{left}_k(i; \sigma)$ be the length of longest subsequence $\sigma_j \cdots \sigma_{i-1}$ of σ such that $\sigma_t < \sigma_i$, for all $j \leq t < i$ and let $\text{right}_k(i; \sigma)$ be the length of longest subsequence $\sigma_i \cdots \sigma_r$ of σ such that $r \leq i + k$ and $\sigma_t \leq \sigma_i$ for all $i \leq t \leq r$. If these subsequences are empty, $\text{left}_k(i; \sigma) = 0$ or $\text{right}_k(i; \sigma) = 0$, respectively. Let $\ell_k(i; \sigma)$ be the function defined by

$$\ell_k(i; \sigma) = \begin{cases} \text{left}_k(i; \sigma) + \text{right}_k(i; \sigma) & \text{if } \text{left}_k(i; \sigma) = i - 1 \\ \max(\text{left}_k(i; \sigma) - k, 0) + \text{right}_k(i; \sigma) & \text{if } \text{left}_k(i; \sigma) < i - 1. \end{cases}$$

Before working through an example illustrating Definition 3.4 we remind the reader that the parameter k is the same as the parameter defining the k -Naples parking functions. Moreover, we can also restate the definition of $\ell_k(i; \sigma)$ in terms of the $\ell(i; \sigma)$ and the reversal map $\text{rev}(\sigma)$ which sends σ_i to σ_{n-i+1} .

Remark 3.2. The values $\text{left}_k(i; \sigma)$ and $\text{right}_k(i; \sigma)$ can be expressed in terms of $\ell(i; \sigma)$. In particular,

$$\begin{aligned} \text{left}_k(i; \sigma) &= \ell(i; \sigma) - 1 \\ \text{right}_k(i; \sigma) &= \min(k + 1, \ell(n + 1 - i; \text{rev}(\sigma))). \end{aligned}$$

Therefore, $\ell_k(i; \sigma)$ can also be expressed in terms of $\ell(i; \sigma)$. In particular,

$$\ell_k(i; \sigma) = \begin{cases} \ell(i; \sigma) - 1 + \min(k + 1, \ell(n + 1 - i; \text{rev}(\sigma))) & \text{if } \ell(i; \sigma) = i \\ \max(\ell(i; \sigma) - 1 - k, 0) + \min(k + 1, \ell(n + 1 - i; \text{rev}(\sigma))) & \text{if } \ell(i; \sigma) < i. \end{cases}$$

Example 3.5. Let $n = 5$ and $k = 2$, and consider the permutation $\sigma = 51423 \in S_5$. Then, $\text{left}_2(1; 51423) = \text{left}_2(2; 51423) = \text{left}_2(4; 51423) = 0$ because the corresponding subsequences are all the empty subsequence. Moreover, $\text{left}_2(3; 51423) = \text{left}_2(5; 51423) = 1$, corresponding to the subsequences 1 and 2, respectively.

Similarly, $\text{right}_2(1; 51423) = \text{right}_2(3; 51423) = 3$, corresponding to the subsequences 514 and 423, respectively, and $\text{right}_2(2; 51423) = \text{right}_2(4; 51423) = \text{right}_2(5; 51423) = 1$, corresponding to the subsequences 1, 2 and 3, respectively.

Therefore, we get the following values for $\ell_k(i; 51423)$: $\ell_2(1; 51423) = 3$, $\ell_2(2; 51423) = 1$, $\ell_2(3; 51423) = \max(1 - 2, 0) + 3 = 3$, $\ell_2(4; 51423) = 1$, and $\ell_2(5; 51423) = \max(1 - 2, 0) + 1 = 1$.

A straight forward computation establishes that there are 9 distinct 2-Naples parking functions of length 5 whose image under φ_2 is the permutation 51423. We list them below,

24531 24532 24533 24541 24542 24543 24551 24552 24553.

In our next result we establish that the count of these nine distinct 2-Naples parking functions arises as the product of the values $\ell_k(i; \sigma)$, which from above we note is given by $3 \times 1 \times 3 \times 1 \times 1 = 9$.

We are now ready to state the main result.

Theorem 3.1. *Let $\sigma \in S_n$ be a permutation. Then,*

$$|\varphi_k^{-1}(\sigma)| = \prod_{i=1}^n \ell_k(i; \sigma).$$

In particular, for $k = 0$, we obtain the result in Proposition 3.1.

Proof. Let $\alpha = (\alpha_1, \dots, \alpha_n) \in \varphi_k^{-1}(\sigma)$ and take $\pi \in S_n$ such that $\sigma_{\pi(i)} = i$. That is, c_i parks in the parking spot $\pi(i)$. To establish the result it suffices to show the following:

Claim: For each $1 \leq i \leq n$, $\ell_k(\pi(i); \sigma)$ is the number of possible preferred parking spots α_i which allows c_i to park in spot $\pi(i)$.

Proving this claim then implies the result as we take the product over all possibilities for $\alpha_1, \dots, \alpha_n$.

