COMBINATORIAL PROBLEMS RELATED TO OPTIMAL TRANSPORT AND PARKING FUNCTIONS

by

Jan Kretschmann

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Mathematics

at

The University of Wisconsin–Milwaukee

December 2023

ABSTRACT

COMBINATORIAL PROBLEMS RELATED TO OPTIMAL TRANSPORT AND PARKING FUNCTIONS

by

Jan Kretschmann

The University of Wisconsin–Milwaukee, 2023
Under the Supervision of Professors Jeb F. Willenbring and Pamela E. Harris

In the first part of this work, we provide contributions to optimal transport through work on the discrete *Earth Mover's Distance* (EMD). We provide a new formula for the mean EMD by computing three different formulas for the sum of width–one matrices: the first two formulas apply the theory of *abstract simplicial complexes* and result from a shelling of the order complex, whereas the last formula uses Young tableaux. Subsequently, we employ this result to compute the EMD under different cost matrices satisfying the *Monge property.* Additionally, we use linear programming to compute the EMD under non-Monge cost matrices, giving an interpretation of the EMD as a distance measure on pie charts. Furthermore, we generalize our result to the n-dimensional EMD, by providing two different formulas for the sum of width–one tensors: once approaching the problem from the perspective of Young tableaux and once through the theory of abstract simplicial complexes by shelling of the n-dimensional order complex.

In the second part, we provide contributions to the topic of *parking functions.* We provide background on the topic and show a connection to the first part of this work through certain statistics on parking functions used in the shuffle conjecture. Furthermore, we provide enumerative formulas for different generalizations of parking functions, allowing cars to have varying lengths. Additionally, we show a surprising connection between certain restricted parking objects and the *Quicksort* algorithm. At last, we will use the intersection of a subset of parking functions and *Fubini rankings* to characterize and enumerate *Boolean algebras* in the weak Bruhat order of S_n.

To my friends and my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

ACKNOWLEDGMENTS

I will always be grateful to my two advisors, Jeb Willenbring and Pamela Harris. Jeb has always given me interesting problems to work on while providing me with literature, further explanations and examples needed for my understanding. He made each of our meetings fun and productive, and he gave me the freedom and support I needed to be successful in graduate school while also preparing for a non–academic job.

Pamela has taught me how to do mathematical research. She has an incredible ability to come up with questions, paired with an admirable generosity in sharing her thoughts and ideas. We have collaborated on several research projects, some of which are shown in this work. Working with Pamela is always greatly enjoyable, and there are few people who had a similarly large impact on my life as a mathematician.

I want to thank Will Erickson, who I deeply enjoy collaborating with and who is one of the best mathematicians I know. Some of our joint projects are shown in this work.

To the members of my defense committee: Allen Bell, David Spade and Richard Stockbridge — thank you for reading my work and serving on my committee.

Finally, I want to thank several people who impacted me or collaborated with me: Jennifer Elder, Carlos Martínez Mori and Kim Harry.

1 INTRODUCTION

This work is separated into two parts, which are unrelated at first glance. In the first part, we will provide contributions to the combinatorics of optimal transport, particularly the *Earth Mover's Distance* (EMD). We recall the original transportation problem posed by Gaspard Monge in the 18th century. The EMD, which is the solution to said problem, provides a measure for a distance between ordered pairs of histograms. It has received much research attention from various academic disciplines, for example computer vision [60], particle physics [50] and viral outbreak investigations [55]. Particularly recently, it was used to examine grade distributions [10, 52] — an application that also provided motivation for the first part of this work. In Chapter 3, we provide several new, efficient formulas to compute the average EMD over all pairs of histograms of a fixed size. The results of Chapter 3 were published in [28]. Moreover, in Chapter 4, we apply the results from Chapter 3 to examine the EMD under a more general underlying cost function. Furthermore, in Chapter 5, we generalize the results from Chapter 3 to higher dimensions and provide new formulas to compute the average EMD over all tuples of histograms of a fixed size. The results of Chapter 5 were published in [27].

The second part of this work provides contributions to the topic of *parking functions*. Parking functions, after being introduced by Konheim and Weiss [51] in 1966, have been found to be in bijection with (or equinumerous to) several different combinatorial objects, such as rooted forest inversions, labeled Dyck paths and noncrossing partitions (we refer to [67, 75] for a comprehensive overview). In Chapter 6, we provide several interesting background results on parking functions. Specifically, we present a connection between certain statistics on parking functions and the EMD. We show that the *area*–statistic can be expressed in terms of the EMD, which can in turn be applied to results found for said statistic. The results of Chapter 7 were published in [30]. Next, in Chapter 7, we introduce first generalizations of parking functions, allowing cars to have different lengths. We present enumerative counts for the generalizations by counting the number of ways to achieve a certain parking order. Furthermore, in Chapter 8, we reveal a surprising connection between certain parking objects and the *Quicksort* algorithm. Some of the results shown in Chapter 8 are to appear in the American Mathematical Monthly and currently available as preprint [41]. Additionally, we present further counts for parking objects with different constraints similarly to those in Chapter 7, by counting the preferences leading to each parking order. Finally, in Chapter 9, we give a characterization and count for the set of *Boolean intervals* in the weak order lattice of the symmetric group, by giving a bijection between Boolean intervals and the intersection of a subset of parking functions with the set of possible rank ings in competitions, where ties are allowed. Chapter 9 is currently under review and available as preprint [21].

Part I

Optimal Transport

2 THE MONGE PROBLEM

In this chapter, we provide background on the discrete *Earth Mover's Distance* (EMD) (also known as *Wasserstein distance*) and transport theory through the *Monge problem*, which was originally posed in 1781 by Gaspard Monge [56]. Additionally, we will briefly survey related results and motivate the remainder of the first part of this work.

2.1 The Earth Mover's Distance

Throughout, for any $x \in \mathbb{N}_{>0}$, we will denote by $[x] = \{1, \ldots, x\}$. Suppose, we have n units of earth stored in k different silos, with each silo labeled $s_1, \ldots, s_k$. We can describe the distribution of earth throughout the silos via $\lambda : [k] \to \{0, 1, \ldots, n\}$, with silo s_i containing $\lambda(i)$ units of earth and $\sum_{i=1}^{n} \lambda(i) = n$. For some reason, the current distribution λ is no longer viable and we are tasked with moving earth from silo to silo, so that afterwards, each silo s_i contains $\mu(i)$ units of earth, where $\mu : [k] \to \{0, 1, \ldots, n\}$ (and of course $\sum_{i=1}^{n} \mu(i) = n$). When we have moved earth until we have successfully achieved distribution μ, all moves we made can be recorded in a *transport plan*, denoted $T \in \Sigma_{\boldsymbol{\lambda},\boldsymbol{\mu}}$, where $\Sigma_{\boldsymbol{\lambda},\boldsymbol{\mu}}$ is the set of all matrices with row sums $\boldsymbol{\lambda} = (\lambda(1), \ldots, \lambda(n))$ and column sums $\boldsymbol{\mu} = (\mu(1), \ldots, \mu(n))$. There are many different transport plans we can follow to achieve the desired outcome, however, some transport plans are more advantageuous than others. Whenever we move earth from s_i to s_j, we incur a "cost". Considering s_i and s_j as being $|i - j|$ distance units apart, we assign a cost of $|i - j|$. Hence, the cost of executing all moves prescribed in a transport plan T amounts to $\sum_{i,j=1}^{n} C_{ij} T_{ij}$, with the standard *cost matrix* defined as

$$C = \begin{bmatrix} 0 & 1 & 2 & \ldots & n-1 \\ 1 & 0 & 1 & \ldots & n-2 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ n-1 & n-2 & n-3 & \ldots & 0 \end{bmatrix}. \tag{2.1}$$

The Monge problem asks for the cost associated with the optimal (cheapest) transport plan, which we denote as T^*. This can be stated as the linear programming problem:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{i,j=1}^{n} C_{ij} T_{ij}, & \\
\text{subject to} \quad & T_{ij} \geq 0 & \text{for all } 1 \leq i, j \leq n, \\
\text{and} \quad & \sum_{j=1}^{n} T_{ij} = \lambda(i) & \text{for each } 1 \leq i \leq n, \\
\text{and} \quad & \sum_{i=1}^{n} T_{ij} = \mu(j) & \text{for each } 1 \leq j \leq n.
\end{aligned}$$

The EMD presents itself as the solution to the above linear programming problem, and hence as the solution to the Monge problem.

In a more formal setting, the EMD can be viewed as a metric on histograms with a grand total s and n bins. Consider $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)$ and $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_n)$ with $\sum \boldsymbol{\mu} = \sum \boldsymbol{\nu} = s$. Then the EMD can be realized as the infimum of matrix products

$$\text{EMD}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \inf_{F \in \Sigma_{\boldsymbol{\mu},\boldsymbol{\nu}}} \text{trace}(C^T F),$$

where C the standard cost matrix from (2.1). As one of the major points of focus for this work, the matrix C satisfies several important properties. For one, it intuitively satisfies the triangle inequality:

$$c_{ij} \leq c_{ik} + c_{kj}, \text{ for all } i, j, k. \tag{2.2}$$

In addition, the matrix C has the *Monge property*:

$$c_{ij} + c_{i'j'} \leq c_{ij'} + c_{i'j}, \text{ for all } i < i', j < j'. \tag{2.3}$$

Matrices which satisfy (2.3) are also referred to as *Monge matrices*.

In Chapter 4, we examine the Monge property further and use different Monge matrices as cost for the EMD. For us, the most important use of the Monge property is a simplification in the computation of the EMD: for any cost matrix C satisfying the Monge property, we have

$$\text{EMD}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{F \in \mathcal{T}^1_{\boldsymbol{\mu},\boldsymbol{\nu}}} \text{trace}(C^T F),$$

where $\mathcal{T}^1_{\boldsymbol{\mu},\boldsymbol{\nu}}$ is the set of width-one matrices with row sums $\boldsymbol{\mu}$ and column sums $\boldsymbol{\nu}$, respectively. A focus of this work, as well as prior related work is the computation of the mean EMD over all ordered pairs of histograms with d bins and grand total s. To facilitate notation, we use the following definitions.

Definition 2.1. For $s, n \in \mathbb{N}_{>0}$, we refer to *weak integer compsitions* of s into n parts as

$$\mathcal{C}(s,n) = \left\{ \lambda \in \mathbb{N}^n \;\middle|\; \sum_{i=1}^{n} \lambda_i = s \right\}.$$

The number of weak compositions for fixed s and d is well–known to be

$$\#\mathcal{C}(s,n) = \binom{s+n-1}{s} = \binom{s+n-1}{n-1}. \tag{2.4}$$

Note, that a histogram with grand total s and d bins is an element $\lambda \in \mathcal{C}(s,d)$.

We additionally make use of the following referring to the mean EMD.

Definition 2.2. For $s \in \mathbb{N}_{>0}$ and $\boldsymbol{n} = (n_1, n_2) \in \mathbb{N}^2_{>0}$, we refer to the mean EMD taken over all $(\lambda, \mu) \in \mathcal{C}(s, n_1) \times \mathcal{C}(s, n_2)$ as

$$\overline{\mathrm{EMD}}_{\boldsymbol{n},s} = \frac{\sum_{(\lambda,\mu)} \mathrm{EMD}(\lambda,\mu)}{\#\left(\mathcal{C}(s,n_1) \times \mathcal{C}(s,n_2)\right)}.$$

In the following section, we provide an overview of results relating to the mean EMD.

2.2 Related work

We begin by summarizing the work of Bourn and Willenbring, who were able to find the mean EMD under the cost matrix $C_{ij} = |i-j|$ by constructing a generating function. For $s \in \mathbb{N}_{>0}$, $\boldsymbol{n} = (n_1, n_2) \in \mathbb{N}^2_{>0}$ and ordered pairs of histograms $(\boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathcal{C}(s, n_1) \times \mathcal{C}(s, n_2)$, they defined

$$H_{\boldsymbol{n}}(z,t) = \sum_{s=0}^{\infty} \left(\sum_{(\boldsymbol{\mu},\boldsymbol{\nu}) \in \mathcal{C}(s,n_1) \times \mathcal{C}(s,n_2)} z^{\mathrm{EMD}(\boldsymbol{\mu},\boldsymbol{\nu})} \right) t^s, \tag{2.5}$$

which encodes the number of ordered histogram pairs with EMD d in coefficients of $t^s z^d$. To compute $H_{\boldsymbol{n}}$, they provide the following recursive formula [10, Th. 3]

$$H_{\boldsymbol{n}}(z,t) = \frac{H_{n_1-1,n_2}(z,t) + H_{n_1,n_2-1}(z,t) - H_{n_1-1,n_2-1}(z,t)}{1 - z^{|n_1-n_2|}t}. \tag{2.6}$$

For histograms with total s and $\boldsymbol{n} = (n_1, n_2)$ bins, the expected EMD is then

$$\overline{\text{EMD}}_{\boldsymbol{n},s} = \frac{\left([t^d]\frac{\partial}{\partial z} H_{n_1,n_2}(z,t)\right)|_{z=1}}{([t^d]H_{n_1,n_2}(z,t))|_{z=1}}. \tag{2.7}$$

Bourn and Willenbring used the EMD to examine grade distributions – histograms with total s and d bins correspond to grade distributions of classes with s students who can each receive one of d grades.

In [52], we extended the generating function in (2.5) to include information about a *weighted total* $\mathcal{A}$ – for a histogram λ with n bins, its weighted total amounts to $\mathcal{A} = \sum_{i=1}^{n} i\lambda(i)$. To achieve this, two additional indeterminates were introduced to the generating function, yielding

$$H_{\boldsymbol{n}}(z,t,g_1,g_2) = \sum_{s=0}^{\infty} \left(\sum_{(\mu,\nu)\in\mathcal{C}(s,n_1)\times\mathcal{C}(s,n_2)} g_1^{\mathcal{A}(\mu)} g_2^{\mathcal{A}(\nu)} z^{\text{EMD}(\mu,\nu)} \right) t^s,$$

which additionally encodes the weighted total in exponents of g_1 and g_2, allowing for more specific comparison of histograms that satisfy a constraint on the weighted total. To compute $H_{\boldsymbol{n}}$, we provided the recursive formula [52, Theorem 3.2] (writing $H_{\boldsymbol{n}}(z,t,g_1,g_2)$ as $H_{\boldsymbol{n}}$ to simplify notation)

$$H_{\boldsymbol{n}}(z,t,g_1,g_2) = \frac{H_{n_1-1,n_2} + H_{n_1,n_2-1} - H_{n_1-1,n_2-1}}{1 - z^{|n_1-n_2|}t g_1^{n_1-1} g_2^{n_2-1}}. \tag{2.8}$$

As shown in [52], the parameters g_1 and g_2 record the weighted total $\mathcal{A}$ of the compared histograms. This extension can be used to apply the EMD to the context of grade distributions, comparing a more particular set of distributions narrowed down by their *Grade Point Average* (GPA). When computing the expected EMD using equation (2.8), we can now find the GPA encoded in the parameters g_1 and g_2 of the resulting generating function.

A further generalization of the result obtained by Bourn and Willenbring in [10] was given by Erickson in 2021, who generalized (2.6) to the d-dimensional EMD [24]. In similar fashion to [10], where width-one matrices were essential to the computation of the 2-dimensional EMD, Erickson was able to tie the computation of the d-dimensional EMD to width-one tensors. Let $d \in \mathbb{N}_{>0}$, $\boldsymbol{n} = (n_1, \ldots, n_d)$ and define the generating function [24, Section 5]

$$H_{\boldsymbol{n}}(z,t) = \sum_{s=0}^{\infty} \left(\sum_{\boldsymbol{\mu} \in \mathcal{C}(s,n_1) \times \cdots \times \mathcal{C}(s,n_d)} z^{\mathrm{EMD}_d(\boldsymbol{\mu})} \right) t^s,$$

where EMD_d is the generalized EMD to d dimensions: let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)$ with each $\mu_i \in \mathcal{C}(s,n)$, define as $\widehat{C}$ is the generalized cost function

$$\widehat{C}(m_1, \ldots, m_d) = \min_{i \in [d]} \left\{ \sum_{i \neq j} |m_i - m_j| \right\},$$

then the d-dimensional EMD is defined as

$$\mathrm{EMD}_d(\boldsymbol{\mu}) = \sum_{m \in [n]^d} \widehat{C}(m) J_{\boldsymbol{\mu}}(m),$$

with $J_{\boldsymbol{\mu}}$ a unique d-dimensional tensor whose coordinate hyperplane—sums agree with $\boldsymbol{\mu}$, obtained through an application of the *Robinson–Schenstedt–Knuth* correspondence (RSK). Erickson also provided a similar recursive formula to compute values of $H_{\boldsymbol{n}}$:

$$H_{\boldsymbol{n}} = \frac{\sum_A (-1)^{|A|-1} H_{\boldsymbol{n}-e(A)}}{1 - z^{\widehat{C}(\boldsymbol{n})} t}, \tag{2.9}$$

where $A \subseteq [d]$ and $e(A)$ an "indicator vector", whose ith component is 1 if $i \in A$ and 0 otherwise. This d-dimensional generalization of the EMD was the motivation for Chapter 5, where we generalize the result for 2 dimensions found in Chapter 3 to higher dimensions.

In yet another fairly recent result, Frohmader and Volkmer [31] approached the problem from the perspective of calculus – they define the EMD as a random variable on the probability simplex $\mathcal{P}_n \times \mathcal{P}_n$ and consider the expected value over all probability distributions.

Using this method, they managed to close the recursion found by Bourn and Willenbring in [10, Theorem 1]:

$$\overline{\mathrm{EMD}}_{(n,n),1} = \frac{2^{2n-3}(n-1)!^2}{(2n-1)!}(n-1)!^2.$$

Additionally, they provided a closed form for higher moments.

In the remainder of the first part of this dissertation, we provide more contributions to the combinatorics of the EMD. We begin by reducing the problem of finding the mean EMD to the problem of finding the sum of all $n_1 \times n_2$ matrices with support on a chain. This allows us to apply the theory of *abstract simplicial complexes*, where ultimately a *shelling* yields the desired result, see Theorem 3.20 in Chapter 3. We then use this result in Chapter 4 to examine the EMD under different cost matrices, showing implementations of our approaches in *Sage* (Python) and comparing experimental data. Finally, in Chapter 5 we generalize our results – much like Erickson in [24] – to higher dimensional tensors, with the result following from an application of *multiset Eulerian polynomials*.

3 THE SUM OF ALL WIDTH–ONE MATRICES

The results found in the following chapter have been published in collaborative work with William Q. Erickson in the *European Journal of Combinatorics*, see [28]. We additionally provide results that are contained in the preprint, see [26]. In this chapter, we give an alternative approach to finding the expected EMD for ordered pairs of compositions of s into n parts. The idea builds on the linearity of the matrix trace: the EMD is the sum of traces of matrix products, which can be simplified by taking the trace of the product of the cost matrix with a sum of width-one matrices. Therefore, the problem of finding the expected EMD can be reduced to the problem of finding the sum of all width-one matrices. We provide three different formulas for this result – the first formula, found in Corollary 3.19, while compact, is the least computationally efficient. The second and third formulas, found in Theorems 3.20 and 3.23 respectively, complement each other: Theorem 3.23 is more efficient under growing matrix dimensions, whereas the runtime of Theorem 3.20 is less affected by growing matrix grand total. We show a comparison in Section 3.7.

3.1 Background and motivation

Throughout this chapter, we let $\boldsymbol{n} = (n_1, n_2) \in \mathbb{N}^2_{>0}$ and $\mathcal{T}^1_{\boldsymbol{n},s}$ be the set of all $n_1 \times n_2$ matrices consisting of nonnegative integer entries summing to $s \in \mathbb{N}_{>0}$, such that all nonzero entries lie on a single path consisting of *south* and *east* steps (or equivalently, of steps in the directions of the standard basis vectors e_1 and e_2). The goal of this chapter is to find the sum of all such matrices, which we denote by $\Sigma^1_{\boldsymbol{n},s}$.