Given i , let $\sigma_j \cdots \sigma_{\pi(i)-1}$ be the longest subsequence of $\sigma = \sigma_1 \cdots \sigma_n$ such that $\sigma_t < \sigma_{\pi(i)}$ for all $j \leq t < \pi(i)$ and $\sigma_{\pi(i)} \cdots \sigma_r$ the longest subsequence of σ such that $\sigma_t \leq \sigma_{\pi(i)}$ for all $\pi(i) \leq t \leq r \leq \pi(i) + k$. These are exactly the subsequences whose length define the values $\mathbf{left}_k(\pi(i); \sigma)$ and $\mathbf{right}_k(\pi(i); \sigma)$ in the definition of $\ell_k(\pi(i); \sigma)$ (See Definition 3.4.)

Our next step is to show that these are precisely all the possibilities for the parking preferences α_i of c_i . By definition, c_i parks in the spot numbered $\pi(i)$. Note that just before c_i parks, the parking spots $\pi(1), \dots, \pi(i-1)$ have been occupied by c_1, \dots, c_{i-1} respectively. By the k -Naples parking rule, we consider three cases depending on where c_i parks:

Case 1: c_i parks in its preferred spot α_i , or

Case 2: c_i parks in a spot before its preferred spot α_i , or

Case 3: c_i parks in a spot after its preferred spot α_i .

By definition, the count for $\mathbf{right}_k(\pi(i); \sigma)$ involves only the cases (1) and (2), whereas the count for $\mathbf{left}_k(\pi(i); \sigma)$ involves only case (3). Note also that the contribution of $\mathbf{right}_k(\pi(i); \sigma)$ to $\ell_k(\pi(i); \sigma)$ is the same independently of the value of $\mathbf{left}_k(\pi(i); \sigma)$.

We now consider the implications arising from each of the possible cases defined above.

Case 1: Assume c_i parks in its preferred spot α_i .

In this case, $\alpha_i = \pi(i)$ and this contributes to the value of $\mathbf{right}_k(\pi(i); \sigma)$.

Case 2: Assume c_i parks in a spot before its preferred spot α_i .

In this case, c_i tries to park in the spot α_i , with $\pi(i) < \alpha_i \leq \pi(i) + k$. However, the parking spots $\pi(i) + 1, \dots, \alpha_i$ are all occupied, and c_i backs up until spot $\pi(i)$. Now, this implies that $\{\pi(i) + 1, \dots, \alpha_i\}$ is a subset of the previously occupied spots $\{\pi(1), \dots, \pi(i-1)\}$. Thus, $\{\sigma_{\pi(i)+1}, \dots, \sigma_{\alpha_i}\} \subset \{\sigma_{\pi(1)}, \dots, \sigma_{\pi(i-1)}\} = \{1, \dots, i-1\}$. Therefore, for any t satisfying $\pi(i) < t \leq \alpha_i$, there exists $r \in \{1, \dots, i-1\}$ such that $\sigma_t = \sigma_{\pi(r)} = r \leq i = \sigma_{\pi(i)}$ and this case contributes to the value of $\mathbf{right}_k(\pi(i); \sigma)$.

Case 3: Assume c_i parks in a spot after its preferred spot α_i .

In this case, c_i tries to park in α_i , with $\alpha_i < \pi(i)$, which is occupied. In this case, the contribution to $\ell_k(\pi(i); \sigma)$ depends on the value of $\mathbf{left}_k(\pi(i); \sigma)$. We look at each possible case:

Subcase 3a: If $\text{left}_k(\pi(i); \sigma) = \pi(i) - 1$, the spots before $\pi(i)$ are all occupied. That is, there are no empty spots available before α_i and between α_i and $\pi(i)$. Then, $\{\alpha_i, \dots, \pi(i) - 1\}$ is a subset of the previously occupied spots $\{\pi(1), \dots, \pi(i-1)\}$. Thus, $\{\sigma_{\alpha_i}, \dots, \sigma_{\pi(i)-1}\} \subset \{\sigma_{\pi(1)}, \dots, \sigma_{\pi(i-1)}\} = \{1, \dots, i-1\}$. Therefore, for any t satisfying $\alpha_i \leq t < \pi(i)$, there exists $r \in \{1, \dots, i-1\}$ such that $\sigma_t = \sigma_{\pi(r)} = r < i = \sigma_{\pi(i)}$ and this contributes to the value of $\text{left}_k(\pi(i); \sigma)$.

Subcase 3b: If $\text{left}_k(\pi(i); \sigma) < \pi(i) - 1$, then there exists an empty spot before $\pi(i)$. In fact, this empty spot has to be before $\alpha_i - k$. Otherwise, by the k -Naples parking rule, c_i may be able to back up into a parking spot before α_i , which would yield a contradiction. Therefore, there are $\max(\text{left}_k(i; \sigma) - k, 0)$ spots available since $\text{left}_k(i; \sigma) - k$ can be negative.

In Cases 1-3, either $\sigma_t < \sigma_{\pi(i)}$, for all t satisfying $\alpha_i \leq t < \pi(i)$, or $\sigma_t \leq \sigma_{\pi(i)}$, for all t satisfying $\pi(i) \leq t \leq \alpha_i \leq \pi(i) + k$. These give necessary conditions for α_i in order for c_i to park in spot $\pi(i)$, proving our claim.

As in the proof of Proposition 3.1, it is clear that any index of the subsequence corresponding to $\ell_k(\pi(i); \sigma)$ is a parking preference α_i which will cause c_i to park at the spot $\pi(i)$, providing the sufficient condition, and thus completing the proof of the theorem. \square

As a consequence of Theorem 3.1, we give a new expression for the number of k -Naples parking functions.

Theorem 3.2. For all $n \geq 1$ and $0 \leq k \leq n - 1$,

$$|PF_{n,k}| = \sum_{\sigma \in S_n} \left(\prod_{i=1}^n \ell_k(i; \sigma) \right).$$

This result provides a new formula enumerating the k -Naples parking functions. Compared to Theorem 2.5, this formula is not recursive and is presented in terms of a familiar set of combinatorial objects, that is, permutations.

4. A logarithmic generating function on the fibers of φ

Next we give a recurrence relation for the fibers of φ . That is, we present a formula for counting the number of permutations whose fiber under φ has a given size.

Let us denote by $F_n(q)$ the generating function of the fiber of φ for each $n \geq 0$.

$$F_n(q) = \sum_{i \geq 1} c_{n,i} q^i,$$

where the coefficient $c_{n,i}$ counts the number of permutations in S_n whose fiber under φ has size i . Note that $F_0(q) = q$ since there is trivially one permutation of length 0 and one parking function of length 0 mapping to that permutation. The first five polynomials are given in [15]:

$$\begin{aligned} F_1(q) &= q, \\ F_2(q) &= q + q^2, \\ F_3(q) &= q + 3q^2 + q^3 + q^6, \\ F_4(q) &= q + 6q^2 + 4q^3 + 4q^4 + 4q^6 + 3q^8 + q^{12} + q^{24}, \\ F_5(q) &= q + 10q^2 + 10q^3 + 20q^4 + q^5 + 20q^6 + 15q^8 + 6q^{10} + 15q^{12} \\ &\quad + 4q^{15} + 4q^{20} + 5q^{24} + 4q^{30} + 3q^{40} + q^{60} + q^{120}. \end{aligned}$$

Defining $c_{n,i} = 0$ if $n < 0$, we next show how the coefficients $c_{n,i}$ can be computed recursively.

Proposition 4.1. For $1 \leq i \leq n!$,

$$c_{n,i} = \sum_{d|i} \left[\binom{n-1}{d-1} \left(\sum_{j|\frac{i}{d}} c_{d-1,j} c_{n-d, \frac{i}{dj}} \right) \right],$$

and $c_{n,i} = 0$, otherwise.

Proof. Given i , we want the number of elements $\sigma \in S_n$ whose fiber has size i , i.e., $|\varphi^{-1}(\sigma)| = i$. Let $\sigma = \sigma_1 \dots \sigma_n \in S_n$ be one of those permutations, and let d be the index such that $\sigma_d = n$. We split σ into three pieces (allowing for the case where some may be empty)

$$\sigma = \underbrace{\sigma_1 \dots \sigma_{d-1}}_{s'} \sigma_d \underbrace{\sigma_{d+1} \dots \sigma_n}_{s''}.$$

Assuming neither s' nor s'' are empty, we look at s' and s'' as permutations of $d-1 := \ell(s')$ and $n-d := \ell(s'')$, by reducing the values to $1, \dots, d-1$ for s' and $1, \dots, n-d$ for s'' while preserving the relative order of the entries. Let us denote by σ' and σ'' those permutations (respectively). Therefore, we have that

$$i = |\varphi^{-1}(\sigma)| = |\varphi^{-1}(\sigma')| \cdot d \cdot |\varphi^{-1}(\sigma'')|.$$

On the other hand, if s' or s'' is empty, then they are the empty permutation in which case either $|\varphi^{-1}(\sigma')| = 1$ or $|\varphi^{-1}(\sigma'')| = 1$.

Since $|\varphi^{-1}(\sigma')|$ and $|\varphi^{-1}(\sigma'')|$ are integers, d must be a factor of i . In fact, for each possible d , there are $\binom{n-1}{d-1}$ ways to choose the first d cars (unordered). The ordering of these $d-1$ cars is uniquely determined by σ' . Thus,

$$c_{n,i} = \sum_{d|i} \left[\binom{n-1}{d-1} \left(\sum_{j|\frac{i}{d}} c_{d-1,j} c_{n-d,\frac{i}{dj}} \right) \right]. \quad \square$$

Consider now the *logarithmic generating function* defined for $n \geq 1$ by

$$G_n(q) = \sum_{i=1}^n c_{n,i} q^{\ln i}.$$

This generating function allows us to manipulate exponents easily, obtaining the following recursive formula.