As mentioned in Chapter 1, one goal of this work is to examine the EMD under different cost matrices. This problem can be solved by using the linearity of the matrix *trace*:

Proposition 3.1. *Let C be any $n_1 \times n_2$ matrix with the Monge property. Then, the mean*

EMD on $\mathcal{C}(s,n_1)\times\mathcal{C}(s,n_2)$ is

$$\frac{\operatorname{trace}\left(C^{\mathrm{T}}\cdot\Sigma^1_{\boldsymbol{n},s}\right)}{\binom{s+n_1-1}{n_1}\binom{s+n_2-1}{n_2}}$$

Proof. We know that, for $(\lambda,\mu)\in\mathcal{C}(s,n_1)\times\mathcal{C}(s,n_2)$:

$$\operatorname{EMD}(\lambda,\mu)=\operatorname{trace}\left(C^{\mathrm{T}}T_{\lambda,\mu}\right),$$

where $T_{\lambda,\mu}\in\mathcal{T}^1_{\boldsymbol{n},s}$ and has row sums λ and columns sums μ. Summing over all compositions $(\lambda,\mu)\in\mathcal{C}(s,n_1)\times\mathcal{C}(s,n_2)$, we get

$$\begin{aligned}\sum_{(\lambda,\mu)}\operatorname{EMD}(\lambda,\mu)&=\sum_{T\in\mathcal{T}^1_{\boldsymbol{n},s}}\operatorname{trace}\left(C^{\mathrm{T}}T\right)\\&=\operatorname{trace}\left(C^{\mathrm{T}}\cdot\sum_{T\in\mathcal{T}^1_{\boldsymbol{n},s}}T\right)\\&=\operatorname{trace}\left(C^{\mathrm{T}}\cdot\Sigma^1_{\boldsymbol{n},s}\right).\end{aligned}$$

Dividing this result by the number of ordered pairs of compositions, $\#(\mathcal{C}(s,n_1)\times\mathcal{C}(s,n_2))=\#\mathcal{C}(s,n_1)\cdot\#\mathcal{C}(s,n_2)=\binom{s+n_1-1}{n_1}\binom{s+n_2-1}{n_2}$, yields the mean EMD. □

This simple "change of perspective" allows us to compute the mean EMD through the computation of a single matrix product and trace, provided we have a formula for $\Sigma^1_{\boldsymbol{n},s}$. Furthermore, this simplified computation allows us to easily exchange the cost matrix C used in the computation of the EMD – as stated in Proposition 3.1, we are now able to compute the mean EMD under any cost matrix satisfying the Monge property.

3.2 Computing $\Sigma^1_{\boldsymbol{n},s}$

Several approaches were found to compute the desired value $\Sigma^1_{\boldsymbol{n},s}$. We dedicate the remainder of this chapter to discuss these different results.

3.2.1 Number of chains and number of compositions

Let us restrict $\Sigma^1_{\boldsymbol{n},s}$ to its individual entries – we refer to the (i,j)th entry as $\Sigma^1_{\boldsymbol{n},s}(i,j)$, for $1 \le i \le n_1$ and $1 \le j \le n_2$. In this section, we will construct a first formula for $\Sigma^1_{\boldsymbol{n},s}(i,j)$ based on the number of *chains* containing (i,j). We begin by recalling necessary definitions. Let $\preceq$ be the product order, i.e.

$$(i,j) \preceq (i',j') \iff i \le i' \text{ and } j \le j', \tag{3.1}$$

then the set $[n_1] \times [n_2]$ together with $\preceq$ forms a *partially–ordered set* (poset), denoted as $([n_1] \times [n_2], \preceq)$. Analogously, we use $\prec$ for strict inequality, i.e.

$$(i,j) \prec (i',j') \iff i \le i' \text{ and } j \le j' \text{ but not } (i,j) = (i',j'), \tag{3.2}$$

as well as $\succeq$ and $\succ$ for the respective opposites. We recall as a *chain* $S \subseteq ([n_1] \times [n_2], \preceq)$ any subset of pairwise comparable elements

$$S = s_1 \preceq s_2 \preceq \cdots \preceq s_{\#S}.$$

An antichain is conversely a subset of pairwise *in*comparable elements. For any chain S, we refer to its cardinality $\#S$ as its *length*, and we will refer to chains of length ℓ as ℓ-chains. We use $(i,j) \preceq S$ or $(i,j) \prec S$ to indicate that $(i,j) \preceq (s_1,s_2)$ or $(i,j) \prec (s_1,s_2)$ for all $(s_1,s_2) \in S$.

Additionally, we recall that the *support* of a matrix is the index set of its nonzero entries, i.e. if $M \in \mathbb{N}^{n_1 \times n_2}_{>0}$, then the support of M is

$$\sup(M) = \{(i,j) \mid M_{ij} \neq 0\}.$$

Thus, a width–one matrix is a matrix whose support is a chain in the poset $([x] \times [y], \preceq)$. Equivalently, the support of a width–one matrix contains no antichain with length greater than 1.

The key to the first result lies in the following definition, which allows to reduce the problem of finding $\Sigma^1_{\boldsymbol{n},s}$ to the problem of finding chains in the poset $([n_1] \times [n_2], \preceq)$.

Definition 3.2. For $\boldsymbol{n} = (n_1, n_2)$, we define as $\mathfrak{C}(i, j, \ell)$ the number of ℓ-chains in the poset $([n_1] \times [n_2], \preceq)$ that contain (i, j).

Proposition 3.3. *For $\boldsymbol{n} = (n_1, n_2)$, the (i, j)th entry of $\Sigma^1_{\boldsymbol{n},s}$ is*

$$\Sigma^1_{\boldsymbol{n},s}(i,j) = \sum_{\ell=1}^{n_1+n_2-1} \binom{s}{\ell} \mathfrak{C}(i,j,\ell). \tag{3.3}$$

Proof. For any index (i, j), with $1 \leq i \leq n_1$ and $1 \leq j \leq n_2$, there are $\mathfrak{C}(i, j, \ell) \cdot \binom{s-1}{\ell-1}$ different matrices $T \in \mathcal{T}^1_{\boldsymbol{n},s}$ contributing to $\Sigma^1_{\boldsymbol{n},s}(i, j)$, with each ℓ–chain occuring exactly $\binom{s-1}{\ell-1}$ times. On average, each such matrix contributes $\frac{s}{\ell}$ to $\Sigma^1_{\boldsymbol{n},s}(i, j)$. The binomial identity $\binom{s-1}{\ell-1} \frac{s}{l} = \binom{s}{\ell}$ yields the desired result. □

Proposition 3.3 allows for the computation of $\Sigma^1_{\boldsymbol{n},s}$ provided we have a formula for $\mathfrak{C}(i, j, \ell)$.

3.2.2 Computing $\mathfrak{C}(i, j, \ell)$

The second part of the problem is to find the value $\mathfrak{C}(i, j, \ell)$. In a collaboration with Erickson, we were able to find an improved solution to this problem through the theory of abstract simplicial complexes, see [28], which is described in more detail in Section 3.4. First, we present an altogether different approach, which relies on a recursive formula. For this, we need to define three additional combinatorial objects counting chains.

Definition 3.4. Let

1. $\mathfrak{C}_{\prec}(i, j, \ell)$ be the number of ℓ-chains strictly northwest of (i, j), i.e. the number of ℓ-chains S such that $S \prec (i, j)$;

2. $\mathfrak{C}_{\succ}(i, j, \ell)$ be the number of ℓ-chains strictly southeast of (i, j), i.e. the number of ℓ-chains S such that $(i, j) \prec S$ and

3. $\mathfrak{C}_{*}(i, j, \ell)$ be the number of ℓ-chains that can be partitioned into $S' \cup S''$ such that $S' \prec (i, j) \prec S''$ (with $\#S' + \#S'' = \ell$).

Partitioning the number of chains into these disjoint subsets, we can compute $\mathfrak{C}(i,j,\ell)$ as follows.

Theorem 3.5. *The number of ℓ-chains containing (i,j) is*

$$\mathfrak{C}(i,j,\ell) = \mathfrak{C}_{\succ}(i,j,\ell-1) + \mathfrak{C}_{*}(i,j,\ell-1) + \mathfrak{C}_{\prec}(i,j,\ell-1).$$

Proof. Any chain S containing index (i,j) can be partitioned into the disjoint subsets $S' \cup \{(i,j)\} \cup S''$. Then S' and S'' must satisfy $\#S' + \#S'' = \#S - 1$. We can partition the set of chains containing (i,j) into the three disjoint sets described in the above definitions: chains S' that are strictly northeast of (i,j), chains S'' that are strictly southwest of (i,j) and all chains $S' \cup S''$. The union of these disjoint subsets yields the total number of ℓ-chains containing (i,j), therefore summing the cardinalities yields the result. □

Next, we provide enumerative formulas for the disjoint parts that make up Theorem 3.5.

Lemma 3.6. *We can compute $\mathfrak{C}_{\prec}(i,j,\ell)$ as*

$$\mathfrak{C}_{\prec}(i,j,\ell) = \sum_{r=1}^{i}\sum_{c=1}^{j} \mathbb{1}_{(r,c)\neq(i,j)} \mathfrak{C}_{\prec}(r,c,\ell-1) \tag{3.4}$$

with $\mathfrak{C}_{\prec}(i,j,\ell) = 0$ if $\ell < 0, i < 1$ or $j < 1$ and $\mathfrak{C}_{\prec}(i,j,\ell) = 1$ if $\ell = 0$.

Proof. Every chain C ending at $(i',j') \prec (i,j)$ can be expressed as $C' \cup (i',j')$, for C' a chain ending northeast of (i',j'). Recursing on all possible chains C' yields the desired result. □

Lemma 3.7. *We can compute $\mathfrak{C}_{\succ}(i,j,\ell)$ as*

$$\mathfrak{C}_{\succ}(i,j,\ell) = \sum_{r=i}^{n_1}\sum_{c=j}^{n_2} \mathbb{1}_{(r,c)\neq(i,j)} \mathfrak{C}_{\succ}(r,c,\ell-1) \tag{3.5}$$

with $\mathfrak{C}_{\succ}(i,j,\ell) = 0$ if $\ell < 0, i < 1$ or $j < 1$ and $\mathfrak{C}_{\succ}(i,j,\ell) = 1$ if $\ell = 0$.

Proof. The argument functions analogously to that of Lemma 3.6. □

Lemma 3.8. *The value of $\mathfrak{C}_{*}(i,j,\ell)$ satisfies*

$$\mathfrak{C}_{*}(i,j,\ell) = \sum_{\ell'=1}^{\ell-1} \mathfrak{C}_{\prec}(i,j,\ell') \cdot \mathfrak{C}_{\succ}(i,j,\ell-\ell'-1). \tag{3.6}$$

Proof. Following the argument given in the proof of Theorem 3.5, we are counting chains that we can partition into $S' \prec (i,j) \prec S''$. Since the union of any such chains $S' \cup S''$ is again a chain, the result follows from summing over all possible different lengths for S' and S''. □

This recursive expression for $\mathfrak{C}(i,j,\ell)$ completes our formula for $\Sigma^1_{\boldsymbol{n},s}$. However, this approach can be simplified when approached from the perspective of *abstract simplicial complexes*. As we see in the next section (and in [26]), the theory of simplicial complexes allows us to avoid the above recursion.

3.3 Abstract simplicial complexes

We start by giving the definition of abstract simplicial complexes. This exposition is standard and follows the works [12, 44].

Definition 3.9. Let Δ be a collection of sets such that

- $\emptyset \in \Delta$, and
- for all $\tau \in \Delta$ and $\delta \subseteq \tau$, $\delta \in \Delta$.

Then we call Δ an abstract simplicial complex.

In other words, an abstract simplicial complex is a collection of sets that is closed under subsets. Elements of a complex are referred to as *faces*. The *dimension* of a face is one less than its cardinality, and the dimension of a complex is the maximum of the dimension of its faces. Zero-dimensional faces are referred to as *vertices*, one-dimensional faces are called *edges*. A maximal face, i.e. a face that is not the subset of another face, is called a *facet*.

Example 3.10. A standard example is the set of vertices, edges and faces of an *octahedron*, see Figure 1. The conditions that make the octahedron an abstract simplicial complex can easily be observed – any face consists of three edges, each which in turn are made up of a pair of vertices.

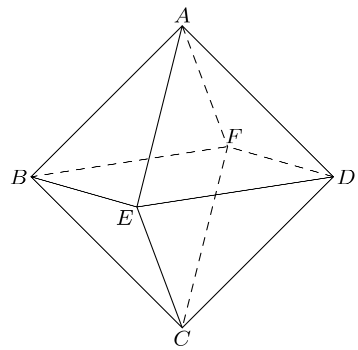


Figure 1: The octahedron.

The following definitions allow for characterizations of abstract simplicial complexes.

Definition 3.11 (f-vector)**.** Let Δ be an abstract simplicial complex of dimension $n-1$. Then, we denote by f_i the number of faces of Δ that have dimension i. The *f-vector* is:

$$f = (f_{-1}, f_0, f_1, \dots, f_{n-1}),$$

where $f_{-1} = 1$ because $\emptyset \in \Delta$.

Remark. Sometimes, it is convenient to speak in terms of the *f-polynomial*, defined as

$$f_\Delta(t) = \sum_{i=0}^{n} f_{i-1} t^i.$$

Definition 3.12 (h-polynomial)**.** The *h-polynomial* of a simplicial complex Δ is defined as

$$h_\Delta(t) \coloneqq \sum_{i=0}^{n} h_i t^i = \sum_{i=0}^{n} f_{i-1} t^i (1-t)^{n-i}. \tag{3.7}$$

The h-vector is obtained by simply reading the coefficients of the h-polynomial:

$$h = ([t^0]h_\Delta(t), [t^1]h_\Delta(t), \dots, [t^n]h_\Delta(t)).$$

Remark. The identity in equation (3.7) implies that we can also retrieve the f-vector from the h-vector through

$$f_{j-1} = \sum_{i=0}^{j} \binom{n-i}{j-i} h_i, \tag{3.8}$$

and the h-vector from the f-vector through

$$h_j = \sum_{i=0}^{j} (-1)^{j-i} \binom{n-i}{j-i} f_{i-1}.$$

For more details on abstract simplicial complexes, see [12, Ch. 5].

3.3.1 Shelling

An important property of an abstract simplicial complex is *shellability*. Shelling a simplicial complex provides an "alternative" way of obtaining the h-vector, and thus the f-vector. As we will see in Corollary 3.17, shelling ultimately allows us to efficiently find $\mathfrak{C}(i, j, \ell)$. We begin by stating this required definition.

Definition 3.13. Let Δ be a simplicial complex with facets $F_1, \ldots, F_n$. A complex Δ is called *pure*, if all its facets have the same dimension, i.e.

$$\dim F_1 = \cdots = \dim F_n.$$

Next, we give the definition and an example of *shelling*, following the exposition of [12, 26, 44].

Definition 3.14 (Shelling)**.** Let Δ be a pure simplicial complex with facets $F_1, \ldots, F_n$. A *shelling* of Δ is any ordering of its facets $F_1, \ldots, F_n$, such that each F_i contains a minimal element $\mathrm{R}(F_i)$ that is not contained in the simplicial complex generated by $F_1, \ldots, F_{i-1}$. A complex is called *shellable* if a shelling exists.

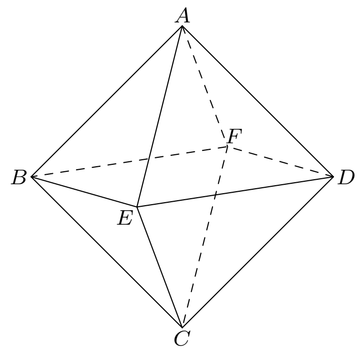
Fundamental to our work is the fact that we can recover the h-vector of a shellable complex Δ with faces $F_1, \ldots, F_n$ through

$$h_\ell = \#\{i \mid \#\mathrm{R}(\mathrm{F_i}) = \ell\}. \tag{3.9}$$

In other words, the ℓth element of the h-vector of a (shellable) simplicial complex counts the number of facets whose restrictions have size ℓ.

We illustrate the concept of shelling at the example of the octahedron, which is also stated in [26].

Example 3.15. Recall the abstract simplicial complex given by the boundary of an octahedron (vertices, edges and faces) as shown in Figure 1.



To show that this complex is shellable, it suffices to find an ordering $F_1, \ldots, F_8$ of its eight facets, such that each F_i contains a unique minimal element $\mathrm{R}(F_i)$ that is not contained in the subcomplex generated by $F_1, \ldots, F_{i-1}$. Note that when $i = 1$, this subcomplex is the empty set. Below, we exhibit a shelling of the boundary of the octahedron:

$F_1 = \{A, D, E\}$. In general, the restriction of F_1 is $\mathrm{R}(F_1) = \emptyset$.

$F_2 = \{A, D, F\}$. We have $\mathrm{R}(F_2) = \{F\}$.

$F_3 = \{C, D, E\}$. We have $\mathrm{R}(F_3) = \{C\}$.

$F_4 = \{A, B, E\}$. We have $\mathrm{R}(F_4) = \{B\}$.

$F_5 = \{A, B, F\}$. We have $\mathrm{R}(F_5) = \{A, F\}$.

$F_6 = \{B, C, E\}$. We have $\mathrm{R}(F_6) = \{B, C\}$.

$F_7 = \{C, D, F\}$. We have $\mathrm{R}(F_7) = \{F, C\}$.

$F_8 = \{B, C, F\}$. We have $\mathrm{R}(F_8) = \{B, C, F\}$.

By (3.9), this gives the h-vector $(1, 3, 3, 1)$. Applying (3.8), we retrieve the f-vector $(6, 12, 8)$. Indeed, the boundary of octahedron contains 6 vertices, 12 edges, and 8 faces (i.e. facets).

Next, we define the *order complex* and walk through the crucial process of shelling the order complex, which leads to the desired result $\mathfrak{C}(i, j, \ell)$.

3.3.2 The order complex Π_{n_1,n_2}

A specific simplicial complex that is crucial to our work is the *order complex*, which we define in this section. Let $\preceq$ be the product order defined in (3.1). Then, the *order complex* $\Pi_{\boldsymbol{n}}$ on $\boldsymbol{n} = (n_1, n_2)$ is defined as the set of all chains in the poset $([n_1] \times [n_2], \preceq)$. The facets of $\Pi_{\boldsymbol{n}}$ are then the *maximal chains* in $[n_1] \times [n_2]$, which all have dimension $n_1 + n_2 - 1$. As we see in the next section, the f-vector of $\Pi_{\boldsymbol{n}}$ will provide a formula for $\mathfrak{C}(i, j, \ell)$.

A shelling of the order complex

Let $\boldsymbol{n} = (n_1, n_2)$ and $\Pi_{\boldsymbol{n}}$ the order complex on $[n_1] \times [n_2]$. We recall, that facets of the order complex are maximal chains of the partially ordered set $([n_1] \times [n_2], \preceq)$, with $\preceq$ the product order defined in (3.1).

We can visualize a facet of $\Pi_{\boldsymbol{n}}$ in the relevant context of $x \times y$ matrices as seen in Figure 2.

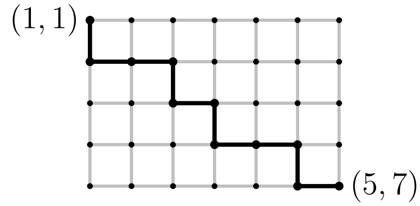


Figure 2: Example of a facet F of $\Pi_{5,7}$, or equivalently, a maximal chain in $([5] \times [7], \preceq)$.