Proposition 4.2. For $n \geq 1$,

$$G_n(q) = \sum_{i=0}^{n-1} \binom{n-1}{i} q^{\ln(i+1)} G_i(q) G_{n-1-i}(q).$$

Proof. By Proposition 4.1, we have that

$$\begin{aligned} G_n(q) &= \sum_{i=1}^n c_{n,i} q^{\ln i} = \sum_{i|n!} c_{n,i} q^{\ln i} = \sum_{i|n!} \sum_{d|i} q^{\ln d} \binom{n-1}{d-1} \left(\sum_{j|\frac{i}{d}} c_{d-1,j} q^{\ln j} c_{n-d,\frac{i}{dj}} q^{\ln \frac{i}{dj}} \right) \\ &= \sum_{i|n!} \sum_{d|i} q^{\ln d} \binom{n-1}{d-1} \left(\sum_{\substack{r \text{ s.t.} \\ d|r, r|i}} c_{d-1,\frac{r}{d}} q^{\ln \frac{r}{d}} c_{n-d,\frac{i}{r}} q^{\ln \frac{i}{r}} \right). \end{aligned}$$

Now, since $\binom{n-1}{d-1} = 0$ for $d > n$, we can restrict the summation over d and also change the order of the summations over d and over i . Thus, we have that

$$\begin{aligned} G_n(q) &= \sum_{i|n!} \sum_{\substack{d|i \\ d \leq n}} q^{\ln d} \binom{n-1}{d-1} \left(\sum_{\substack{r \text{ s.t.} \\ d|r, r|i}} c_{d-1,\frac{r}{d}} q^{\ln \frac{r}{d}} c_{n-d,\frac{i}{r}} q^{\ln \frac{i}{r}} \right) \\ &= \sum_{d=1}^n q^{\ln d} \binom{n-1}{d-1} \left(\sum_{\substack{i \text{ s.t.} \\ d|i, i|n!}} \sum_{\substack{r \text{ s.t.} \\ d|r, r|i}} c_{d-1,\frac{r}{d}} q^{\ln \frac{r}{d}} c_{n-d,\frac{i}{r}} q^{\ln \frac{i}{r}} \right) \end{aligned}$$

$$= \sum_{d=1}^n q^{\ln d} \binom{n-1}{d-1} \left(\sum_{i' \mid \frac{n!}{d}} \sum_{r' \mid i'} c_{d-1, r'} q^{\ln r'} c_{n-d, \frac{i'}{r'}} q^{\ln \frac{i'}{r'}} \right),$$

where in the last equality we rewrite the summation with a new index. Again, we have that $c_{d-1, r} = 0$ if $r \nmid (d-1)!$, and similarly for $\frac{i'}{r'}$. Therefore,

$$G_n(q) = \sum_{d=1}^n q^{\ln d} \binom{n-1}{d-1} \left(\sum_{r'} \sum_{i'} c_{d-1, r'} q^{\ln r'} c_{n-d, \frac{i'}{r'}} q^{\ln \frac{i'}{r'}} \right),$$

where the second summation is over the r' such that $r' \mid \frac{n!}{d}$ and $r' \mid (d-1)!$, and the third summation is over i' such that $\frac{i'}{r'} \mid \frac{n!}{r'd}$ and $\frac{i'}{r'} \mid (n-d)!$, that is,

$$G_n(q) = \sum_{d=1}^n q^{\ln d} \binom{n-1}{d-1} \left(\sum_{r' \mid (d-1)!} c_{d-1, r'} q^{\ln r'} \sum_{\frac{i'}{r'} \mid (n-d)!} c_{n-d, \frac{i'}{r'}} q^{\ln \frac{i'}{r'}} \right).$$

(Note that $r' \mid (d-1)! \Rightarrow r' \mid (d-1)! \frac{n(n-1)\dots d}{d} = \frac{n!}{d}$ so the first condition on r' is unnecessary. For $i', \frac{i'}{r'} \mid (n-d)! \Rightarrow i'$ divides $r'(n-d)!$, which in turn divides $r'(n-d)! \binom{n}{d} = r'(n-d)! \frac{n(n-1)\dots(n-d+1)}{d(d-1)!}$, which divides $(n-d)! \frac{n(n-1)\dots(n-d+1)}{d} = \frac{n!}{d}$ since $r' \mid (d-1)!$, so the first condition on i' is also redundant). Now, the summation over i' is exactly the definition of $G_{n-d}(q)$ and the summation over r' is the definition of $G_{d-1}(q)$. Therefore,

$$G_n(q) = \sum_{d=1}^n \binom{n-1}{d-1} q^{\ln(d)} G_{d-1}(q) G_{n-d}(q).$$

Writing $i = d - 1$, we have that

$$G_n(q) = \sum_{i=0}^{n-1} \binom{n-1}{i} q^{\ln(i+1)} G_i(q) G_{n-1-i}(q). \quad \square$$

5. Finding q -analogues

Given a non-negative integer n , we denote its q -analog by

$$[n]_q = \frac{1 - q^n}{1 - q} = 1 + q + q^2 + \dots + q^{n-1}.$$

With this q -analog, one may define q -analogs of almost any formula. For instance, the q -factorial or the q -binomial coefficients. In general, we denote by $[\star]_q$ the q -analog of the formula \star by substituting any number appearing in \star by its q -analog as defined above. This allows us to turn almost anything into a polynomial.

Together with q -analogs, we also have q -statistics, which allow us to understand combinatorially the exponents of q appearing in the q -analog formulas, mostly polynomials in q , once they are expressed as summations over the exponents of q . *FindStat* [15] is a great source to find many statistics as well as good references.

The q -analog appears naturally in several contexts in algebraic combinatorics, and in particular, in the the framework of parking functions and diagonal harmonics. In [12], Loehr presents several identities involving q -analogs in the framework of diagonal harmonics. We refer to [12] for more details on the diagonal harmonics. One of the q -statistics that appears in our setting is the **area** of a parking function, which we now define.

Definition 5.1. *Given a parking function $\alpha \in PF_n$, let P be the labeled Dyck path associated to it. We define $\mathbf{area}(\alpha)$ as the number of complete lattice squares between P and the line $y = x$. Equivalently, the **area** of the parking function α can be computed using the more explicit formula*

$$\mathbf{area}(\alpha) = \sum_{i=1}^n (n - i - \alpha_i + 1).$$

Our first result provides the distribution of the area statistic over the fiber $\varphi^{-1}(\sigma)$, for any permutation σ , by looking at the q -analog of Corollary 3.1.

Proposition 5.1. *For any permutation $\sigma \in S_n$,*

$$\sum_{p \in \varphi^{-1}(\sigma)} q^{\text{area}(p)} = \prod_{i=1}^n [\ell(i; \sigma)]_q. \tag{2}$$

We present two proofs of this result: one bijective and one inductive. The bijective proof is generalized for k -Naples parking functions in Section 5.1.

Bijective proof. Using notation from [12, Fermionic formula], the right-hand side of (2) gives a generating function for objects (u_1, u_2, \dots, u_n) with $u_i < \ell(i; \sigma)$, where q^m corresponds to such an object with $\sum u_i = m$. Call the set of these objects \mathcal{I} and define $\text{stat}((u_i)) = \sum u_i$. Immediately

$$\prod_{i=1}^n [\ell(i; \sigma)]_q = \sum_{(u_i) \in \mathcal{I}} q^{\text{stat}((u_i))},$$

so it remains to find a bijection $f : \mathcal{I} \rightarrow PF_n$ with $\text{stat} = \text{area} \circ f$. Consider arbitrary $(u_i) \in \mathcal{I}$ and choose π based on σ as in the proof of Proposition 3.1. Now we define the map

$$\begin{aligned} f : \mathcal{I} &\longrightarrow PF_n \\ (u_i) &\longmapsto (\pi_1 - u_1, \pi_2 - u_2, \dots, \pi_n - u_n). \end{aligned}$$

We claim that f is a bijection. It is well-defined because π is a parking function, and weakly reducing all terms of a parking function gives a parking function. Injectivity follows since π can be recovered as the outcome of the image, and then $u_i = \pi_i - \alpha_i$. The function is surjective by the casework from Proposition 3.1. Since swapping columns preserves area, the permutation π can be rearranged to the permutation $n \ n-1 \cdots 21$ which has no area. Now reducing each π_i by u_i adds u_i to the area, so $\text{area} \circ f = \sum u_i$ as desired. \square

Inductive proof. For $n = 1$, there is only one permutation $\sigma = (1)$, for which $\varphi^{-1}(\sigma) = PF_1 = \{(1)\}$. Therefore,

$$\sum_{p \in \varphi^{-1}(\sigma)} q^{\text{area}(p)} = q^{\text{area}(1)} = q^0 = 1 = [\ell(1; 1)]_q = \prod_{i=1}^n [\ell(i; \sigma)]_q.$$

Now, suppose that the statement is true for $1, \dots, n-1$. Let $\sigma = \sigma_1 \cdots \sigma_n \in S_n$. Let d be the index such that $\sigma_d = n$. As in the proof of Proposition 4.2, we split σ into three pieces

$$\sigma = \underbrace{\sigma_1 \cdots \sigma_{d-1}}_{s'} \sigma_d \underbrace{\sigma_{d+1} \cdots \sigma_n}_{s''},$$

and we consider the permutations $\sigma' \in S_{d-1}$ and $\sigma'' \in S_{n-d}$.

Since the value of $\ell(i; \sigma)$ only depends on the relative order of the numbers, $\ell(i; \sigma) = \ell(i; \sigma')$ for all $1 \leq i \leq d-1$, $\ell(i; \sigma) = \ell(i-d; \sigma'')$ for all $d+1 \leq i \leq n$, and $\ell(d; \sigma) = d$. Thus by induction hypothesis,

$$\begin{aligned} \prod_{i=1}^n [\ell(i; \sigma)]_q &= \left(\prod_{i=1}^{d-1} [\ell(i; \sigma')]_q \right) [d]_q \left(\prod_{i=1}^{n-d} [\ell(i; \sigma'')]_q \right) \\ &= [d]_q \left(\sum_{p' \in \varphi^{-1}(\sigma')} q^{\text{area}(p')} \right) \left(\sum_{p'' \in \varphi^{-1}(\sigma'')} q^{\text{area}(p'')} \right). \end{aligned}$$