From the fact that each maximal chain in the poset has the same length, we know that the complex $\Pi_{\boldsymbol{n}}$ is pure. To show that the complex is shellable, we have to find a shelling order satisfying (3.14). To accomplish this, we restrict each maximal chain to the set of its $\llcorner$-corners, as shown in Figure 3. For a facet F, we have

$$R(F) = \{(x, y) \in F \mid (x - 1, y) \in F \text{ and } (x, y + 1) \in F\}. \tag{3.10}$$

Following (3.9), we can obtain the h-vector of $\Pi_{\boldsymbol{n}}$ by counting the number of chains that contain k $\llcorner$-corners. To count the number of facets whose restricted size is k, we can see that in a maximal chain, we have exactly $n_1 - 1$ rows from which we can choose the "top"-part of the $\llcorner$-corner, and $n_2 - 1$ columns for the "right" end.

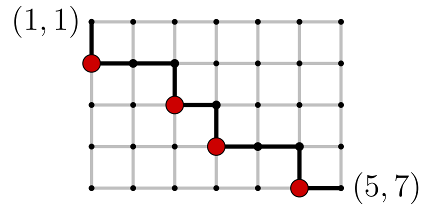


Figure 3: Example of a facet F of $\Pi_{5,7}$, or equivalently, a maximal chain in $([5] \times [7], \preceq)$. The restriction of F, R(F), is given by coordinates highlighted with red circles.

Proposition 3.16. *Let $\Pi_{\boldsymbol{n}}$ be the order complex with $\boldsymbol{n} = (n_1, n_2)$. Then*

$$h_k = \binom{n_1 - 1}{k}\binom{n_2 - 1}{k}$$

is the h-vector belonging to $\Pi_{\boldsymbol{n}}$.

Proof. Let F be a facet in $\Pi_{\boldsymbol{n}}$ such that $\#\mathrm{R(F)} = \ell$. That is, F is a maximal chain in $\Pi_{\boldsymbol{n}}$ with ℓ corners:

$$\mathrm{R}(F) = \{(a_1, b_1), \ldots, (a_\ell, b_\ell)\}, \qquad 2 \leq a_1 < \cdots < a_\ell \leq n_1 \text{ and } 1 \leq b_i < \cdots < b_\ell \leq n_2 - 1.$$

Therefore, we have $\binom{n_1-1}{\ell}$ choices for the a_i and $\binom{n_2-1}{\ell}$ choices for the b_i to determine a unique set R(F). The lemma then follows from (3.9). □

Corollary 3.17. *The f-vector of Π_n has entries*

$$f_\ell = \sum_{k=0}^{\ell} \binom{n_1 + n_2 - k - 1}{\ell - k}\binom{n_1 - 1}{k}\binom{n_2 - 1}{k},$$

and the f-polynomial is $f(t) = \sum_{i=0}^{n_1+n_2-1} f_i t^i$.

3.4 First explicit formula for $\Sigma^1_{\boldsymbol{n},s}$

To improve notation, we proceed as in [26] and define the polynomials:

$$g_{ij} = g_{ij}(t) = f_{i,j-1}(t) + f_{i-1,j}(t) - f_{i-1,j-1}(t) \tag{3.11}$$

and for fixed dimensions $\boldsymbol{n} = (n_1, n_2)$ we let

$$\overline{g_{ij}} = g_{n_1+i-1,n_2+j-1}. \tag{3.12}$$

Now, we are able to give a formula for $\mathfrak{C}(i, j, \ell)$, which we then use to provide a first explicit formula for $\Sigma^1_{\boldsymbol{n},s}$.

Theorem 3.18. *Let $\Pi_{\boldsymbol{n}}$ be the order complex with $\boldsymbol{n} = (n_1, n_2)$. Then, the number of ℓ-chains containing any index (i, j) are recorded as the coefficient*

$$\mathfrak{C}(i, j, \ell) = [t^{\ell-1}]g_{ij}\overline{g_{ij}}.$$

Proof. Any chain S containing index (i, j) can be split into the union of three distinct subsets:

$$S' \cup (i, j) \cup S'',$$

where S' is a chain strictly "north" and "west" of (i, j) and S'' is a chain strictly "south" and "east" of (i, j). The polynomial $f_{i-1,j}$ counts all chains north of (i, j). The polymial $f_{i,j-1}$ counts all chains west of (i, j). Adding those together, we are double counting all chains strictly northwest of (i, j), which we can omit by using inclusion-exclusion via

$$g_{ij} = f_{i,j-1} + f_{i-1,j} - f_{i-1,j-1}.$$

Analogously, the polynomial $\overline{g_{ij}}$ counts all chains south and east of (i, j). Every chain beginning in the northwest of (i, j) forms a valid chain through (i, j) with any chain in the southeast of (i, j), yielding $g_{ij}\overline{g_{ij}}$ options. Then the number of ℓ-chains containing (i, j) is the coefficient of $t^{\ell-1}$, since (i, j) itself is not accounted for in the length of chains computed by $g_{ij}\overline{g_{ij}}$. □

Now, having found a formula for $\mathfrak{C}(i, j, \ell)$, we are able to give a first explicit formula for $\Sigma^1_{\boldsymbol{n},s}$.

Corollary 3.19. *For $\boldsymbol{n} = (x, y)$, the (i, j)th entry of the sum of all width-one, nonnegative integer matrices with grand total s is given by*

$$\Sigma^1_{\boldsymbol{n},s}(i, j) = \sum_{\ell=1}^{n_1+n_2-1} \binom{s}{\ell} [t^{\ell-1}]g_{ij}\overline{g_{ij}}.$$

Proof. Follows from Proposition 3.3 and Theorem 3.18. □

3.5 Alternative approach through Stanley–Reisner theory

A further result we obtained in [26] was a different, more computationally viable, formula for the value $\Sigma^1_{\boldsymbol{n},s}$. In this section, the exposition follows [70]. Let K be a field, and $K[X] \coloneqq K[x_{ij} \mid (i,j) \in \Pi_{\boldsymbol{n}}]$ be the polynomial ring in variables $x_{i,j}$. For any monomial $m \in K[X]$, the *support* of m is the set of ordered pairs $(i,j) \in \Pi_{\boldsymbol{n}}$, such that x_{ij} divides m. Let I be the prime ideal containing all nonfaces of $\Pi_{\boldsymbol{n}}$, i.e. for the product order $\preceq$:

$$I \coloneqq \langle x_{i'j} x_{j'i} \mid i < i', j < j' \rangle.$$

Then, a basis for I is given by monomials who have width greater than 1. The *Stanley–Reisner* ring is the quotient

$$K[\Pi_{\boldsymbol{n}}] \coloneqq K[X]/I.$$

Clearly, $K[\Pi_{\boldsymbol{n}}]$ has a basis that consists of width-one monomials, and thus monomials whose support is a chain in $\Pi_{\boldsymbol{n}}$. Moreover, since I is generated by homogeneous monomials, the quotient $K[\Pi_{\boldsymbol{n}}]$ has a natural grading by degree. We denote by $K[\Pi_{\boldsymbol{n}}]_s$ the graded component consisting of homogeneous polynomials of degree s. Reading each matrix as the exponent matrix of a monomial in $K[\Pi_{\boldsymbol{n}}]_s$ leads us to a bijection

$$\mathcal{T}^1_{\boldsymbol{n},s} \leftrightarrow K\text{−basis of } K[\Pi_{\boldsymbol{n}}]$$

$$T \leftrightarrow \prod_{i,j} x_{ij}^{T_{ij}}.$$

This implies, that

$$\prod_{m \in K[\Pi_{\boldsymbol{n}}]_s} m = \prod_{(i,j)\in \Pi_n} x_{ij}^{\Sigma^1_{\boldsymbol{n},s}(i,j)}. \tag{3.13}$$

Recall the shelling found by restricting each facet F (maximal chain) in $\Pi_{\boldsymbol{n}}$ to the set of its $\llcorner$-corners, i.e. $\mathrm{R}(F) = \{(i,j) \mid (i-1,j) \in F \text{ and } (i,j+1) \in F\}$. This shelling induces a *Stanley decomposition* on $K[\Pi_{\boldsymbol{n}}]$:

$$K[\Pi_{\boldsymbol{n}}] = \bigoplus_F K[F] x_{\mathrm{R(F)}}, \tag{3.14}$$

with $K[F] = K[x_{ij} \mid (i,j) \in F]$ and $x_{\mathrm{R(F)}} = \prod_{(i,j)\in \mathrm{R(F)}} x_{ij}$. The crucial observation here is that each monomial of $K[\Pi_{\boldsymbol{n}}]$ lies in exactly one summand of (3.14). Since each component consists of homogeneous polynomials, we can further decompose into

$$K[\Pi_{\boldsymbol{n}}]_s = \bigoplus_{k} \bigoplus_{F:\#\mathrm{R}(F)=k} x_{\mathrm{R}(F)} K[\Pi_{\boldsymbol{n}}]_{s-k}. \tag{3.15}$$

3.5.1 Second formula for $\Sigma^1_{\boldsymbol{n},s}$

We present the second formula for $\Sigma^1_{\boldsymbol{n},s}$ through an application of Stanley–Reisner theory.

Theorem 3.20. *For $s, n_1, n_2 \in \mathbb{N}_{>0}$ and $n = (x, y)$, we have*

$$\Sigma^1_{\boldsymbol{n},s}(i,j) = \sum_{k=0}^{\min\{s,n_1,n_2\}-1} \binom{n_1+n_2+s-k-2}{n_1+n_2-1} \sum_{\ell=0}^{k} \binom{i-1}{\ell}\binom{j-1}{\ell}\binom{n_1-i}{k-\ell}\binom{n_2-j}{k-\ell}.$$

Analagous to [28], we facilitate the proof of Theorem 3.20 by listing several counting lemmas.

Lemma 3.21. *Let $(i,j) \in \Pi_{\boldsymbol{n}}$, and let $F \ni (i,j)$ be a facet of $\Pi_{\boldsymbol{n}}$. Then $\binom{n_1+n_2+s-k-2}{n_1+n_2-1}$ equals the exponent of x_{ij} in the product of all monomials in*

$$x_{ij} x_{\mathrm{R}(F)\setminus\{(i,j)\}}\, K[F]_{s-k-1}. \tag{3.16}$$

Proof. It suffices to show that

$$\binom{n_1+n_2+s-k-2}{n_1+n_2-1} = \Big(\#\text{ monomials in (3.16)}\Big)\Big(\text{avg. exponent of } x_{ij} \text{ in each monomial}\Big).$$

The number of monomials in (3.16) equals the number of monomials in $K[F]_{s-k-1}$, which is the number of weak compositions of the degree $s-k-1$ into $\#F$ many parts. Thus, recalling that $\#F = n_1+n_2-1$ for any facet F of $\Pi_{\boldsymbol{n}}$, and using the elementary formula (5.6), we have

$$\#\text{ monomials in (3.16)} = \binom{s-k-1+(n_1+n_2-1)-1}{(n_1+n_2-1)-1} = \binom{n_1+n_2+s-k-3}{n_1+n_2-2}. \tag{3.17}$$

The average exponent of x_{ij}, taken over all the monomials in $K[F]_{s-k-1}$, equals the degree $s-k-1$ divided by the number $\#F$ of variables. Adding 1 to this average to account for the factor of x_{ij} present in (3.16), we obtain

$$\text{average exponent of } x_{ij} \text{ in each monomial} = 1+\frac{s-k-1}{n_1+n_2-1} = \frac{n_1+n_2+s-k-2}{n_1+n_2-1}. \quad (3.18)$$

Multiplying the expressions in (3.17) and (3.18), we obtain

$$\binom{n_1+n_2+s-k-3}{n_1+n_2-2} \cdot \frac{n_1+n_2+s-k-2}{n_1+n_2-1} = \binom{n_1+n_2+s-k-2}{n_1+n_2-1},$$

as desired. □

Lemma 3.22. *Let* $(i,j) \in \Pi_{\boldsymbol{n}}$. *Then*

$$\sum_{\ell=0}^{k} \binom{i-1}{\ell}\binom{j-1}{\ell}\binom{n_1-i}{k-\ell}\binom{n_2-j}{k-\ell}$$

equals the number of facets $F \ni (i,j)$ *of* $\Pi_{\boldsymbol{n}}$ *such that* $\#\Big(\mathrm{R}(F) \setminus \{(i,j)\}\Big) = k$.

Proof. Every facet $F \ni (i,j)$ of $\Pi_{\boldsymbol{n}}$ is the union of two saturated chains

$$F' : (1,1) \preceq \cdots \preceq (i,j) \qquad \text{and} \qquad F'' : (i,j) \preceq \cdots \preceq (x,y),$$

which intersect only at (i,j). Clearly F' can be any facet of $\Pi_{(i,j)}$, viewed as a subposet of $\Pi_{\boldsymbol{n}}$. Likewise, F'' can be any facet of Π_{n_1-i+1,n_2-j+1}, viewed as a subposet of $\Pi_{\boldsymbol{n}}$ after translating coordinates. Since (i,j) is either the maximal or minimal element of these two subposets, it cannot occur as an element of $\mathrm{R}(F')$ or of $\mathrm{R}(F'')$. Therefore $\#(\mathrm{R}(F)\setminus\{(i,j)\}) = \mathrm{R}(F') + \mathrm{R}(F'')$. By (3.9), we thus have

$$\begin{aligned} h_{ij}(t)h_{n_1-i+1,n_2-j+1}(t) &= \left(\sum_{F'} t^{\#\mathrm{R}(F')}\right)\left(\sum_{F''} t^{\#\mathrm{R}(F'')}\right) \\ &= \sum_{F',F''} t^{\#\mathrm{R}(F')+\#\mathrm{R}(F'')} \\ &= \sum_{F\ni(i,j)} t^{\#(\mathrm{R}(F)\setminus\{(i,j)\})}, \end{aligned}$$

where the sums range over facets F, F', and F'' of $\Pi_{\boldsymbol{n}}$, $\Pi_{(i,j)}$, and Π_{n_1-i+1,n_2-j+1}, respectively. Therefore the number of facets described in the lemma equals the coefficient of t^k in the following product, which we expand via Lemma 3.16:

$$h_{ij}(t)h_{n_1+1-i,n_2+1-j}(t) = \left(\sum_{\ell=0}^{\min\{i-1,\,j-1\}} \binom{i-1}{\ell}\binom{j-1}{\ell}t^\ell\right)\left(\sum_{m=0}^{\min\{n_1-i,\,n_2-j\}} \binom{n_1-i}{m}\binom{n_2-j}{m}t^m\right).$$

The coefficient of t^k in this expansion equals $\sum_{\ell+m=k}\binom{i-1}{\ell}\binom{j-1}{\ell}\binom{n_1-i}{m}\binom{n_2-j}{m}$. Upon substituting $k-\ell$ for m, the proof is complete. □

Proof of Theorem 3.20. By (3.15) and (3.13), we know that $\Sigma^1_{\boldsymbol{n},s}(i,j)$ equals the exponent of x_{ij} in the product of all monomials in the graded component

$$K[\Pi_{\boldsymbol{n}}]_s = \bigoplus_{k=0}^{\min\{s,\,n_1-1,\,n_2-1\}} \bigoplus_{\substack{F:\\ \#\mathrm{R}(F)=k}} x_{\mathrm{R}(F)}K[F]_{s-k}, \tag{3.19}$$

where the inside sum ranges over the facets F of $\Pi_{\boldsymbol{n}}$. But the only monomials contributing to this exponent are those divisible by x_{ij}. Hence we may ignore all summands in (3.19) such that $(i,j)\notin F$. If $(i,j)\in F$, then the subspace of $K[F]_{s-k}$ spanned by the monomials divisible by x_{ij} is

$$x_{ij}\,K[F]_{s-k-1}.$$

Then since (i,j) may or may not lie in $\mathrm{R}(F)$, the subspace of $x_{\mathrm{R}(F)}K[F]_{s-k}$ spanned by monomials divisible by x_{ij} is

$$x_{ij}x_{\mathrm{R}(F)\setminus\{(i,j)\}}K[F]_{s-k-1}.$$

Combining this with (3.19), we conclude that $\Sigma^1_{\boldsymbol{n},s}(i,j)$ equals the exponent of x_{ij} in the product of all monomials in

$$\bigoplus_{k=0}^{\min\{s-1,\,n_1-1,\,n_2-1\}} \bigoplus_{\substack{F\ni(i,j):\\ \#(\mathrm{R}(F)\setminus\{(i,j)\})=k}} x_{ij}x_{\mathrm{R}(F)\setminus\{(i,j)\}}K[F]_{s-k-1}. \tag{3.20}$$

Applying Lemma 3.21 to (3.20), we see that the desired exponent of x_{ij} equals

$$\begin{aligned}\Sigma^1_{\boldsymbol{n},s}(i,j) &= \sum_{k=0}^{\min\{n_1,n_2,s\}-1} \sum_{\substack{F\ni(i,j):\\ \#(\mathrm{R}(F)\setminus\{(i,j)\})=k}} \binom{n_1+n_2+s-k-2}{n_1+n_2-1} \\ &= \sum_{k=0}^{\min\{n_1,n_2,s\}-1} \binom{n_1+n_2+s-k-2}{n_1+n_2-1} \cdot \#\Big\{F \ni (i,j) : \#(\mathrm{R}(F)\setminus\{(i,j)\}) = k\Big\},\end{aligned}$$

and we have already computed the second factor in Lemma 3.22. □

3.6 Third formula through tableaux and hypergeometric series

In this section, we present a third formula for $\Sigma^1_{\boldsymbol{n},s}$ which relies on Young tableaux and reveals an identity for the hypergeometric series ${}_4F_3$. As we see in Section 3.7, the different formulas we present complement each other well, as they are better suited for different contexts. We begin with an exposition on hypergeometric series, following [28] verbatim.

3.6.1 Hypergeometric series

We recall the (ordinary, or Gaussian) *hypergeometric series*

$${}_2F_1\left[\begin{matrix}a,b\\c\end{matrix};z\right] \coloneqq \sum_{k=0}^{\infty} \frac{(a)_k(b)_k}{(c)_k}\frac{z^k}{k!},$$

where $(a)_k \coloneqq a(a+1)\cdots(a+k-1)$ is the Pochhammer symbol for the rising factorial. In the special case $c = 1$, it is easy to see that

$${}_2F_1\left[\begin{matrix}a,b\\1\end{matrix};z\right] = \sum_{k=0}^{\infty} \frac{(a)_k}{k!}\frac{(b)_k}{k!} z^k = \sum_{k=0}^{\infty} \binom{a+k-1}{k}\binom{b+k-1}{k} z^k, \tag{3.21}$$

where the second equality holds only if $a, b \in \mathbb{N}$. The *generalized* hypergeometric series ${}_pF_q$ is defined analogously:

$${}_pF_q\left[\begin{matrix}a_1,\dots,a_p\\b_1,\dots,b_q\end{matrix};z\right] \coloneqq \sum_{k=0}^{\infty} \frac{(a_1)_k\cdots(a_p)_k}{(b_1)_k\cdots(b_q)_k}\frac{z^k}{k!}.$$

Since our interest in this section is purely combinatorial, we disregard issues of convergence and treat ${}_pF_q$ as a formal power series. Note that if some b_j is a nonpositive integer, then infinitely many of the coefficients are undefined. But if, for example, some a_i is also a nonpositive integer with $a_i \geq b_j$, then the series terminates before these undefined coefficients, and so the series as a whole is still defined (and is a polynomial in z).

The discovery of identities involving hypergeometric series is a longstanding, and yet still quite active, research area within combinatorics; see the books [23, Ch. II] and [58], for example. We will appeal to the following identity 4.3(14) in [23, p. 187], which expresses the convolution of two ordinary hypergeometric series:

$$ {}_2F_1\left[\begin{matrix}a, b\\ c\end{matrix}; \alpha z\right] {}_2F_1\left[\begin{matrix}a', b'\\ c'\end{matrix}; \beta z\right] = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{(\alpha z)^k}{k!} {}_4F_3\left[\begin{matrix}a',\ b',\ 1-k-c,\ -k\\ c',\ 1-k-a,\ 1-k-b\end{matrix}; \beta/\alpha\right]. $$

We will encounter the specialization where $c = c' = \alpha = \beta = 1$, which yields

$$ {}_2F_1\left[\begin{matrix}a, b\\ 1\end{matrix}; z\right] {}_2F_1\left[\begin{matrix}a', b'\\ 1\end{matrix}; z\right] = \sum_{k=0}^{\infty} \underbrace{\frac{(a)_k}{k!} \frac{(b)_k}{k!}}_{\substack{\binom{a+k-1}{k}\binom{b+k-1}{k} \\ \text{if } a, b \in \mathbb{N}}} {}_4F_3\left[\begin{matrix}a',\ b',\ -k,\ -k\\ 1,\ 1-k-a,\ 1-k-b\end{matrix}; 1\right] z^k. \tag{3.22} $$

3.6.2 Main result, third version

The *Robinson-Schenstedt-Knuth* correspondence, short RSK, provides a bijection between ordered pairs of semistandard Young tableaux and arrays with 2 rows, often referred to as *biwords*. For any nonnegative integer matrix T of dimensions $\boldsymbol{n}$, we can create a biword by listing in the first row the row-index of each positive entry in T and in the second row the column-index of each positive entry in T; listing the indices i and j in the biword T_{ij} consecutive times. Therefore, RSK provides a bijection between nonnegative integer matrices and pairs of semistandard Young tableaux. Following the construction of Knuth in [49], we see that the width of the matrix corresponds to the number of rows in both tableaux – therefore, every $T \in \mathcal{T}^1_{\boldsymbol{n},s}$ corresponds to a pair of one–row semistandard Young tableaux.

Theorem 3.23. *Let* $\boldsymbol{n} = (n_1, n_2)$*. Then, for all* $1 \leq i \leq x, 1 \leq j \leq y$*, the* (i, j) *entry of* $\Sigma^1_{\boldsymbol{n},s}$ *is*

$$ \Sigma^1_{\boldsymbol{n},s}(i,j) = \binom{i+s-2}{s-1}\binom{j+s-2}{s-1} {}_4F_3\left[\begin{matrix}n_1-i+1,\ n_2-j+1,\ 1-s,\ 1-s\\ 1,\ 2-s-i,\ 2-s-j\end{matrix}; 1\right]. $$

Proof. For each $T \in \mathcal{T}^1_{\boldsymbol{n},s}$ with $\boldsymbol{n} = (n_1, n_2)$, consider its corresponding biword, in which each column $\begin{smallmatrix} i \\ j \end{smallmatrix}$ contributes 1 to the (i,j) entry in T. Hence the entry $\Sigma^1_{\boldsymbol{n},s}(i,j)$ equals the total number of occurrences of the column $\begin{smallmatrix} i \\ j \end{smallmatrix}$ within all possible biwords.

Suppose that the ℓth column of a biword is $\begin{smallmatrix} i \\ j \end{smallmatrix}$. In the top row, this implies that the $\ell - 1$ entries to the left of i lie in the set $\{1, \ldots, i\}$, and the $s - \ell$ entries to the right of i lie in the set $\{i, i+1, \ldots, n\}$, which contains $n + 1 - i$ elements. Hence by (5.6), the following product of binomial coefficients equals the number of ways to fill the top row of the biword such that the ℓth entry is i:

$$\binom{(\ell-1)+i-1}{\ell-1}\binom{(s-\ell)+(n_1+1-i)-1}{s-\ell}.$$

Using the same argument for the bottom row of the biword (replacing i by j), and then multiplying the top and bottom results, we conclude that the number of biwords with $\begin{smallmatrix} i \\ j \end{smallmatrix}$ as the ℓth column equals

$$\binom{i+\ell-2}{\ell-1}\binom{j+\ell-2}{\ell-1}\binom{n_1-i+s-\ell}{s-\ell}\binom{n_2-j+s-\ell}{s-\ell}.$$

To obtain the number of times $\begin{smallmatrix} i \\ j \end{smallmatrix}$ occurs as any column in a biword, we sum over all columns $\ell = 1, \ldots, d$. Following this with the substitution $k = \ell - 1$, we have

$$\Sigma^1_{\boldsymbol{n},s}(i,j) = \sum_{k=0}^{s-1} \underbrace{\binom{i+k-1}{k}\binom{j+k-1}{k}}_{\substack{\text{coeff. of } z^k \text{ in} \\ {}_2F_1\left[\substack{i,j \\ 1};z\right]}} \underbrace{\binom{n_1-i+s-1-k}{s-1-k}\binom{n_2-j+s-1-k}{s-1-k}}_{\substack{\text{coeff. of } z^{(s-1)-k} \text{ in} \\ {}_2F_1\left[\substack{x-i+1,\ y-j+1 \\ 1};z\right]}},$$

where we have recognized the two coefficients from the hypergeometric series in (3.21). This makes it clear that $\Sigma^1_{\boldsymbol{n},s}(i,j)$ is the coefficient of z^{s-1} in the product

$$\begin{aligned} &{}_2F_1\left[\begin{matrix} i,j \\ 1 \end{matrix};z\right] {}_2F_1\left[\begin{matrix} n_1-i+1,\ n_2-j+1 \\ 1 \end{matrix};z\right] \\ &= \sum_{k=0}^{\infty} \binom{i+k-1}{k}\binom{j+k-1}{k} {}_4F_3\left[\begin{matrix} n_1-i+1,\ n_2-j+1,\ -k,\ -k \\ 1,\ 1-k-i,\ 1-k-j \end{matrix};1\right] z^k, \end{aligned}$$

where the equality is just the identity (3.22). Reading off the coefficient for $k = s - 1$, we obtain the expression in the theorem. □

3.7 Values of $\Sigma^1_{\boldsymbol{n},s}$ and plots

In this section, we show some values of $\Sigma^1_{\boldsymbol{n},s}$ for $\boldsymbol{n} = (5,5)$ and varying s, as well as compare the time efficiency of our different approaches. We begin by listing values of $\Sigma_{(5,5),d}$:

$$\Sigma^1_{(5,5),1} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\Sigma^1_{(5,5),2} = \begin{pmatrix} 26 & 22 & 18 & 14 & 10 \\ 22 & 20 & 18 & 16 & 14 \\ 18 & 18 & 18 & 18 & 18 \\ 14 & 16 & 18 & 20 & 22 \\ 10 & 14 & 18 & 22 & 26 \end{pmatrix}$$

$$\Sigma^1_{(5,5),3} = \begin{pmatrix} 251 & 193 & 141 & 95 & 55 \\ 193 & 173 & 150 & 124 & 95 \\ 141 & 150 & 153 & 150 & 141 \\ 95 & 124 & 150 & 173 & 193 \\ 55 & 95 & 141 & 193 & 251 \end{pmatrix}$$

$$\Sigma^1_{(5,5),4} = \begin{pmatrix} 1476 & 1064 & 720 & 440 & 220 \\ 1064 & 960 & 816 & 640 & 440 \\ 720 & 816 & 848 & 816 & 720 \\ 440 & 640 & 816 & 960 & 1064 \\ 220 & 440 & 720 & 1064 & 1476 \end{pmatrix}$$

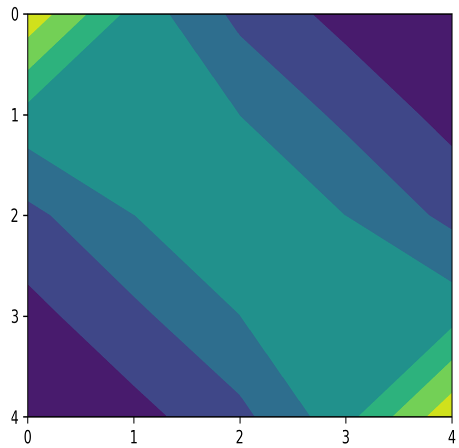
$$\Sigma^1_{(5,5),5} = \begin{pmatrix} 6376 & 4385 & 2805 & 1595 & 715 \\ 4385 & 4006 & 3360 & 2530 & 1595 \\ 2805 & 3360 & 3546 & 3360 & 2805 \\ 1595 & 2530 & 3360 & 4006 & 4385 \\ 715 & 1595 & 2805 & 4385 & 6376 \end{pmatrix}$$

$$\Sigma^1_{(5,5),6} = \begin{pmatrix} 22252 & 14762 & 9042 & 4862 & 2002 \\ 14762 & 13672 & 11352 & 8272 & 4862 \\ 9042 & 11352 & 12132 & 11352 & 9042 \\ 4862 & 8272 & 11352 & 13672 & 14762 \\ 2002 & 4862 & 9042 & 14762 & 22252 \end{pmatrix}$$

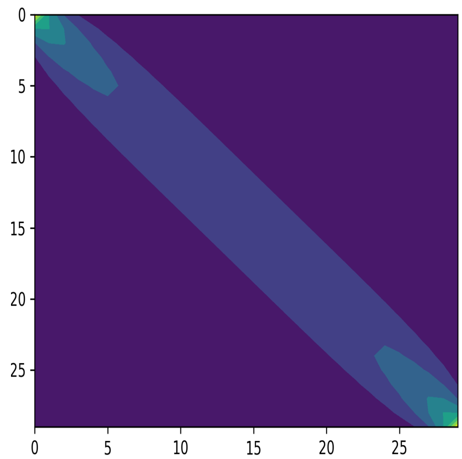
$$\Sigma^1_{(5,5),7} = \begin{pmatrix} 66352 & 42779 & 25311 & 13013 & 5005 \\ 42779 & 40150 & 33066 & 23452 & 13013 \\ 25311 & 33066 & 35706 & 33066 & 25311 \\ 13013 & 23452 & 33066 & 40150 & 42779 \\ 5005 & 13013 & 25311 & 42779 & 66352 \end{pmatrix}$$

$$\Sigma^1_{(5,5),8} = \begin{pmatrix} 175252 & 110396 & 63492 & 31460 & 11440 \\ 110396 & 104896 & 85800 & 59488 & 31460 \\ 63492 & 85800 & 93456 & 85800 & 63492 \\ 31460 & 59488 & 85800 & 104896 & 110396 \\ 11440 & 31460 & 63492 & 110396 & 175252 \end{pmatrix}$$

To better visualize $\Sigma^1_{\boldsymbol{n},s}$, we show contour plots in Figure 4.



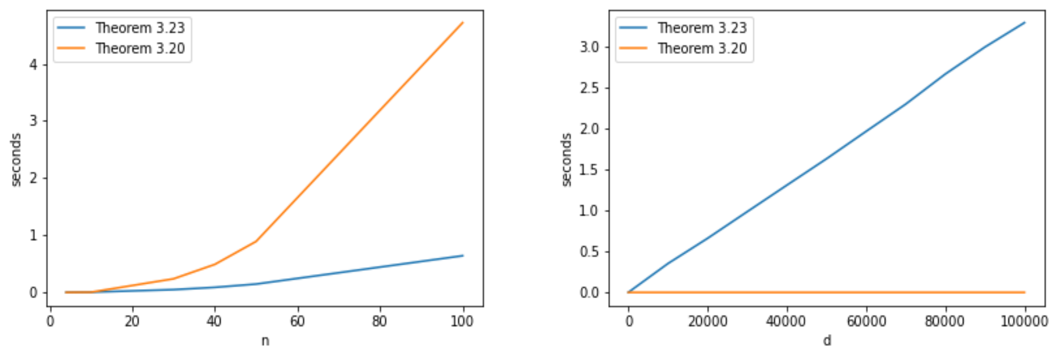
(a) Contour plot of $\Sigma^1_{(5,5),30}$.



Contour plot of $\Sigma^1_{(30,30),10000}$.

Figure 4: Contour plots of $\Sigma^1_{\boldsymbol{n},s}$. Blue corresponds to lower level curves, yellow to higher level curves. In $\Sigma^1_{(5,5),30}$, the entries lie in the interval $[1.6 \times 10^8, 6.7 \times 10^9]$; in $\Sigma^1_{(30,30),10000}$ the entries lie in the interval $[8.6 \times 10^{155}, 2.4 \times 10^{172}]$.

Finally, we examine the runtime of the formulas found for $\Sigma^1_{\boldsymbol{n},s}$ in Theorem 3.20 and 3.23 under varying parameters. We restrict our examination to square matrices, i.e. we let $\boldsymbol{n} = (n, n)$. In one experiment, we fix the value of s and let n grow. Conversely, afterwards we will fix n and inspect the runtime for growing s.



(a) (b)

Figure 5: Comparison of computing time with respect to the parameters s and n. In 5a, we fix $s = 30$ and compare the runtime (in seconds) of both approaches for varying n. In 5b, we fix $n = 5$ and let d vary.

In Figure 5, we can see that Theorem 3.20 provides a formula that is almost immune to changes in the grand total s of the matrices that are summed. On the other hand, increasing the dimensions has a large effect on the runtime of the formula. The behavior is exactly opposite for Theorem 3.23. Generally, our results provide an efficient formula depending on the circumstance.

4 GENERALIZING THE COST MATRIX

When working with the EMD, it is natural to question the choice of the cost matrix C. While intuitive, and justified in some applications, one can imagine that there exist applications with vastly different costs of "moving earth".

A convenient solution for this problem presents itself through Proposition 3.1 in Chapter 3:

$$\overline{\text{EMD}}_{\boldsymbol{n},s} = \frac{\text{trace}\left(C^{\text{T}} \cdot \Sigma^1_{\boldsymbol{n},s}\right)}{\binom{s+n_1-1}{x}\binom{s+n_2-1}{y}}. \tag{4.1}$$

Reducing the expected EMD to a single multiplication with C, we are able to use our results for $\Sigma^1_{\boldsymbol{n},s}$ to examine the behavior of the EMD under different cost matrices. However, as we recall from Chapter 3, in applying (4.1) we make the assumption that the cost matrix C satisfies the Monge property. If we wish to compute the expected EMD under a cost matrix that does not satisfy the Monge property, we have to resort to its original definition as a linear programming problem. In necessary cases throughout this chapter, we add a subscript C to EMD_C to indicate the choice of cost matrix.

4.1 Cost matrices with Monge property

An elementary fact about Monge matrices is their convexity: given Monge matrices M_1, M_2, any convex combination $tM_1 + (1-t)M_2$ for all $t \in [0,1]$ is also a Monge matrix. Therefore, to examine the EMD under different monge matrices of dimensions $\boldsymbol{n} = (n_1, n_2)$, we define the *Monge polytope* $\mathcal{M}_{\boldsymbol{n},s}$ for some $s \in \mathbb{N}_{>0}$ through the inequalities:

$$M_{ij} \geq 0 \tag{4.2}$$

$$\sum_{i,j} M_{ij} = s \tag{4.3}$$

$$M_{ij} + M_{i'j'} \leq M_{i'j} + M_{ij'} \text{ for all } 1 \leq i < i' \leq n_1 \text{ and } 1 \leq j < j' \leq n_2. \tag{4.4}$$

The Monge polytope is embedded in $\mathbb{R}^{n_1 n_2}$. Each point $p \in \mathbb{R}^{n_1 n_2}$ in the polytope is also a Monge matrix $m \in \mathbb{R}^{n_1 \times n_2}$. The Monge polytope itself, despite living in $\mathbb{R}^{n_1 n_2}$, lives in

$(n_1 n_2 - 1)$-dimensional space due to constraints on 4-tuples of points. See [25] for more details on the Monge polytope. A similar structure, the *cone of Monge matrices*, has been examined by Rudolf in 1995, see [61]. As the name suggests, the cone of Monge matrices can be obtained by removing condition (4.3). A structure such as the Monge polytope allows us to "reverse" the problem of computing the EMD: instead of fixing a cost matrix and computing the EMD for different ordered pairs of compositions, we can fix a pair of compositions and compute the EMD for different cost matrices.

An example of the Monge polytope can be seen in Figure 6, where we restrict the grand total to $s = 2$ (purely to avoid noninteger vertices) and the dimensions to $\boldsymbol{n} = (2, 2)$. The resulting Monge polytope is 3-dimensional; its facets are 2 trapezoids, 2 triangles and 1 rectangle.

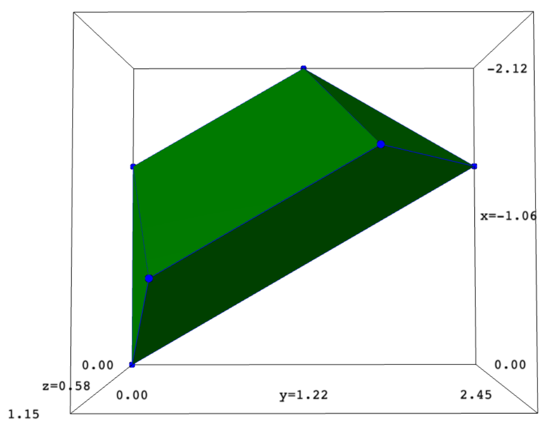


Figure 6: The Monge polytope with grand total 2 and dimensions $\boldsymbol{n} = (2, 2)$.

Listing 4.1 shows the Python code used to obtain the Monge polytope. In the code, we make use of the Sage-provided linear programming class *MixedIntegerLinearProgram*. Alternatively, for users who wish to use pure Python, libraries such as *SciPy* come with linear programming functionality, see [62].

```
def Monge_Polytope(dimensions, grand_total):
  rows, cols = dimensions
  problem = MixedIntegerLinearProgram(maximization=False)
  points = problem.new_variable(real=True, nonnegative=True)

  # add monge condition: c(i,j)+c(i’,j’) <= c(i’,j)+c(i,j’) for all i<i’ and j<j’
  for row1 in range(rows):
    for row2 in range(row1+1,rows):
      for col1 in range(cols):
        for col2 in range(col1+1, cols):
          problem.add_constraint(points[row1,col1] - points[row1,col2] - points[row2,col1] + points[row2,col2] <= 0)


  # restrict grand total to obtain a polytope instead of a cone
  problem.add_constraint(sum(points[i,j] for i in range(rows) for j in range(cols))==grand_total)

  return problem.polyhedron()
```

Listing 4.1: Python code to compute the Monge polytope using linear programming.

To compute the EMD under different cost matrices with Monge property, we focus on the vertices of the Monge polytope. For the polytope shown in Figure 6, in which each point has grand total $s = 2$ and is $n^2 = 4$-dimensional, we have vertices

$$V = \{(1,0,1,0), (1,1,0,0), (0,2,0,0), (0,0,2,0), (0,1,0,1), (0,0,1,1)\}.$$

The corresponding 2×2 Monge matrices are

$$\left\{ \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \right\}. \tag{4.5}$$

Using our results from Chapter 3, we can see that the sum of all width-one matrices with dimensions 2×2 and grand total 2 is $\Sigma^1_{(2,2),2} = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$. We can now compute the mean EMD under any given (Monge) cost matrix C as

$$\overline{\text{EMD}}_C = \frac{\text{trace}\left(C^{\text{T}} \cdot \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}\right)}{\binom{3}{2}^2}.$$

The previously used cost matrix $C_{ij} = |i - j|$ yields the mean EMD[1] of $\frac{8}{9}$. Examining the vertices of the Monge polytope as alternative cost matrices, we find that the only cost matrices which yield the same result are the two matrices in which the grand total is contained at a single index, $\begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}$. When used as cost matrices, all other matrices in 4.5 yield a mean EMD of 1.

A perspective assumed in [10, 24, 52], which shows an application of the EMD, was that of grade distributions. Suppose we teach a class with 30 students, each of which will receive one of the 5 grades A, B, C, D or F. Then, to compare the performance of our class on, say, two different exams, we might use the EMD. For that reason, in [10, 52], the authors computed the mean EMD of the set $\mathcal{C}(30, 5) \times \mathcal{C}(30, 5)$ – interpreting compositions of 30 into 5 parts (or histograms with 30 units and 5 bins) as grade distributions of classes with 30 students and 5 grades. We assume the same perspective, and compute a mean EMD of 0.7304 over all ordered pairs of compositions of 30 into 5 parts. To compare this result to the mean EMD yielded under cost defined by vertices of a Monge polytope, we first recall the standard 5×5 cost matrix C with $C_{ij} = |i - j|$. C has grand total 40, so we focus on the Monge polytope $\mathcal{M}_{(5,5),40}$ which contains C. Computing $\mathcal{M}_{(5,5),40}$ in Sage, we see that there are 42 vertices. The resulting values, in no particular order, are displayed in Figure 7.