Let t_1, \dots, t_{d-1} and t_{d+1}, \dots, t_n be the increasing rearrangements of $\sigma_1, \dots, \sigma_{d-1}$ and $\sigma_{d+1}, \dots, \sigma_n$, respectively. For the RHS in the formula of Proposition 5.1, since the n^{th} car c_n parks at spot d , the parking functions $(\alpha_1, \dots, \alpha_n)$ which map to $\sigma = \sigma_1 \cdots \sigma_n$ are precisely those such that $(\alpha_{t_1}, \dots, \alpha_{t_{d-1}}) \in \varphi^{-1}(\sigma')$, $(\alpha_{t_{d+1}}, \dots, \alpha_{t_n}) \in \varphi^{-1}(\sigma'')$, and $\alpha_n \in \{1, \dots, d\}$. Moreover, the area of a parking function is the area of its decreasing rearrangement, and so

$$\text{area}(\alpha_1, \dots, \alpha_n) = \text{area}(\alpha_{t_1}, \dots, \alpha_{t_{d-1}}) + \text{area}(\alpha_{t_{d+1}}, \dots, \alpha_{t_n}) + d - \alpha_n.$$

Therefore, we can say that

$$\begin{aligned} \sum_{p \in \varphi^{-1}(\sigma)} q^{\text{area}(p)} &= \sum_{r=1}^d \sum_{p' \in \varphi^{-1}(\sigma')} \sum_{p'' \in \varphi^{-1}(\sigma'')} q^{\text{area}(p')} q^{\text{area}(p'')} q^{d-r} \\ &= \left(1 + q + \dots + q^{d-1}\right) \left(\sum_{p' \in \varphi^{-1}(\sigma')} q^{\text{area}(p')}\right) \left(\sum_{p'' \in \varphi^{-1}(\sigma'')} q^{\text{area}(p'')}\right) \\ &= [d]_q \left(\sum_{p' \in \varphi^{-1}(\sigma')} q^{\text{area}(p')}\right) \left(\sum_{p'' \in \varphi^{-1}(\sigma'')} q^{\text{area}(p'')}\right) = \prod_{i=1}^n [\ell(i; \sigma)]_q, \end{aligned}$$

which completes the inductive step. □

Corollary 5.1. *For each n , we have*

$$\sum_{p \in PF_n} q^{\text{area}(p)} = \sum_{\sigma \in S_n} \prod_{i=1}^n [\ell(i; \sigma)]_q.$$

Proof. By Proposition 5.1, summing over all fibers, we have that

$$\sum_{\sigma \in S_n} \prod_{i=1}^n [\ell(i; \sigma)]_q = \sum_{\sigma \in S_n} \sum_{p \in \varphi^{-1}(\sigma)} q^{\text{area}(p)} = \sum_{p \in PF_n} q^{\text{area}(p)}. \quad \square$$

By [12, Univariate symmetry], we have the following chain of identities.

Corollary 5.2. *The following chain of identities holds:*

$$\sum_{\sigma \in S_n} \left(\prod_{i=1}^n [\ell(i; \sigma)]_q\right) = \sum_{p \in PF_n} q^{\text{area}(p)} = \sum_{p \in PF_n} q^{\text{dinv}(p)} = \sum_{p \in PF_n} q^{\text{pmaj}(p)},$$

where *dinv* and *pmaj* are other well-known statistics on parking functions (see [12]).

In fact, following the q, t -statistics presented in [12], we arrive at the following result.

Corollary 5.3. *For any statistic stat on permutations,*

$$\sum_{\sigma \in S_n} t^{\text{stat}(\sigma)} \left(\prod_{i=1}^n [\ell(i; \sigma)]_q\right) = \sum_{p \in PF_n} q^{\text{area}(p)} t^{\text{stat}(\varphi(p))}.$$

5.1 The k -Naples area statistics

Next, we generalize Proposition 5.1 to k -Naples parking functions. We begin by introducing an *area statistic* for the k -Naples parking functions. For parking functions, the area statistic counts the distance between the Dyck path and the main diagonal, see Definition 5.1. Alternatively, one can think about the area as distance between the lattice path and the highest point among lattice paths in the same fiber of φ . Therefore, our definition for the k -Naples area counts the distance between the lattice path given by the α_i 's and the highest that path can be while the corresponding parking function still remains in the fiber of φ_k .

Definition 5.2. *Given a parking function $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$, we define the k -Naples area by*

$$\text{area}_k(\alpha) = \sum_{i=1}^n [n - i + \text{right}_k(i; \varphi_k(\alpha)) - \alpha_i].$$

Note that for $k = 0$, $\text{right}_k(i; \sigma) = 1$ and we recover the definition of area for parking functions. Bear in mind, however, that $\text{right}_k(i; \sigma)$ also depends on k .