[1] To compare this value to [10, §5.1], we need to unit normalize the mean EMD through division by $s(\max\{n_1, n_2\} - 1)$, in this example division by 2.

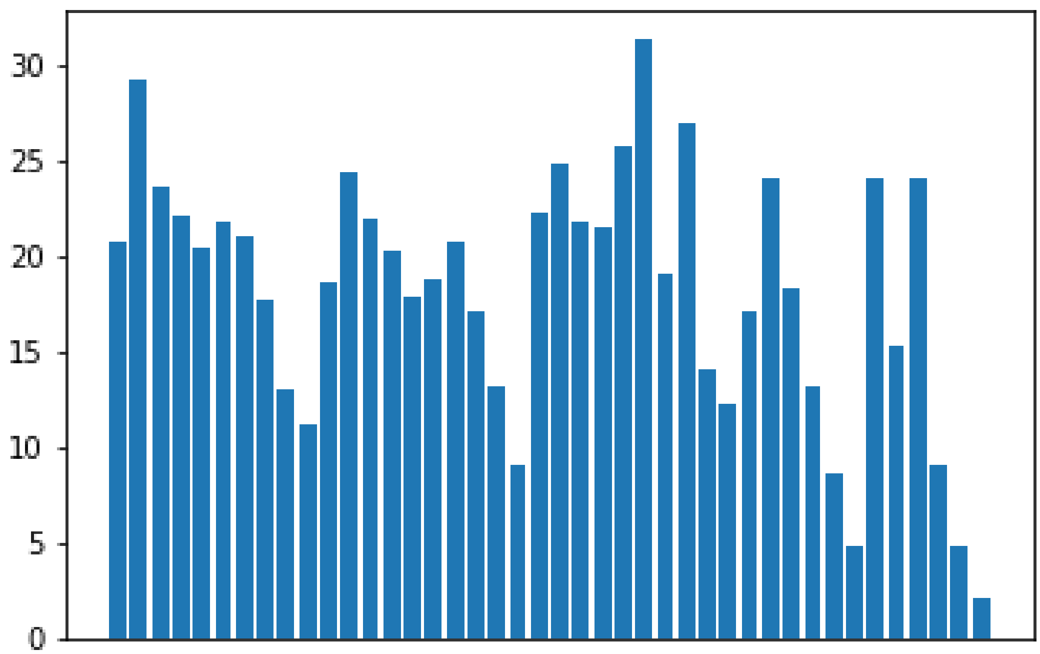


Figure 7: Mean EMD values resulting from all 42 cost matrices that are vertices of $\mathcal{M}_{(5,5),40}$ (in no particular order).

The value of the mean EMD has a large range – the minimal mean EMD with a value of 2.1758 is found when the cost matrix is

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 40 & 0 & 0 & 0 & 0 \end{bmatrix},$$

that is, the entire grand total lies in the bottom left corner of the matrix. On the other hand, the maximal mean EMD with a value of 31.3557 is found using the cost matrix

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 8 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 8 & 0 \end{bmatrix}.$$

Taken over all examined cost matrices, the "mean mean" EMD is 18.2952 with a variance of 43.4421. We show a histogram of the distribution in Figure 8.

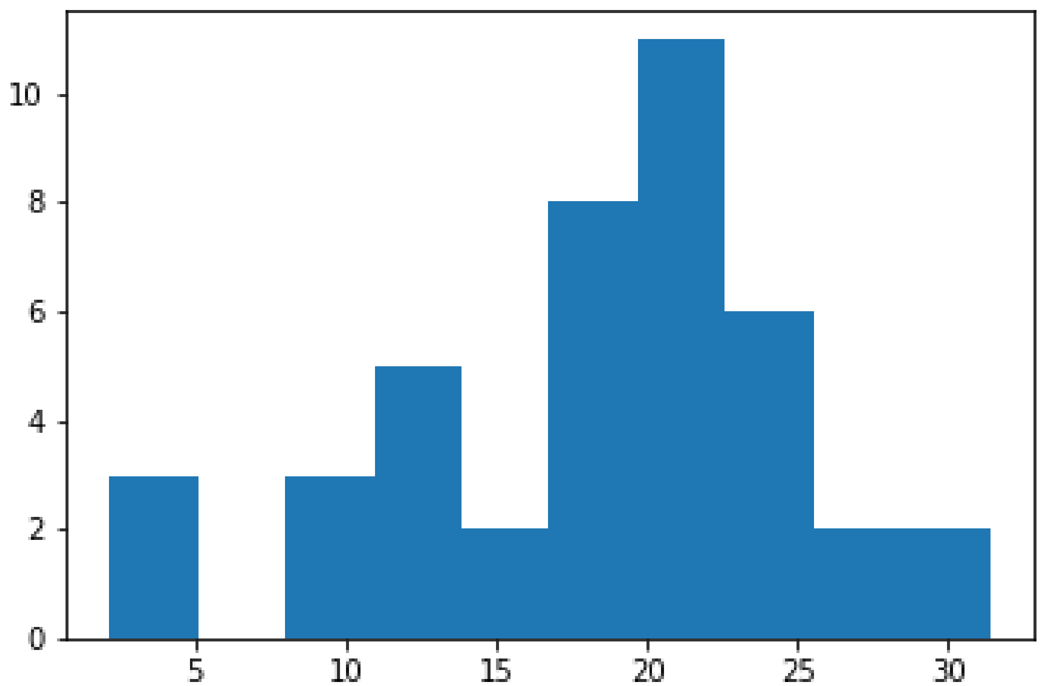


Figure 8: Distribution of the mean EMD values under varying cost matrices.

After computing the EMD under general Monge cost matrices, we use the next section to compute the EMD under non-Monge cost matrices, by returning to its definition as the solution to a linear programming problem.

4.2 Non-Monge cost matrices and linear programming

We begin by recalling the definition of the EMD as the solution of the Monge problem:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i,j=1}^{n} C_{ij} T_{ij}, & & \\ \text{subject to} \quad & T_{ij} \geq 0 & & \text{for all } 1 \leq i, j \leq n, \\ \text{and} \quad & \sum_{j=1}^{n} T_{ij} = \lambda(i) & & \text{for each } 1 \leq i \leq n, \\ \text{and} \quad & \sum_{i=1}^{n} T_{ij} = \mu(j) & & \text{for each } 1 \leq j \leq n. \end{aligned}$$

This definition, although less efficient to execute, allows for the use of a truly general cost matrix. In this section, we compute the (expected) EMD under different cost matrices and compare results. The implementation of the linear programming algorithm is shown in Listing 4.2.

To examine the EMD under a non-Monge cost matrix, we first require to choose a non-Monge matrix. Let therefore G be an unweighted, circular graph, an example with 5 nodes is shown in Figure 9.

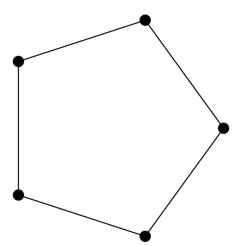


Figure 9: An unweighted, circular graph with 5 nodes.

Alternatively, we provide the graph in the form of an adjacency matrix:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

We define the cost matrix C' as the distance matrix of G. For the graph in Figure 9, the cost matrix is

$$C' := \begin{bmatrix} 0 & 1 & 2 & 2 & 1 \\ 1 & 0 & 1 & 2 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 2 & 1 & 0 & 1 \\ 1 & 2 & 2 & 1 & 0 \end{bmatrix}.$$

The matrix C', as well as the distance matrix of a graph with an arbitrary number of vertices, is clearly non-Monge – for an example of a 2×2 block breaking the Monge condition, we refer to the upper right corner: $C'_{14} + C'_{25} = 4$, which is greater than $C'_{15} + C'_{24} = 3$. In general, for a circular graph constructed in this way, we have $C'_{1,n-1} + C'_{2,n} > C'_{1,n} + C'_{2,n-1}$.

Remark. To note the difference between C' and the earlier used 5×5 matrix C with $C_{ij} = |i - j|$, it suffices to examine which pair of histograms yields the largest EMD. Under C, movement from 0 to 5 or vice versa is associated with the highest cost of 4. Under C', however, the highest cost that can be incurred is 2, and it is not found by moving from 0 to 5. If we focus on histograms with s units and $d = 5$ bins, the pair with the largest EMD under C is clearly (λ, μ) with $\lambda = (s, 0, 0, 0, 0)$ and $\mu = (0, 0, 0, 0, s)$. On the other hand, the largest EMD under C' can be found, for example, by comparing (λ', μ') with $\lambda' = (s, 0, 0, 0, 0)$ and $\mu' = (0, 0, s, 0, 0)$.

With the goal of showing the difference between the linear programming–based approach and the result in Chapter 3, we pick C' as our non–Monge cost matrix and compute the mean EMD incurred by all ordered pairs of histograms with $s = 10$ units and $d = 5$ bins. Due to the absence of an efficient formula for the linear programming approach, we are forced to evaluate

$$\overline{\text{EMD}}_{C'} = \frac{\sum_{(\lambda,\mu)\in\mathcal{C}(15,5)^2} \text{EMD}_{C'}(\lambda, \mu)}{\binom{14}{10}} \approx 7.3141.$$

We use the code in Listing 4.2 to compute the EMD under the cost matrix C'.

This computation takes 645.2771 seconds, or almost 11 minutes. For comparison, computing the mean EMD with the formula found in Chapter 3, we find a result much faster, in only 0.0004 seconds. However, as expected, we find the wrong result:

$$\overline{\text{EMD}}_{C'} = \frac{\operatorname{trace}(C^{\mathrm{T}}\Sigma^1_{(5,5),15})}{\binom{14}{10}} \approx 8.9739.$$

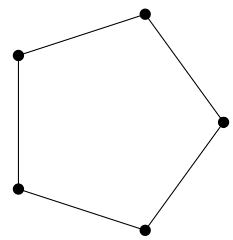
```python
def EMD_LP(hist1, hist2, C):
  # normalize histograms
  colSums = hist1
  rowSums = hist2
  bins = len(hist1)

  # use sage classes for linear programming
  problem = MixedIntegerLinearProgram(maximization=False)
  transportPlan = problem.new_variable(real=True, nonnegative=True)

  # set LP objective: minimize EMD
  problem.set_objective(sum(C[i,j]*transportPlan[i,j] for i in range
(bins) for j in range(bins)))

  # constraints of transportplan: row and columnsums; nonnegative
already set above
  for col in range(bins):
    problem.add_constraint(sum(transportPlan[row, col] for row in
range(bins)) == colSums[col])

  for row in range(bins):
    problem.add_constraint(sum(transportPlan[row, col] for col in
range(bins)) == rowSums[row])

  return problem.solve()
```

Listing 4.2: Python code to compute the EMD as linear programming problem.

Remark. When using the Monge cost matrix $C_{ij} = |i-j|$, the two approaches agree on a resulting mean EMD of 9.8473.

4.2.1 Application to pie charts

In this section, we use a different approach to selecting non-Monge cost matrices. We begin by recalling the circular graph from Figure 9:



Due to the circular shape, we can make slight modifications to achieve resemblance to a *pie chart*. The result is displayed in Figure 10.

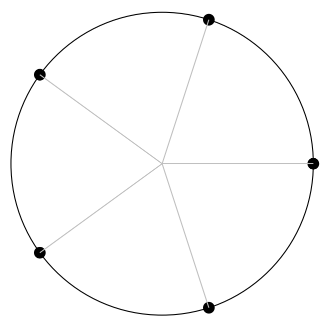
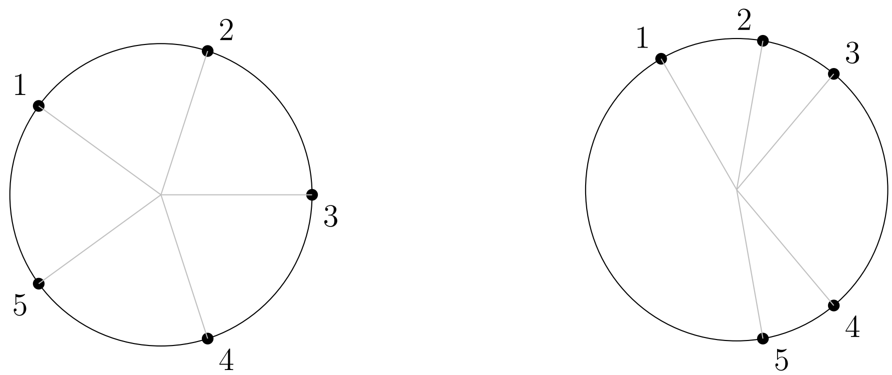


Figure 10: A circular graph with an additional point in the center, connected to each vertex to resemble a pie chart.

We can now assign values to edges, thus turning the graph from unweighted to weighted, by measuring the *angle* between the lines connecting 2 vertices to the center point. In the example shown in Figure 10, where the vertices are evenly spaced, each edge is assigned the value $\frac{360}{5} = 72$. We obtain a distance matrix by measuring the angles between the lines connecting two vertices to the center and similar to the circular graph, the matrix is not Monge:

$$C' = \begin{bmatrix} 0 & 72 & 144 & 144 & 72 \\ 72 & 0 & 72 & 144 & 144 \\ 144 & 72 & 0 & 72 & 144 \\ 144 & 144 & 72 & 0 & 72 \\ 72 & 144 & 144 & 72 & 0 \end{bmatrix}.$$

To see that C' is not Monge, we refer to $C_{14} + C_{25} = 288$ and $C_{15} + C_{24} = 216$. In addition to providing another example of a non-Monge cost matrix, approaching the problem from the perspective of pie charts allows for a new interpretation: while we defined the EMD as a distance measure on histograms, we can now view it as a distance measure on pie charts. For any pie chart p we define as $C'(p)$ the cost matrix obtained by measuring its angles. Furthermore, the measured angles yield a composition of 360 into d parts – in Figure 10, we have $d = 5$. We call this composition $\lambda(p)$. Thus, to measure the distance between an ordered pair of pie charts (p, q), we can simply compute $\text{EMD}_{C'(p)}(\lambda(p), \lambda(q))$ to obtain a distance between the charts, or a cost of moving units along the edges within p such that the resulting pie chart is q.



(a) Pie chart p. (b) A second pie chart, q.

Figure 11: Pie charts p and q. We compute the cost of transforming p into q using the cost matrix defined by p.

For the example given in Figure 11, we have already defined

$$C' = \begin{bmatrix} 0 & 72 & 144 & 144 & 72 \\ 72 & 0 & 72 & 144 & 144 \\ 144 & 72 & 0 & 72 & 144 \\ 144 & 144 & 72 & 0 & 72 \\ 72 & 144 & 144 & 72 & 0 \end{bmatrix}.$$

Moreover, we have $\lambda(p) = (72, 72, 72, 72, 72)$ as well as $\lambda(q) = (40, 30, 100, 30, 160)$. We can use the Sage code in Listing 4.2 to obtain the resulting distance $\text{EMD}_{C'(p)}(\lambda(p), \lambda(q)) = 8224$.

Following this procedure, one can compute the mean EMD among all pairs of pie charts, a step we omit for lack of computing power.

In the next chapter, we generalize our results from Chapter 3 from 2-dimensional $n_1 \times n_2$ matrices to d dimensional $n_1 \times \cdots \times n_d$ tensors. We present a shelling of the d-dimensional order complex and provide 2 formulas to compute the sum of all width–one tensors.

5 THE SUM OF ALL WIDTH-ONE TENSORS

Throughout, we again focus on finding formulas for $\Sigma^1_{\boldsymbol{n},s}$. However, instead of setting $\boldsymbol{n} = (n_1, n_2)$, we let $\boldsymbol{n} = (n_1, \ldots, n_d)$. We are, therefore, generalizing to d-dimensional tensors (or hypermatrices). We show two different formulas to compute $\Sigma^1_{\boldsymbol{n},s}$; the first obtained by using a connection to one-row semistandard Young Tableaux. To obtain a second formula, we show that the theory of simplicial complexes can again be used to obtain the h- and f-vector of the n-dimensional order complex. We obtain the result through an application of multiset Eulerian polynomials.

Remark. With slight changes to notation, the remainder of this chapter has been published in a collaboration with William Q. Erickson in [27]. Similar to the generalization Erickson in [24] provides to Bourn and Willenbring in [10], we extend our new approach from Chapter 3 to higher dimensions.

5.1 Introduction

In this chapter, we let $\mathcal{T}^1_{\boldsymbol{n},s}$ be the set of all $\boldsymbol{n} = (n_1, \ldots, n_d)$-dimensional tensors (equivalently, hypermatrices) with nonnegative integer entries summing to s, such that the nonzero entries lie on a single path consisting of steps in the positive directions of the standard basis vectors $\mathbf{e}_1, \ldots, \mathbf{e}_d$. For example, Figure 12 shows a typical element of $\mathcal{T}^1_{(3,3,3),s}$, where the nonzero entries lie on the lattice points along the marked path.

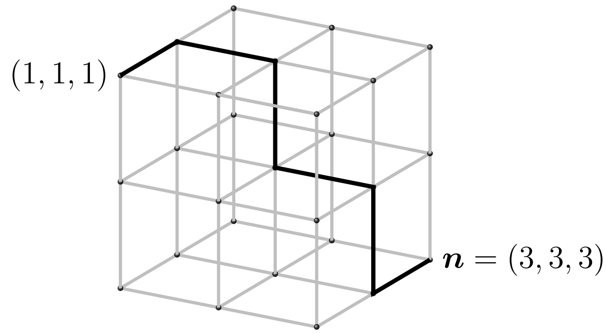


Figure 12: Visualization of the support of an element of $\mathcal{T}^1_{(3,3,3),s}$.

As shown in [28], computing the sum of such matrices in $d = 2$ dimensions has useful applications in optimal transport, drastically simplifying the problem (first solved recursively in [10]) of computing the expected value of the EMD between two compositions. The methods of [10] were generalized in [24] to find a recursion for the expected value of the generalized EMD between an arbitrary number d of compositions. (For computational treatments of the d-dimensional transport problem, see [3] and [47], for example.) The relationship between the present paper and [24] can be regarded as the d-dimensional analogue of the relationship between [28] and [10]: in particular, we give two explicit formulas for the sum $\Sigma^1_{\boldsymbol{n},s}$ of all tensors in $\mathcal{T}^1_{\boldsymbol{n},s}$. These formulas, in turn, can be used to obtain non-recursive formulas for the expected value of the d-fold EMD (easily seen by adapting the argument in [28, §6]).

This connection between width-one tensors and optimal transport theory arises as follows (see [3] for details). In the d-dimensional analogue of the classical transportation problem, the inputs are "supply–demand" vectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ and a "cost" tensor C with dimensions $\boldsymbol{n}$. Each input vector $\mathbf{v}_i$ has length n_i, and its entries are nonnegative integers summing to s; in combinatorial language, $\mathbf{v}_i$ is a composition of s into n_i parts. The objective is to find a tensor T, with coordinate hyperplane sums prescribed by the $\mathbf{v}_i$, which minimizes the Hadamard product of C and T. This minimum value is said to be the EMD between the input vectors. It turns out that when C satisfies the *d-dimensional Monge property* (see [3, Def. 2.1]), there is a greedy algorithm called the northwest corner rule, which outputs the optimal solution T in the form of a width-one tensor. In fact, for fixed s, the northwest corner rule yields a bijection between the set of all possible d-tuples of input vectors and the set $\mathcal{T}^1_{\boldsymbol{n},s}$. Therefore, the sum $\Sigma^1_{\boldsymbol{n},s}$ can be used to obtain the expected value of the EMD.

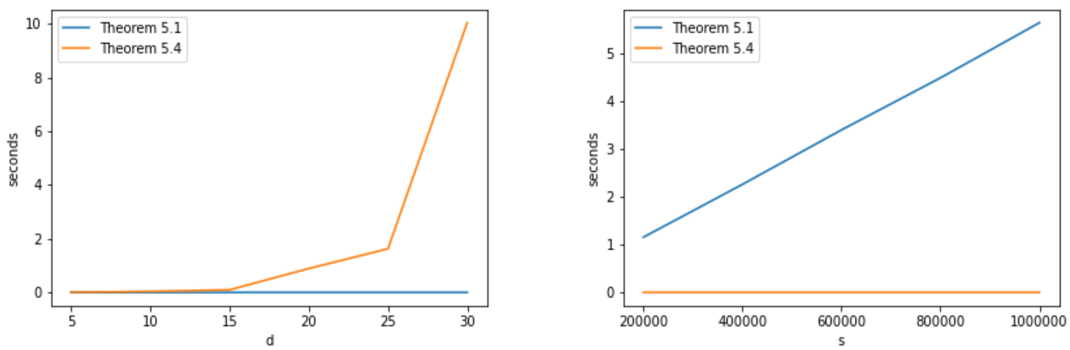
In Section 5.3 we give our first formula for $\Sigma^1_{\boldsymbol{n},s}$. We set up a bijection between $\mathcal{T}^1_{\boldsymbol{n},s}$ and tuples of one-row semistandard tableaux (essentially the inverse of the northwest corner rule mentioned above), which allows us to write down a formula for our desired sum $\Sigma^1_{\boldsymbol{n},s}$ in terms of binomial coefficients (Theorem 5.1). In Sections 5.4 and 5.5, we consider the problem through the lens of Stanley–Reisner theory. We describe the order complex on the standard basis of the $\boldsymbol{n}$-dimensional tensors, and we use a special case of an EL-shelling to find the corresponding h-polynomial. This h-polynomial turns out to be a multiset Eulerian polynomial, as we show in Section 5.5; these polynomials, studied by MacMahon and many others since, enumerate the descents in multiset permutations. We conclude by presenting a second explicit formula for $\Sigma^1_{\boldsymbol{n},s}$ using techniques from Stanley–Reisner theory (Theorem 5.4).

Similar to the two formulas for matrices in [28], the two formulas in this section behave in opposite ways with regard to computing time. Although this issue falls outside the focus of the section, nonetheless it is not hard to verify that Theorem 5.1 outperforms Theorem 5.4 as s increases for fixed $\boldsymbol{n}$; the opposite is true, however, as the dimension d or the parameters n_i increase for fixed s. Therefore, as shown in Figure 13, the user who wishes to compute $\Sigma^1_{\boldsymbol{n},s}$ should choose between the two theorems according to the sizes of $\boldsymbol{n}$ and s: roughly speaking, Theorem 5.4 is preferable for low values of d and $|\boldsymbol{n}|$, while Theorem 5.1 is more efficient for low values of s.

5.2 Notation and statement of the problem

Throughout, we denote by $\boldsymbol{n} := (n_1, \ldots, n_d)$ and $\boldsymbol{x} := (x_1, \ldots, x_d)$. We write $|\boldsymbol{x}| := \sum_i x_i$, as well as $\min(\boldsymbol{x}) := \min_i\{x_i\}$ and $\max(\boldsymbol{x}) := \max_i\{x_i\}$. Let $\mathbf{1} := (1, \ldots, 1)$. Define the poset

$$\Pi_{\boldsymbol{x}} := [x_1] \times \cdots \times [x_d] = \{(a_1, \ldots, a_d) : 1 \leq a_i \leq x_i \text{ for all } i\}, \tag{5.1}$$



(a)

(b)

Figure 13: Comparison of computing time with respect to the parameters d and s. In 13a, we fix $s = 5$ and compare the runtime (in seconds) of both approaches for varying d. For arbitrary d, we measure the time it takes to compute the entry at $\boldsymbol{x} = (\lfloor \frac{d}{2} \rfloor, \ldots, \lfloor \frac{d}{2} \rfloor)$ in the d-dimensional hypercube with $\boldsymbol{n} = (d, \ldots, d)$. In 13b, we fix $d = 4$ and let s vary.

equipped with the product order, so that $\boldsymbol{a} \preceq \boldsymbol{b} \iff a_i \leq b_i$ for each $i = 1, \ldots, d$. Because our main problem addresses $n_1 \times \cdots \times n_d$ tensors, we will always be working inside $\Pi_{\boldsymbol{n}}$, but it will be useful to consider the subposets $\Pi_{\boldsymbol{x}}$, which are the lower-order ideals generated by each $\boldsymbol{x} \in \Pi_{\boldsymbol{n}}$.

Analagous to the 2-dimensional case described in Chapter 3, a *chain* is a totally ordered subset of $\Pi_{\boldsymbol{x}}$, and an *antichain* is a subset whose elements are pairwise incomparable. The *width* of a subset $S \subseteq \Pi_{\boldsymbol{x}}$ is the size of the largest antichain contained in S. In particular, S has width 1 if and only if S is a chain.

In this section, we consider certain tensors of order d. Equivalently, the reader may prefer to consider d-dimensional arrays (also called hypermatrices). Taking the real numbers $\mathbb{R}$ as our ground field, we let

$$\mathcal{T}_{\boldsymbol{n}} := \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_d}$$

denote the space of order-d tensors with dimensions $\boldsymbol{n}$. Upon fixing the standard basis $\{e_1, \ldots e_{n_i}\}$ for each factor $\mathbb{R}^{n_i}$, every tensor $T \in \mathcal{T}_{\boldsymbol{n}}$ can be written uniquely in the form

$$T = \sum_{\boldsymbol{x} \in \Pi_{\boldsymbol{n}}} T[\boldsymbol{x}] \, e_{x_1} \otimes \cdots \otimes e_{x_d}, \tag{5.2}$$

where the scalars $T[\boldsymbol{x}] \in \mathbb{R}$ are called the *components* of T. Hence T is described completely by its components $T[\boldsymbol{x}]$, which can be regarded as the entries in an $n_1 \times \cdots \times n_d$ array. The *elementary tensor* $E_{\boldsymbol{x}}$ is given by

$$E_{\boldsymbol{x}}[\mathbf{y}] = \delta_{\boldsymbol{x}\mathbf{y}}, \tag{5.3}$$

where δ is the Kronecker delta. Hence $E_{\boldsymbol{x}}$ can be regarded as an array with 1 in position $\boldsymbol{x}$ and 0's elsewhere.

The *support* of a tensor T is the set

$$\operatorname{supp}(T) := \{\boldsymbol{x} \in \Pi_{\boldsymbol{n}} : T[\boldsymbol{x}] \neq 0\}.$$

We say that T is a *width-one* tensor if $\operatorname{supp}(T)$ has width 1 as a subposet of $\Pi_{\boldsymbol{n}}$. In this section, we restrict our attention to those width-one tensors whose components are nonnegative integers summing to some positive integer s. We denote this set by

$$\mathcal{T}^1_{\boldsymbol{n},s} := \left\{ T \in \mathcal{T}_{\boldsymbol{n}} : \begin{array}{l} T \text{ is width-one,} \\ T[\boldsymbol{x}] \in \mathbb{Z}_{\geq 0} \text{ for all } \boldsymbol{x} \in \Pi_{\boldsymbol{n}}, \\ \sum_{\boldsymbol{x} \in \Pi_{\boldsymbol{n}}} T[\boldsymbol{x}] = s. \end{array} \right\}. \tag{5.4}$$

The main problem of this section is to write down an explicit formula for the sum of all tensors in $\mathcal{T}^1_{\boldsymbol{n},s}$, under the usual componentwise addition. We denote this sum by

$$\Sigma^1_{\boldsymbol{n},s} := \sum_{T \in \mathcal{T}^1_{\boldsymbol{n},s}} T. \tag{5.5}$$

5.3 Main result, first version

We generalize the argument in [28, §3]. It follows from (5.2) and (5.3) that each tensor $T \in \mathcal{T}_{\boldsymbol{n}}$ can be written as a unique sum of elementary tensors

$$T = \sum_{\boldsymbol{x} \in \Pi_{\boldsymbol{n}}} T[\boldsymbol{x}] E_{\boldsymbol{x}} = \sum_{\boldsymbol{x} \in \operatorname{supp}(T)} T[\boldsymbol{x}] E_{\boldsymbol{x}}.$$

Now suppose that $T \in \mathcal{T}^1_{\boldsymbol{n},s}$, as defined in (5.4). Then by definition, the $\boldsymbol{x}$'s appearing in the right-hand sum above form a chain in $\Pi_{\boldsymbol{n}}$. Writing out all of these $\boldsymbol{x}$'s as column vectors in ascending order, with multiplicities $T[\boldsymbol{x}]$, we obtain a $d \times s$ matrix. Note that the entries in each row of this matrix are weakly increasing, and the entries in the ith row are elements in $[n_i]$. We write each row as a row of boxes. (We do this primarily to evoke a one-row semistandard Young tableau, which is the same thing as a weakly increasing sequence.) We thus obtain d rows with s boxes each:

$$\begin{array}{ccccccl}
\square\square\square & \cdots & \boxed{x_1} & \cdots & \square\square\square & & \leftarrow \text{weakly increasing in } [n_1] \\
\square\square\square & \cdots & \boxed{x_2} & \cdots & \square\square\square & & \leftarrow \text{weakly increasing in } [n_2] \\
\vdots & & \vdots & & \vdots & & \qquad\qquad\vdots \\
\square\square\square & \cdots & \boxed{x_d} & \cdots & \square\square\square & & \leftarrow \text{weakly increasing in } [n_d]
\end{array}$$

This procedure is invertible: given d rows of length s, with entries in the ith row weakly increasing in $[n_i]$, we recover the associated width-one tensor in $\mathcal{T}^1_{\boldsymbol{n},s}$ by summing the s elementary tensors $E_{\boldsymbol{x}}$, for each column $\boldsymbol{x}$. In our picture above, we have filled in the entries $\boldsymbol{x} = (x_1, \ldots, x_d)$ for one typical column; note that this column contributes 1 to the component $T[\boldsymbol{x}]$. It follows that we have a bijection between $\mathcal{T}^1_{\boldsymbol{n},s}$ and the set of d-tuples of weakly increasing sequences in $[n_1], \ldots, [n_d]$.

It will be a useful fact that the number of weakly increasing sequences of length ℓ, taken from the set $[p]$, equals the number of (weak) integer compositions of ℓ into p parts, which is well known to be

$$\binom{\ell + p - 1}{\ell} = \binom{\ell + p - 1}{p - 1}. \tag{5.6}$$

Theorem 5.1. *Let $\Sigma^1_{\boldsymbol{n},s}$ be the sum of all tensors in $\mathcal{T}^1_{\boldsymbol{n},s}$. For each $\boldsymbol{x} \in \Pi_{\boldsymbol{n}}$, we have*

$$\Sigma^1_{\boldsymbol{n},s}[\boldsymbol{x}] = \sum_{j=1}^{s} \prod_{i=1}^{d} \binom{x_i + j - 2}{j - 1} \binom{n_i - x_i + s - j}{s - j}.$$

Proof. For each $T \in \mathcal{T}^1_{\boldsymbol{n},s}$, consider its corresponding d-tuple of weakly increasing rows, as described above. Recall that each column $\boldsymbol{x}$ contributes 1 to the component $T[\boldsymbol{x}]$. Hence the component $\Sigma^1_{\boldsymbol{n},s}[\boldsymbol{x}]$ equals the number of occurrences of the column $\boldsymbol{x}$, counted in all possible d-tuples.

First, for fixed j such that $1 \le j \le s$, we find the number of d-tuples whose jth column is $\boldsymbol{x}$. In other words, we seek the number of d-tuples such that the jth entry in row i is x_i, for each $i = 1, \ldots, d$. For each i, this implies that the $j - 1$ entries to the left of x_i lie in the set $[x_i]$, and that the $s-j$ entries to the right of x_i lie in the set $\{x_i, x_i+1, \ldots, n_i\}$, which contains $n_i - x_i + 1$ elements. Hence by (5.6), the number of ways to fill each row i such that the jth entry is x_i equals

$$\binom{(j-1)+x_i-1}{j-1}\binom{(s-j)+(n_i-x_i+1)-1}{s-j},$$

which simplifies to the expression in the theorem. Taking the product over all rows $i = 1, \ldots, d$, we obtain the number of d-tuples whose jth column equals $\boldsymbol{x}$, as desired.

Finally, to obtain the number of times $\boldsymbol{x}$ occurs as any column in a d-tuple, we sum over all columns $j = 1, \ldots, s$. □

Remark. As mentioned in the introduction, the formula in Theorem 5.1 generalizes the two-dimensional formula in our previous work [28, Theorem. 3.2], described in Theorem 3.23 of Chapter 3. This may not be obvious at first glance, since the two-dimensional version was expressed as a hypergeometric series. Upon setting $d = 2$ in Theorem 5.1 above, one recovers the sum (where each summand is the product of four binomial coefficients) displayed in the proof in [28], immediately before its simplification via hypergeometric identities. (Note, however, that the parameter s in the present work was denoted by d in [28].)

5.4 Stanley–Reisner theory

This section, along with the following section, sets out the theory required to prove our second formula for $\Sigma^1_{\boldsymbol{n},s}$, which we do in Section 5.6. We omit restating the exposition on abstract simplicial complexes provided in [27] and instead refer the reader to Section 3.3 in Chapter 3. To facilitate reading, we restate the relation between the shelling of a complex and its h-vector, which can also be found in Chapter 3 as (3.9):

$$h_\ell = \#\{i \mid \#\mathrm{R}(\mathrm{F_i}) = \ell\}. \tag{5.7}$$

The generalization of our result to higher dimensions starts to differ vastly in the shelling of the d-dimensional order complex, which we describe in the following section.

5.4.1 Lexicographic shellings of posets

Let Π be a finite bounded poset, with $\lessdot$ denoting the covering relation. Let $\mathcal{E} := \{(a, b) : a \lessdot b\}$ be the set of edges of the Hasse diagram of Π. A *labeling* of Π is a function $\lambda : \mathcal{E} \to \mathbb{Z}_{>0}$ assigning a positive integer to each edge of the Hasse diagram. Each labeling λ induces a lexicographic ordering on the set of saturated chains $a_1 \lessdot a_2 \lessdot \cdots \lessdot a_\ell$, via the lexicographic order on the *label sequences* $(\lambda(a_1, a_2), \ldots, \lambda(a_{\ell-1}, a_\ell))$. Following Björner and Wachs [8], we define a special kind of labeling known as an edge-lexicographical (EL) labeling:

Definition 5.2 ([8])**.** We say that λ is an *EL-labeling* of Π if, for all $a < c$ in Π, there exists a unique saturated chain $a \lessdot b_1 \lessdot \cdots \lessdot b_\ell \lessdot c$ such that

$$\lambda(a, b_1) \leq \lambda(b_1, b_2) \leq \cdots \leq \lambda(b_\ell, c), \tag{5.8}$$

and this chain lexicographically precedes all other saturated chains $a \lessdot \cdots \lessdot c$. A chain with the property (5.8) is called an *ascending chain* with respect to λ.

The *order complex* is the simplicial complex whose faces are the chains in Π; hence the facets of $\Delta(\Pi)$ are the maximal chains in the poset Π. An EL-labeling of Π induces a shelling order on the facets of $\Delta(\Pi)$, via the lexicographic order on the maximal chains [6, Thm. 2.3]. Note that an EL-labeling does not guarantee that the maximal chains are totally ordered; nevertheless, arbitrarily breaking ties results in a shelling order.

Let λ be an EL-labeling of Π, and F a facet of $\Delta(\Pi)$. An element $b \in F$ is said to be a *descent* of F (with respect to λ) if F contains $a \lessdot b \lessdot c$ such that $\lambda(a, b) > \lambda(b, c)$. With respect to any shelling induced by λ, the restriction of each facet is precisely its set of descents:

$$\mathrm{R}(F) = \{b : b \text{ is a descent of } F\}. \tag{5.9}$$

(See [9, Thm. 5.8].) We also use the term *descent* in the context of label sequences, in the obvious sense: namely, i is a descent of the label sequence $(\lambda_1, \ldots, \lambda_\ell)$ if $\lambda_i > \lambda_{i+1}$. In this way, the number of descents in a facet equals the number of descents in its label sequence.

5.4.2 The Stanley–Reisner ring

The Stanley–Reisner ring has been described in more detail in Section 3.5 of Chapter 3. We briefly recall important properties in this section. Let Δ be a simplicial complex on the vertex set V. Let K be a field, and $K[\Delta]$ the Stanley–Reisner ring of Δ.

We know from Section 3.5 that $K[\Delta]$ has a K-basis consisting of the monomials whose support is a face of Δ, where we identify these monomials with their images in the quotient ring.

Each shelling of Δ induces a *Stanley decomposition* of the Stanley–Reisner ring:

$$K[\Delta] = \bigoplus_{F} K[F]\, z_{\mathrm{R}(F)}, \tag{5.10}$$

where the direct sum ranges over the facets F, and their restrictions $\mathrm{R}(F)$ are determined by the shelling. Additionally, each monomial in $K[\Delta]$ lies in exactly one summand of (5.10). Similar to Section 3.5, $K[\Delta]$ again inherits the natural grading by degree. Writing $K[\Delta]_s$ to denote the graded component consisting of homogeneous polynomials of degree s, we can restrict (5.10) to a decomposition of each component:

$$K[\Delta]_s = \bigoplus_{k} \bigoplus_{\substack{F:\\ \#\mathrm{R}(F)=k}} K[F]_{s-k}\, z_{\mathrm{R}(F)}, \tag{5.11}$$

where k ranges from 0 to the size of the largest restriction $\mathrm{R}(F)$.

5.4.3 Application to the problem

In this final subsection, we apply the general theory above to the poset $\Pi_{\boldsymbol{x}}$ defined in (5.1). We write $\Delta_{\boldsymbol{x}} := \Delta(\Pi_{\boldsymbol{x}})$ for its order complex. The facets of $\Delta_{\boldsymbol{x}}$ are the maximal chains $\mathbf{1} \lessdot \cdots \lessdot \boldsymbol{x}$. Thus for any facet F of $\Delta_{\boldsymbol{x}}$, we have

$$\#F = |\boldsymbol{x}| - d + 1, \tag{5.12}$$

so $\Delta_{\boldsymbol{x}}$ is indeed pure.

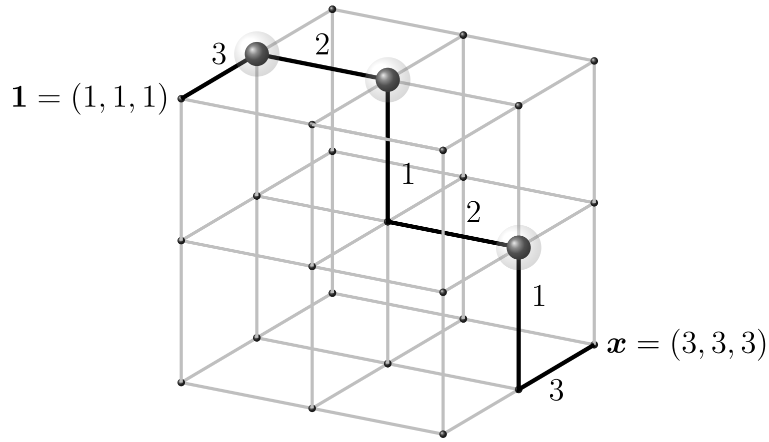


Figure 14: Visualization of a facet F in the order complex $\Delta_{\boldsymbol{x}}$, where $\boldsymbol{x} = (3,3,3)$. Starting in the upper-left at $\mathbf{1}$, we imagine the coordinate vector $\mathbf{e}_1$ pointing downward, $\mathbf{e}_2$ pointing to the right, and $\mathbf{e}_3$ pointing away from the viewer. With respect to the labeling λ in (5.13), the label sequence $(3,2,1,2,1,3)$ of F is shown in the figure. The descents are indicated by the three large dots; by (5.9), these are the elements of $\mathrm{R}(F)$.