Example 5.1. *Consider $\alpha = (3, 2, 2)$ as a 1-Naples parking function, for which $\varphi_1(\alpha) = 321$. Now $\text{right}_1(1; 321) = \text{right}_1(2; 321) = 2$ and $\text{right}_1(3; 321) = 1$. Therefore,*

$$\text{area}_1(\alpha) = (3 - 1 + 2 - 3) + (3 - 2 + 2 - 2) + (3 - 3 + 1 - 2) = 1.$$

Now, we study the distribution of the \mathbf{area}_k statistic over the fiber $\varphi_k^{-1}(\sigma)$, where σ is a permutation. We find that it is the q -analog of the formula in Theorem 3.1.

Proposition 5.2. *For any permutation $\sigma \in S_n$,*

$$\sum_{p \in \varphi_k^{-1}(\sigma)} q^{\mathbf{area}_k(p)} = \prod_{i=1}^n [\ell_k(i; \sigma)]_q.$$

Proof. As in the proof of Proposition 5.1, consider the q -analogue as a generating function for objects (u_i) , this time with $u_i < \ell_k(i; \sigma)$. Again, let \mathcal{I} be the set of these objects, and define a statistic $\mathbf{stat}((u_i)) = \sum u_i$. Now the q -analogue is a generating function satisfying

$$\prod_{i=1}^n [\ell_k(i; \sigma)]_q = \sum_{(u_i) \in \mathcal{I}} q^{\mathbf{stat}((u_i))}.$$

It remains to find a bijection $f : \mathcal{I} \rightarrow \varphi_k^{-1}(\sigma)$ with $\mathbf{stat} = \mathbf{area}_k \circ f$. We define the map f as

$$f((u_i)) = (\pi_1 + \mathbf{right}_k(1; \sigma) - u_1 - 1, \pi_2 + \mathbf{right}_k(2; \sigma) - u_2 - 1, \dots, \pi_n + \mathbf{right}_k(n; \sigma) - u_n - 1),$$

for $(u_i) \in \mathcal{I}$, where π is defined in the proof of Theorem 3.1. We claim that f is a bijection. To see that f is well-defined, we start with the permutation $\pi = \pi_1 \cdots \pi_n$, which is a parking function. Then, adding $\mathbf{right}_k(i; \sigma) - 1$ to π_i leads to the highest point that the path could be, which is still a k -Naples parking function. Finally, weakly reducing each entry by u_i gives a k -Naples parking function in the same fiber of φ_k . Moreover, given $\alpha \in \varphi_k^{-1}(\sigma)$, we have that $f^{-1}(\alpha)$ is defined by $u_i = \alpha_i - \pi_i - \mathbf{right}_k(i; \sigma) + 1$. This formula is obtained directly from the definition of f by taking α_i as the i^{th} entry of $f((u_i))$ and writing u_i in terms of α_i , π_i and $\mathbf{right}_k(i; \sigma)$, using that the values of π_i and $\mathbf{right}_k(i; \sigma)$ are given by σ .

Finally, the fact that f is surjective follows from the casework in the proof of Theorem 3.1.

Now the k -Naples area is

$$\begin{aligned} \mathbf{area}_k \circ f &= \sum [n - i + \mathbf{right}_k(i; \sigma) - (\pi_i + \mathbf{right}_k(i; \sigma) - u_i - 1)] \\ &= \sum [n - i - \pi_i + u_i + 1] = \sum u_i = \mathbf{stat}((u_i)). \end{aligned} \quad \square$$

Taking the q -analogues of the $\ell_k(i; \sigma)$ in Theorem 3.2 gives the distribution of the \mathbf{area}_k statistic over $PF_{n,k}$.

Proposition 5.3. *For $n \geq 1$,*

$$\sum_{p \in PF_{n,k}} q^{\mathbf{area}_k(p)} = \sum_{\sigma \in S_n} \prod_{i=1}^n [\ell_k(i; \sigma)]_q.$$

Proof. By Proposition 5.2, summing over all fibers gives

$$\sum_{p \in PF_{n,k}} q^{\mathbf{area}_k(p)} = \sum_{\sigma \in S_n} \sum_{p \in \varphi_k^{-1}(\sigma)} q^{\mathbf{area}_k(p)} = \sum_{\sigma \in S_n} \prod_{i=1}^n [\ell_k(i; \sigma)]_q. \quad \square$$

We conclude with Table 1, where we list values of the k -Naples area for some small values of k and n .

Acknowledgements

Part of this research was performed with support from the Institute for Pure and Applied Mathematics (IPAM), which is supported by the National Science Foundation (Grant No. DMS-1440415), from the EDGE Foundation, and from private donations of Joan Barksdale and Nancy Sinclair. LC was partially supported by MTM2016-75024-P. PEH was partially supported by The Karen Uhlenbeck EDGE Fellowship. ARVM was partially supported by NSF Graduate Research Fellowship DGE-1247392 and the NSF KY-WV LSAMP Bridge to Doctorate Fellowship HRD-2004710. The authors want to thank Raman Sanyal for sharing his work with Elias Drohla in private communication, and for pointing out the reference to *Exercise 5.49(d,e)* in [21], and Ayo Adeniran for his helpful insights throughout the project. ES would also like to thank Ralph Morrison for connecting her to PEH and Bernd Sturmfels for funding through his BEAR fund at UC Berkeley.