Let $\mathbf{e}_i$ denote the vector whose ith coordinate is 1, with 0's elsewhere. If $\mathbf{a} \lessdot \mathbf{b}$, then we have $\mathbf{b} = \mathbf{a} + \mathbf{e}_i$ for some $1 \leq i \leq d$. We define the following labeling on $\Pi_{\boldsymbol{x}}$:

$$\lambda(\mathbf{a}, \mathbf{b}) = i \iff \mathbf{b} = \mathbf{a} + \mathbf{e}_i. \tag{5.13}$$

For example, if $\mathbf{a} = (3,6,4,1)$ and $\mathbf{b} = (3,7,4,1)$, then $\lambda(\mathbf{a}, \mathbf{b}) = 2$. See Figure 14 for a visualization in the case where $\boldsymbol{x} = (3,3,3)$.

It is easy to see that λ is an EL-labeling: for $\mathbf{a} < \mathbf{b}$, the unique ascending chain $\mathbf{a} \lessdot \cdots \lessdot \mathbf{b}$ with respect to λ is obtained from $\mathbf{a}$ by first adding $\mathbf{e}_1$ a total of $b_1 - a_1$ times, then adding $\mathbf{e}_2$ a total of $b_2 - a_2$ times, etc., and finally adding $\mathbf{e}_d$ a total of $b_d - a_d$ times. Moreover, this chain precedes any other maximal chain between $\mathbf{a}$ and $\mathbf{b}$. Being an EL-labeling, λ induces a unique shelling of $\Delta_{\boldsymbol{x}}$, since the lexicographical order (in this case) gives a total ordering of the facets.

We now turn to our main problem: writing down a formula for each component of $\Sigma^1_{\boldsymbol{n},s}$, which we recall from (5.5) is the sum of all tensors in $\mathcal{T}^1_{\boldsymbol{n},s}$. To this end, note that a K-basis for $K[\Delta_{\boldsymbol{n}}]$ is given by the monomials whose support is a chain in $\Pi_{\boldsymbol{n}}$. Restricting to the degree-s component, we observe the bijection

$$\begin{aligned}\mathcal{T}^1_{\boldsymbol{n},s} &\longleftrightarrow K\text{-basis of } K[\Delta_{\boldsymbol{n}}]_s,\\ T &\longleftrightarrow \prod_{\boldsymbol{x}\in\Pi_{\boldsymbol{n}}} z_{\boldsymbol{x}}^{T[\boldsymbol{x}]}.\end{aligned}$$

Under this correspondence, adding tensors corresponds to multiplying monomials. Therefore, letting m range over the monomials, we have

$$\prod_{m\in K[\Delta_{\boldsymbol{n}}]_s} m = \prod_{\boldsymbol{x}\in\Pi_{\boldsymbol{n}}} z_{\boldsymbol{x}}^{\Sigma^1_{\boldsymbol{n},s}[\boldsymbol{x}]}. \tag{5.14}$$

Hence our main problem is equivalent to finding the exponent of each indeterminate $z_{\boldsymbol{x}}$ in the product of monomials on the left-hand side of (5.14). We do this in Section 5.6. Before that, however, we must explain and exploit the fact that the h-polynomial of $\Delta_{\boldsymbol{x}}$ is actually a well-known object called the *multiset Eulerian polynomial*.

5.5 Multiset Eulerian polynomials

Let $\mathbf{p} = (p_1, \ldots, p_d)$. A *multipermutation* of the multiset $\{1^{p_1}, 2^{p_2}, \ldots, d^{p_d}\}$ is a word $\pi = \pi_1 \cdots \pi_{|\mathbf{p}|}$ in which i appears exactly p_i times, for each $1 \le i \le d$. Let $\mathfrak{S}_{\mathbf{p}}$ be the set of all such multipermutations. A *descent* of a multipermutation π is an index i such that $\pi_i > \pi_{i+1}$. Let $\operatorname{des}(\pi)$ denote the number of descents of π. Then the *multiset Eulerian polynomial* $A_{\mathbf{p}}(t)$ is defined to be

$$A_{\mathbf{p}}(t) := \sum_{\pi\in\mathfrak{S}_{\mathbf{p}}} t^{\operatorname{des}(\pi)}.$$

(The special case $A_{\mathbf{1}}(t)$ is just the dth Eulerian polynomial $A_d(t)$, i.e., the descent-generating function over the symmetric group S_d. See, for example, [68, p. 22], although the convention there is to multiply through by t.) The multiset Eulerian polynomial occurs as the numerator of the following generating function, due to MacMahon [54, p. 211]:

$$\frac{A_{\mathbf{p}}(t)}{(1-t)^{|\mathbf{p}|+1}} = \sum_{\ell=0}^{\infty}\prod_{i=1}^{d}\binom{p_i+\ell}{\ell}t^{\ell}. \tag{5.15}$$

From (5.15) we can obtain an explicit expression for the coefficients in $A_{\mathbf{p}}(t)$; see also [1, 19] for combinatorial proofs of the formula below. (These coefficients are known as "Simon Newcomb numbers.") Writing $[t^k]$ for the coefficient of t^k, we have

$$[t^k]A_{\mathbf{p}}(t) = \#\Big\{\pi \in \mathfrak{S}_{\mathbf{p}} : \mathrm{des}(\pi) = k\Big\} = \sum_{\ell=0}^{k}(-1)^{\ell}\binom{|\mathbf{p}|+1}{\ell}\prod_{i=1}^{d}\binom{p_i+k-\ell}{k-\ell}. \tag{5.16}$$

It is shown in [19, Lemma 2] that the maximum number of descents, i.e., the degree of $A_{\mathbf{p}}(t)$, equals

$$|\mathbf{p}| - \max(\mathbf{p}). \tag{5.17}$$

For example, if $\mathbf{p} = (3, 2, 4)$, then $A_{\mathbf{p}}(t) = 24t^5 + 260t^4 + 580t^3 + 345t^2 + 50t + 1$. This is computed directly via (5.16), and we verify (5.17) by observing that the degree is indeed $|\mathbf{p}| - \max(\mathbf{p}) = 9 - 4 = 5$.

Lemma 5.3. *The h-polynomial $h_{\boldsymbol{x}}(t)$ of $\Delta_{\boldsymbol{x}}$ is the multiset Eulerian polynomial $A_{\boldsymbol{x}-\mathbf{1}}(t)$.*

Proof. Recall the labeling λ in (5.13), which induces a shelling of $\Delta_{\boldsymbol{x}}$. With respect to λ, the label sequence of each facet $\mathbf{1} \lessdot \cdots \lessdot \boldsymbol{x}$ contains $x_i - 1$ copies of the label i, for each $i = 1, \ldots, d$; conversely, each possible permutation of these labels (where the copies of each i are indistinguishable from each other) is the label sequence of a unique facet. Hence we have a bijection between the set of facets of $\Delta_{\boldsymbol{x}}$ and the set $\mathfrak{S}_{\boldsymbol{x}-\mathbf{1}}$. By (5.9), the size of $\mathrm{R}(F)$ equals the number of descents in F, which in turn equals the number of descents in the label sequence of F. Therefore, comparing (5.7) and (5.16), it is clear that $h_{\boldsymbol{x}}(t) = A_{\boldsymbol{x}-\mathbf{1}}(t)$. ☐

Remark. Combining (5.7) with the Stanley decomposition (5.10), it is easy to see that the Hilbert series of a Stanley–Reisner ring $K[\Delta]$ is given by

$$\frac{h_{\Delta}(t)}{(1-t)^{\#F}},$$

where F is any facet of Δ (since Δ is assumed to be pure). Thus by Lemma 5.3 and (5.12), and by MacMahon's expansion (5.15), our particular ring $K[\Delta_{\boldsymbol{n}}]$ has the Hilbert series

$$\frac{A_{\boldsymbol{n}-\mathbf{1}}(t)}{(1-t)^{|\boldsymbol{n}|-d+1}} = \sum_{\ell=0}^{\infty}\prod_{i=1}^{d}\binom{n_i+\ell-1}{\ell}t^{\ell}.$$

This is also the Hilbert series of the coordinate ring of the set of simple (also called pure, or decomposable) tensors in $\mathcal{T}_{\boldsymbol{n}}$, which is isomorphic to $K[\Delta_{\boldsymbol{n}}]$. See [57, Thm. 5], where this ring is also viewed as the toric ring defined by the Segre embedding of $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_d}$.

Combining (5.17) with Lemma 5.3, we note that the degree of $h_{\boldsymbol{x}}(t) = A_{\boldsymbol{x}-\mathbf{1}}(t)$ equals

$$|\boldsymbol{x} - \mathbf{1}| - \max(\boldsymbol{x} - \mathbf{1}) = |\boldsymbol{x}| - \max(\boldsymbol{x}) - d + 1. \tag{5.18}$$

Equivalently, (5.18) is the maximum size of $\mathrm{R}(F)$, taken over all facets F of $\Delta_{\boldsymbol{x}}$.

5.6 Main result, second version

Theorem 5.4. *For each $\boldsymbol{x} \in \Pi_{\boldsymbol{n}}$, we have*

$$\Sigma^1_{\boldsymbol{n},s}[\boldsymbol{x}] = \sum_{k=0}^{\min\{\omega(\boldsymbol{n},\boldsymbol{x}),\, s-1\}} \binom{|\boldsymbol{n}| - d + s - k}{s - k - 1} \cdot [t^k]\, A_{\boldsymbol{x}-\mathbf{1}}(t) A_{\boldsymbol{n}-\boldsymbol{x}}(t),$$

where $\omega(\boldsymbol{n}, \boldsymbol{x}) := |\boldsymbol{n}| - \max(\boldsymbol{x}) - \max(\boldsymbol{n} - \boldsymbol{x}) - d + 1$.

We record two key lemmas before giving the proof of Theorem 5.4:

Lemma 5.5. *Let $\boldsymbol{x} \in \Pi_n$, and let $F \ni \boldsymbol{x}$ be a facet of $\Delta_{\boldsymbol{n}}$. Then $\binom{|\boldsymbol{n}|-d+s-k}{s-k-1}$ equals the exponent of $z_{\boldsymbol{x}}$ in the product of all monomials in*

$$K[F]_{s-k-1} z_{\mathrm{R}(F)\cup\{\boldsymbol{x}\}}. \tag{5.19}$$

Proof. It suffices to show that

$$\binom{|\boldsymbol{n}| - d + s - k}{s - k - 1} = \Big(\#\text{ monomials in (5.19)}\Big)\Big(\text{average exponent of } z_{\boldsymbol{x}} \text{ in each monomial}\Big).$$

The number of monomials in (5.19) equals the number of monomials in $K[F]_{s-k-1}$. This number, in turn, equals the number of weak compositions of the degree $s - k - 1$ into $\#F$ many parts. Thus, recalling from (5.12) that $\#F = |\boldsymbol{n}| - d + 1$ for any facet F of $\Delta_{\boldsymbol{n}}$, and using the elementary formula (5.6), we have

$$\#\text{ monomials in (5.19)} = \binom{s - k - 1 + |\boldsymbol{n}| - d}{|\boldsymbol{n}| - d}. \tag{5.20}$$

The average exponent of $z_{\boldsymbol{x}}$, taken over all the monomials in $K[F]_{s-k-1}$, equals the degree $s-k-1$ divided by the number of variables $\#F$. Adding 1 to this average to account for the factor of $z_{\boldsymbol{x}}$ present in $z_{\mathrm{R}(F)\cup\{\boldsymbol{x}\}}$ in (5.19), we obtain

$$\text{average exponent of } z_{\boldsymbol{x}} \text{ in each monomial} = 1 + \frac{s-k-1}{|\boldsymbol{n}|-d+1} = \frac{|\boldsymbol{n}|-d+s-k}{|\boldsymbol{n}|-d+1}. \tag{5.21}$$

Multiplying the expressions in (5.20) and (5.21), we obtain

$$\binom{s-k-1+|\boldsymbol{n}|-d}{|\boldsymbol{n}|-d} \cdot \frac{|\boldsymbol{n}|-d+s-k}{|\boldsymbol{n}|-d+1} = \binom{|\boldsymbol{n}|-d+s-k}{|\boldsymbol{n}|-d+1} = \binom{|\boldsymbol{n}|-d+s-k}{s-k-1}. \qquad \square$$

Lemma 5.6. *For $\boldsymbol{x} \in \Pi_{\boldsymbol{n}}$, the coefficient*

$$[t^k]\, A_{\boldsymbol{x}-\boldsymbol{1}}(t) A_{\boldsymbol{n}-\boldsymbol{x}}(t)$$

equals the number of facets $F \ni \boldsymbol{x}$ of $\Delta_{\boldsymbol{n}}$, such that $\#\big(\mathrm{R}(F) \setminus \{\boldsymbol{x}\}\big) = k$.

Proof. Every facet $F \ni \boldsymbol{x}$ of $\Delta_{\boldsymbol{n}}$ can be written uniquely as the union of two saturated chains

$$F' : \boldsymbol{1} \lessdot \cdots \lessdot \boldsymbol{x} \qquad \text{and} \qquad F'' : \boldsymbol{x} \lessdot \cdots \lessdot \boldsymbol{n},$$

which intersect only at $\boldsymbol{x}$. Then F' can be any facet of $\Delta_{\boldsymbol{x}}$, viewed as a subposet of $\Delta_{\boldsymbol{n}}$. Likewise, F'' can be any facet of $\Delta_{\boldsymbol{n}+\boldsymbol{x}-\boldsymbol{1}}$, viewed as a subposet of $\Delta_{\boldsymbol{n}}$ after translating coordinates by $\boldsymbol{x}-\boldsymbol{1}$. Therefore $\#(\mathrm{R}(F) \setminus \{\boldsymbol{x}\}) = \mathrm{R}(F') + \mathrm{R}(F'')$. Thus by (5.7), we have

$$\begin{aligned} h_{\boldsymbol{x}}(t) h_{\boldsymbol{n}-\boldsymbol{x}+\boldsymbol{1}}(t) &= \left(\sum_{F'} t^{\#\mathrm{R}(F')}\right)\left(\sum_{F''} t^{\#\mathrm{R}(F'')}\right) \\ &= \sum_{F',F''} t^{\#\mathrm{R}(F')+\#\mathrm{R}(F'')} \\ &= \sum_{F \ni \boldsymbol{x}} t^{\#(\mathrm{R}(F)\setminus\{\boldsymbol{x}\})}, \end{aligned}$$

where the sums range over facets F, F', and F'' of $\Delta_{\boldsymbol{n}}$, $\Delta_{\boldsymbol{x}}$, and $\Delta_{\boldsymbol{n}-\boldsymbol{x}+\boldsymbol{1}}$, respectively. By Lemma 5.3, we can rewrite $h_{\boldsymbol{x}}(t) h_{\boldsymbol{n}-\boldsymbol{x}+\boldsymbol{1}}(t)$ as $A_{\boldsymbol{x}-\boldsymbol{1}}(t) A_{\boldsymbol{n}-\boldsymbol{x}}(t)$. $\square$

Proof of Theorem 5.4. By (5.14), we know that $\Sigma^1_{\boldsymbol{n},s}[\boldsymbol{x}]$ equals the exponent of $z_{\boldsymbol{x}}$ in the product of all monomials in the graded component $K[\Delta_{\boldsymbol{n}}]_s$, which by (5.11) has the decomposition

$$K[\Delta_{\boldsymbol{n}}]_s = \bigoplus_k \bigoplus_{\substack{F: \\ \#\mathrm{R}(F)=k}} K[F]_{s-k}\, z_{\mathrm{R}(F)}, \tag{5.22}$$

where the F's in the inside sum are facets of $\Delta_{\boldsymbol{n}}$. The outside sum in (5.22) ranges from $k = 0$ to $k = \min\{|\boldsymbol{n}| - \max(\boldsymbol{n}) - d + 1,\ s\}$; this follows from (5.18) and from the fact that the degree $s - k$ must be nonnegative. Obviously, the only monomials contributing to the exponent of $z_{\boldsymbol{x}}$ are those divisible by $z_{\boldsymbol{x}}$; hence we may ignore all summands in (5.22) such that $\boldsymbol{x} \notin F$. If $\boldsymbol{x} \in F$, then the subspace of $K[F]_{s-k}$ spanned by the monomials divisible by $z_{\boldsymbol{x}}$ is

$$K[F]_{s-k-1} z_{\boldsymbol{x}}.$$

Then since $\boldsymbol{x}$ may or may not lie in $\mathrm{R}(F)$, the subspace of $K[F]_{s-k} z_{\mathrm{R}(F)}$ spanned by monomials divisible by $z_{\boldsymbol{x}}$ is

$$K[F]_{s-k-1} z_{\boldsymbol{x}} z_{\mathrm{R}(F)\setminus\{\boldsymbol{x}\}} = K[F]_{s-k-1} z_{\mathrm{R}(F)\cup\{\boldsymbol{x}\}}.$$

Combining this with (5.22), we conclude that $\Sigma^1_{\boldsymbol{n},s}[\boldsymbol{x}]$ equals the exponent of $z_{\boldsymbol{x}}$ in the product of all monomials in

$$\bigoplus_{k} \bigoplus_{\substack{F \ni \boldsymbol{x}: \\ \#(\mathrm{R}(F)\setminus\{\boldsymbol{x}\}) = k}} K[F]_{s-k-1} z_{\mathrm{R}(F)\cup\{\boldsymbol{x}\}}. \tag{5.23}$$

Now applying Lemma 5.5 and Lemma 5.6 to (5.23), we see that the desired exponent of $z_{\boldsymbol{x}}$ equals

$$\begin{aligned} \Sigma^1_{\boldsymbol{n},s}[\boldsymbol{x}] &= \sum_k \sum_{\substack{F \ni \boldsymbol{x}: \\ \#(\mathrm{R}(F)\setminus\{\boldsymbol{x}\}) = k}} \binom{|\boldsymbol{n}| - d + s - k}{s - k - 1} \\ &= \sum_k \binom{|\boldsymbol{n}| - d + s - k}{s - k - 1} \cdot [t^k]\, A_{\boldsymbol{x}-\mathbf{1}}(t) A_{\boldsymbol{n}-\boldsymbol{x}}(t), \end{aligned}$$

where the nonzero summands are those for which k is less than or equal to both $s - 1$ (otherwise the binomial coefficient is zero) and the degree of $A_{\boldsymbol{x}-\mathbf{1}}(t) A_{\boldsymbol{n}-\boldsymbol{x}}(t)$. This degree is easily computed to be $\omega(\boldsymbol{n}, \boldsymbol{x})$, using (5.18). □

Remark. It is not obvious that Theorem 5.4 specializes to the two–dimensional formula in our previous work [28, Thm. 5.1]. This is because in the case $d = 2$, it was straightforward to write down explicitly the coefficient of t^k in the product of two multiset Eulerian polynomials, since these polynomials could be expressed without signs (see [28, Lemma 4.2]).

Part II

Parking functions

6 BACKGROUND ON PARKING FUNCTIONS

In this second part, we highlight results in connection with *parking functions*. We begin by providing necessary definitions, prerequisites and well-known results connecting parking functions to various mathematical objects. The study of parking functions, while interesting in its own right, receives further motivation due to its application across a wide area of mathematics, as we will see for example in Section 6.4, where we show a connection to representation theory through the famous *shuffle conjecture*.

In Chapter 7, we will provide enumerative formulas for two different generalizations of parking functions by approaching the problem "backwards" – examining the order of parked cars and computing the number of preferences leading to the given parking order. The generalizations examined add additional realistic context by allowing cars to have different lengths.

In Chapter 8, we reveal a surprising connection between parking functions of length n and the *Quicksort* algorithm. We impose constraints on the outcome and the *lucky* statistic on parking functions and establish equinumerosity between the resulting parking objects and comparisons performed by the *Quicksort* algorithm when sorting all permutations of n letters.

Finally, in Chapter 9, we provide a complete characterisation and enumerative formulas for the *Boolean intervals* in the *right weak Bruhat order* of S_n. We prove our results by bijecting to the intersection of *Fubini rankings* and *unit interval parking functions*, which, as the name indicates, are a subset of parking functions.