n	k	Distribution of area_k over $PF_{n,k}$
1	0	1
2	0	$q + 2$
	1	$2q + 2$
3	0	$q^3 + 3q^2 + 6q + 6$
	1	$2q^3 + 7q^2 + 9q + 6$
	2	$3q^3 + 9q^2 + 9q + 6$
4	0	$q^6 + 4q^5 + 10q^4 + 20q^3 + 30q^2 + 36q + 24$
	1	$2q^6 + 9q^5 + 24q^4 + 41q^3 + 53q^2 + 50q + 24$
	2	$3q^6 + 13q^5 + 34q^4 + 58q^3 + 60q^2 + 48q + 24$
	3	$4q^6 + 16q^5 + 40q^4 + 64q^3 + 60q^2 + 48q + 24$
5	0	$q^{10} + 5q^9 + 15q^8 + 35q^7 + 70q^6 + 120q^5 + 180q^4 + 240q^3 + 270q^2 + 240q + 120$
	1	$2q^{10} + 11q^9 + 35q^8 + 84q^7 + 165q^6 + 263q^5 + 361q^4 + 429q^3 + 435q^2 + 320q + 120$
	2	$3q^{10} + 16q^9 + 50q^8 + 121q^7 + 238q^6 + 384q^5 + 502q^4 + 529q^3 + 462q^2 + 306q + 120$
	3	$4q^{10} + 21q^9 + 65q^8 + 155q^7 + 295q^6 + 464q^5 + 576q^4 + 550q^3 + 450q^2 + 300q + 120$
	4	$5q^{10} + 25q^9 + 75q^8 + 175q^7 + 325q^6 + 500q^5 + 600q^4 + 550q^3 + 450q^2 + 300q + 120$

Table 1: Distribution of the k -Naples area for first values of $k < n$ computed with [18].

References

- [1] A. Adeniran, S. Butler, G. Dorpalen-Barry, P. E. Harris, C. Hettle, Q. Liang, J. L. Martin and H. Nam, *Enumerating parking completions using join and split*, Electron. J. Combin. 27(2) (2020), #P2.44.
- [2] A. Adeniran, *Gončarov polynomials, partition lattices and parking sequences*, Doctoral dissertation, Texas A&M University, 2020. Available electronically from <http://hdl.handle.net>.
- [3] A. Adeniran and C. Yan, *Gončarov polynomials in partition lattices and exponential families*, Adv. in Appl. Math., 2020.
- [4] A. Adeniran and C. Yan, *On increasing and invariant parking sequences*, 2020, <https://arxiv.org/pdf/2005.04759.pdf>.
- [5] A. Baumgardner, *The Naples parking function*, Honors Contract-Graph Theory, Florida Gulf Coast University, 2019.
- [6] B. Benson, D. Chakrabarty, and P. Tetali, *G-parking functions, acyclic orientations and spanning trees*, Discrete Math. 310(8) (2010), 1340–1353.
- [7] M. Bóna, *Handbook of enumerative combinatorics*, Discrete Mathematics and Its Applications, CRC Press, Hoboken, NJ, 2015.
- [8] J. Carlson, A. Christensen, P.E. Harris, Z. Jones, and A. Ramos Rodríguez, *Parking functions: Choose your own adventure*, 2020.
- [9] E. Colaric, R. DeMuse, J. Martin, and M. Yin, *Interval parking functions*, 2020, <https://arxiv.org/pdf/2006.09321.pdf>.
- [10] 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. Combin. 27(1) (2020), #P1.33.
- [11] A. Konheim and B. Weiss, *An occupancy discipline and applications*, SIAM J. Appl. Math. 14(6) (1966), 1266–1274.
- [12] N. Loehr, *Combinatorics of q , t -parking functions*, Adv. in Appl. Math. 34(2) (2005), 408–425.
- [13] J. Pitman and R. P. Stanley, *A polytope related to empirical distributions, plane trees, parking functions, and the associahedron*, Discrete Comput. Geom. 27(4) (2002), 603–634.
- [14] R. Pyke, *The supremum and infimum of the Poisson process*, Ann. Math. Statist. 30 (1959), 568–576.

- [15] M. Rubey, C. Stump, et al., *FindStat - The combinatorial statistics database*, Accessed July 2020.
- [16] M. Schützenberger, *On an enumeration problem*, J. Combinatorial Theory, 4 (1968), 219–221.
- [17] R. Sanyal and E. Drohla, *Parking-functions, schattenvektoren und Catalan-zahlen*, Bachelor thesis, 2020.
- [18] W. Stein et al., *Sage Mathematics Software (Version 9.0)*, The Sage Development Team, 2020.
- [19] R. Stanley, *Hyperplane arrangements, interval orders, and trees*, Proc. Nat. Acad. Sci. U.S.A. 93(6) (1996), 2620–2625.
- [20] R. Stanley, *Parking functions and noncrossing partitions*, volume 4(2), The Wilf Festschrift (Philadelphia, PA, 1996), 1997.
- [21] R. Stanley, *Enumerative combinatorics. Vol. 2*, volume 62 of Cambridge Studies in Advanced Mathematics, Cambridge University Press, Cambridge, 1999.