6.1 Prerequisites and definitions

Consider a one-way street with n parking spots, and n cars labeled $c_1, \ldots, c_n$, which enter the street one by one. Each car c_i has a preferred spot $p_i \in [n]$. When a car enters the street, it will first drive to its preferred spot to attempt to park. However, if the spot is occupied, the car continues driving along the one-way street until it finds an empty spot. If there is no empty spot before the end of the street, the car fails to park. The preference list $p = (p_1, \ldots, p_n) \in [n]^n$ containing the preferred spots for all cars is called a *parking function* if all cars are able to park on the street. We denote the set of all parking functions with n cars PF_n.

Parking functions were first introduced by Konheim and Weiss in 1966 [51]. While studying hashing using *linear probing* to handle collisions, they put the problem into the context of parking on a one-way street. One of their results was that $\#\mathrm{PF}_n = (n+1)^{n-1}$. Ever since, parking functions have received an immense amount of attention in combinatorial research and have been examined in vastly different research contexts such as polyhedral combinatorics [2, 4], hyperplane arrangements [66], the *Quicksort* algorithm [41] and the Tower of Hanoi game [76]. For a comprehensive survey of parking functions, we recommend [75].

Example 6.1. Figure 15 shows an example of parking functions and the respectively resulting parking orders through the preference lists $p = (2, 1, 1, 3)$ and $q = (2, 1, 4, 4)$.

(a) $p = (2, 1, 1, 3)$ (b) $q = (2, 1, 4, 4)$

Figure 15: Comparison of the preference lists $p = (2, 1, 1, 3)$ and $q = (2, 1, 4, 4)$. We can see that with preferences p, all cars (shown as gray squares) are able to park in the spots on the street (shown as white squares). Therefore $p \in \mathrm{PF}_4$. However, with preferences q, car 4 is unable to park, hence $q \notin \mathrm{PF}_4$.

An interesting result that is important for our study is the fact that PF_n is invariant under permutation. Denote by S_n the symmetric group, then S_n acts on PF_n by permuting preferences: that is, for all $p \in \mathrm{PF}_n$ and $\pi \in \mathrm{S}_n$, $\pi(p) = (p_{\pi(1)}, \ldots, p_{\pi(n)}) \in \mathrm{PF}_n$. Permuting the elements within a parking function allows us to state the following result, which is sometimes used as an alternative definition of parking functions [74].

Theorem 6.2. *A vector $p \in [n]^n$ is a parking function or order n if and only if its weakly increasing rearrangement, p', satisfies $p'_i \leq i$ for all $i \in [n]$.*

Proof. First, let $p \in \mathrm{PF}_n$ and let p' be its weakly increasing rearrangement. Note, that p' will park all cars in order: that is, c_1 will park in spot 1, c_2 will park in spot 2, and so on. Assume there exists i such that $p'_i > i$. Then, there will be no car parking in spot i and consequently one car that will not be able to park. Therefore, $p'_i \leq i$ must be true. To show the converse, we pick as p' the weakly increasing vector that satisfies $p'_i = i$ for all $i \in [n]$. Clearly, p' is a parking function which allows each car to park in its preferred spot. Additionally, we note that we can arbitrarily decrease the preference of each car (potentially sorting p' again to remain weakly increasing) without "pushing" a car off the street. The result then follows from the permutation invariance of parking functions. □

Before giving a brief survey of several well–established results associated with parking functions, we state an additional definition that will be useful to us later.

Definition 6.3. Let $p \in \mathrm{PF}_n$. Then, we define $\mathcal{O} : \mathrm{PF}_n \to \mathrm{S}_n$ with $\mathcal{O}((p_1, \ldots, p_n)) = \pi_1 \cdots \pi_n \in \mathrm{S}_n$ (in one-line notation), to be the *outcome map*. That is, $\mathcal{O}(p)$ yields the parking order of cars $c_1, \ldots, c_n$ with preferences $p_1, \ldots, p_n$.

6.2 A connection to lattice paths

An important element in Chapter 3 is the concept of a lattice path, or in the context of the problem: a chain in the poset $n_1 \times n_2$, for $n_1, n_2 \in \mathbb{N}_{>0}$. Interestingly, lattice paths additionally play an important role in the world of parking functions. Let $n \in \mathbb{N}_{>0}$, then a *Dyck path* of order n is an $n \times n$ lattice path consisting of n north ("N") and n east ("E") steps, while never crossing below the diagonal $y = x$.

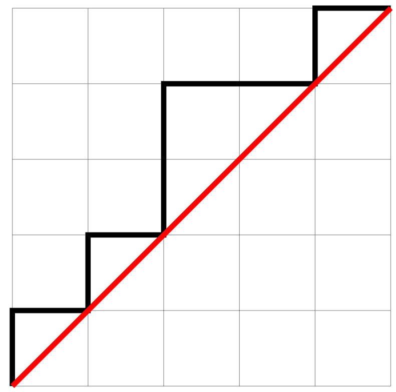


Figure 16: A 5×5 Dyck path. The path has to stay above the red diagonal ($y = x$). The Dyck path can be expressed as the string "NENENNEENE", listing the order of north and east steps.

The following well–known result connects parking functions to Dyck paths.

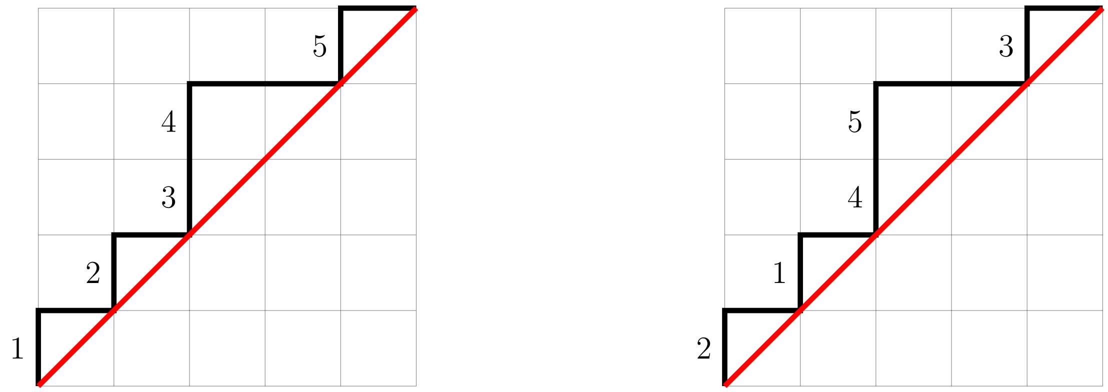
Theorem 6.4. *The set of weakly increasing parking functions with n cars $\mathrm{PF}_n^{\uparrow}$ is in one-to-one correspondence with the set of $n \times n$ Dyck paths.*

Theorem 6.4 implies that the cardinality of $\mathrm{PF}_n^{\uparrow}$ is given by the nth *Catalan number*[1]:

$$\#\mathrm{PF}_n^{\uparrow} = C_n, \text{ where } C_n = \frac{1}{n+1}\binom{2n}{n}.$$

While Theorem 6.4 only related Dyck paths to the set of weakly increasing parking functions $\mathrm{PF}_n^{\uparrow}$, we can extend this correspondence to PF_n by labeling the north steps of Dyck paths with the numbers $1, \ldots, n$, assigning consecutive north steps increasing labels. An example of two such *labeled* Dyck paths is illustrated in Figure 17.

[1] Over 200 different combinatorial objects are enumerated by the Catalan numbers. For an overview, see [69].



(a) Labeled Dyck path corresponding to parking function $(1, 2, 3, 3, 5)$

(b) Labeled Dyck path corresponding to parking function $(2, 1, 5, 3, 3)$.

Figure 17: Two *labeled* Dyck paths. In Figure 17a, we see a Dyck path corresponding to a weakly increasing parking function. In Figure 17b, we see how we can permute the labels to create a correspondence to a not-weakly-increasing parking function.

The following theorem states the corresponding result.

Theorem 6.5. *The set of labeled $n \times n$ Dyck paths is in one-to-one correspondence with the set of parking functions with n cars.*

There are many other combinatorial objects that allow a bijection to parking functions. For a comprehensive overview, we recommend [67, 75].

6.3 Statistics on parking functions

When it comes to parking functions, it is not only interesting to study their count or their relation to different combinatorial objects. It is also of interest to examine what kinds of properties parking functions can satisfy. This is where the study of discrete *statistics* on parking functions resides.

Statistics that are explored on permutations can be extended to parking functions, which allows for the further study of *ascents*, *descents*, *peaks* and *ties*.

Definition 6.6. Let $p \in [n]^n$. Then, we say p has

- an *ascent* at i if $p_i < p_{i+1}$ for $1 \leq i < n$,

- a *descent* at i if $p_i > p_{i+1}$ for $1 \leq i < n$,
- a *peak* at i if $p_{i-1} < p_i > p_{i+1}$ for $1 < i < n$, and
- a *valley* at i if $p_{i-1} > p_i < p_{i+1}$ for $1 < i < n$.

We denote the set of all ascents of p as $\mathrm{Asc}(p)$ and the set of all descents of p as $\mathrm{Des}(p)$.

See [5] for a study of peaks in permutations and [18] for results on these statistics in randomized parking functions.

In this work, the two statistics that are most important are the number of *lucky* cars and the *displacement* of a parking function.

Definition 6.7 (Lucky cars)**.** Let $p \in \mathrm{PF}_n$ and $\mathcal{O}(p) = \pi_1 \cdots \pi_n$. Then, we define $\mathrm{L} : \mathrm{PF}_n \to \{0, \dots, n\}$ by $\mathrm{L}(p) = \sum_{i=1}^{n} \delta_{i,p_{\pi_i}}$ where $\delta_{x,y}$ the Kronecker delta. In other words, $\mathrm{L}(p)$ is the number of cars c_i that park in their preferred spot p_i.

Definition 6.8 (Displacement)**.** Let $p \in \mathrm{PF}_n$ and $\mathcal{O}(p) = \pi_1 \cdots \pi_n$. Then, we define $\mathrm{d}(p) : \mathrm{PF}_n \to \left\{0, \dots, \frac{n(n-1)}{2}\right\}$ by $\mathrm{d}((p_1, \dots, p_n)) = \sum_{i=1}^{n}(\pi_i - p_{\pi_i})$. Note, that the maximum displacement is incurred by $p = (1, 1, \dots, 1)$ with $\mathrm{d}(p) = \frac{n(n-1)}{2}$; the minimum displacement of 0 is attained by any permutation in PF_n.

We refer to Chapter 8 for a result involving the lucky statistic and to Section 6.5 for a result involving the displacement of parking functions.

Moreover, we provide a further statistic making use of the Dyck paths associated with parking functions. This allows us to make a connection between parking functions and representation theory in Section 6.4.

Definition 6.9 (Diagonal inversions)**.** Let $p \in \mathrm{PF}_n$, d be the Dyck path associated with p and consider a pair of cars $\{c_i, c_j\}$ in p. We call $\mathrm{dist}(c_k)$ the distance of the square in d associated with c_k to the diagonal $y = x$. The pair $\{c_i, c_j\}$ is a

- *primary diagonal inversion* if $\mathrm{dist}(c_i) = \mathrm{dist}(c_j)$ and $\max\{c_i, c_j\}$ occurring farther right.

- *secondary diagonal inversion* if $|\text{dist}(c_i) - \text{dist}(c_j)| = 1$ and $\max\{c_i, c_j\}$ is on the left of and in the higher diagonal than $\min\{c_i, c_j\}$.

We define the set of primary diagonal inversions in p as $\text{dinv}_1(p)$, the set of secondary diagonal inversions as $\text{dinv}_2(p)$ and lastly

$$\text{dinv}(p) = \#\text{dinv}_1(p) + \#\text{dinv}_2(p).$$

We provide an example to illustrate the statistics described in this section.

Example 6.10. Consider $p = (4, 1, 2, 5, 4, 1, 7)$. Clearly, $p \in \text{PF}_7$ and $\mathcal{O}(p) = 2361457$. The parking function p has

- ascent set $\text{Asc}(p) = \{2, 3, 6\}$
- descent set $\text{Des}(p) = \{2, 3, 6\}$
- a peak at $\{4\}$
- valleys at $\{2, 6\}$.

Furthermore, we count $\text{L}(p) = 5$ lucky cars and a displacement of $\text{d}(p) = 4$. The Dyck path corresponding to p is pictured below.

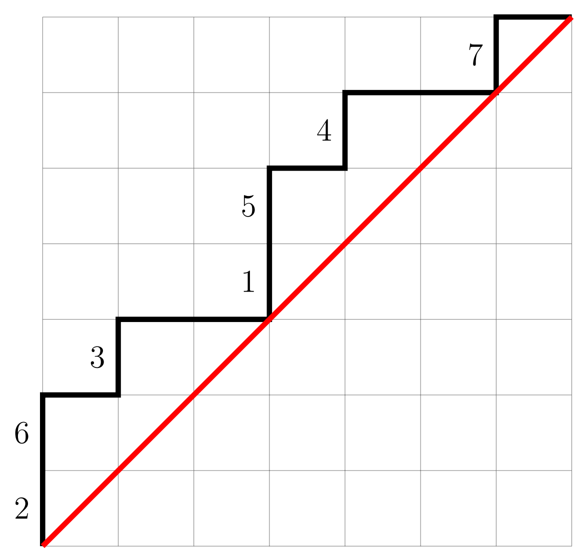


Figure 18: Dyck path associated with the parking function $p = (4, 1, 2, 5, 4, 1, 7)$.

We identify as primary diagonal inversions

$$\mathrm{dinv}_1(p) = \{\{2,7\},\{1,7\},\{3,5\},\{3,4\}\}\,.$$

The secondary diagonal inversions are

$$\mathrm{dinv}_2(p) = \{\{1,6\},\{1,3\}\}\,,$$

therefore $\mathrm{dinv}(p) = 4 + 2 = 6$.

6.4 A connection to the representation theory of S_n

In this section, we will more closely examine another object related to parking functions: the space of *diagonal harmonics*, $\mathcal{DH}_n$. Haiman [40] proved in 2002 that

$$\dim\left(\mathcal{DH}_n\right) = (n+1)^{n-1},$$

which prompted a closer examination of other objects enumerated by $(n+1)^{n-1}$, such as parking functions. The research effort eventually led to the formulation of the famous *shuffle conjecture* by Haglund et al. [37], which describes the *bigraded Frobenius characteristic* of $\mathcal{DH}_n$ in terms of statistics on parking functions. The conjecture was later proved by Carlsson and Mellit [14]. We include it here by way of preview, deferring to Section 6.4.3 for the details:

$$\mathcal{DH}_n\left[Z;q,t\right] = \sum_{p\in\mathrm{PF}_n} t^{\mathrm{area}(p)} q^{\mathrm{dinv}(p)} F_{n,\mathrm{ides}(p)}.$$

We provide elementary background on the result, following [74] and [38] in exposition. For a more detailed historical recount, we refer to [74].

6.4.1 Diagonal harmonics

Haiman [38] introduced the space of diagonal harmonics in his work describing the quotient ring

$$R_n = \mathbb{Q}\left[X_n, Y_n\right]/I, \tag{6.1}$$

where $X_n = \{x_1, \ldots, x_n\}$, $Y_n = \{y_1, \ldots, y_n\}$ and I is the ideal generated by all S_n-invariant polynomials in $\mathbb{Q}[X_n, Y_n]$. Using the symmetric bilinear form $\langle \cdot, \cdot \rangle$ on $\mathbb{Q}[X_n, Y_n]$ with

$$\langle f, g \rangle = f(\partial X_n, \partial Y_n) g(X_n, Y_n)|_{X=Y=0}$$

and the shorthand $f(\partial X_n, \partial Y_n) = f(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}, \frac{\partial}{\partial y_1}, \ldots, \frac{\partial}{\partial y_n})$, diagonal harmonics are defined as the *orthogonal complement* of the ideal I – that is,

$$\mathcal{DH}_n = \{f \in \mathbb{Q}[X_n, Y_n] \mid \langle f, g \rangle = 0 \text{ for all } g \in I\}.$$

A useful interpretation provided in [38, Proposition 1.3.1] is, that if the ideal I is regarded as a system of partial differential equations, $\mathcal{DH}_n$ is the space of its solutions.

Throughout, we use $\mathbb{C}$ instead of $\mathbb{Q}$ as the ground field. Garsia and Haiman mostly use $\mathbb{Q}$, which we freely change to $\mathbb{C}$ when citing their work. We provide the following equivalent definition of $\mathcal{DH}_n$, which is for example found in [36] and [74].

Definition 6.11 (Diagonal harmonics [74])**.** Let $X_n = \{x_1, \ldots, x_n\}$ and $Y_n = \{y_1, \ldots, y_n\}$. Then, we define the *space of diagonal harmonics* as the vector space of polynomials

$$\mathcal{DH}_n = \left\{ f(X_n, Y_n) \in \mathbb{C}[X_n, Y_n] \mid \sum_{i=1}^{n} \partial_{x_i}^a \partial_{y_i}^b f(X_n, Y_n) = 0 \text{ for all } a, b \geq 0 \text{ with } a + b > 0 \right\}.$$

The symmetric group S_n acts "diagonally" on $\mathcal{DH}_n$, i.e. by permuting both the variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ at once. Namely, for $\pi \in \mathrm{S}_n$ and $f \in \mathcal{DH}_n$, we have

$$\pi(f(X_n, Y_n)) = f(x_{\pi(1)}, \ldots, x_{\pi(n)}, y_{\pi(1)}, \ldots, y_{\pi(n)}) \in \mathcal{DH}_n.$$

This allows us to decompose $\mathcal{DH}_n$ into a bigraded S_n module:

$$\mathcal{DH}_n = \bigoplus_{c,d \geq 0} \mathcal{DH}_n^{c,d}.$$

In fact, the same is true for the ring R_n described in Equation 6.1: the ideal I can similarly be decomposed into bigraded homogeneous components, making R_n a doubly graded ring. Through the same S_n-action, we receive the decomposition

$$R_n = \bigoplus_{c,d \geq 0} R_n^{c,d},$$

with each component $R_n^{c,d}$ being isomorphic to the corresponding $\mathcal{DH}_n^{c,d}$ (see [38] for more details).

Following Haiman [38], we construct a *Frobenius series* or *bigraded Frobenius characteristic* [74]:

$$\mathcal{DH}_n\,[Z;q,t] = \sum_{c,d\geq 0} t^c q^d \phi(\mathcal{DH}_n^{c,d}), \tag{6.2}$$

with ϕ being the *Frobenius map* defined by

$$\phi(\mathcal{DH}_n^{c,d}) = \sum_{\lambda\vdash n} s_\lambda \mathrm{mult}(\chi^\lambda, \mathrm{char}(\mathcal{DH}_n^{c,d})). \tag{6.3}$$

Here, s_λ is a *Schur polynomial* in the variables $Z = \{z_1, z_2, \ldots\}$. Equation (6.2) assigns to each irreducible S_n-character χ^λ a corresponding Schur polynomial. The second factor in (6.3), $\mathrm{mult}(\chi^\lambda, \mathrm{char}(\mathcal{DH}_n^{c,d}))$, gives the multiplicity of χ^λ in the character of $\mathcal{DH}_n^{c,d}$ under the S_n-action.

Garsia and Haiman simplified the Frobenius characteristic by expressing it in terms of the characteristic of *Garsia–Haiman modules.* We will give an overview of Garsia–Haiman modules next.

6.4.2 Garsia–Haiman modules

In an effort to prove the *Macdonald positivity conjecture*, Garsia and Haiman [33] constructed what is now known as Garsia–Haiman modules. Garsia–Haiman modules, referred to as $\mathcal{H}_\lambda$ with λ a partition, are a subspace of $\mathcal{DH}_n$ for all λ. Thus, using findings about $\mathcal{H}_\lambda$, Garsia and Haiman were able to express the Frobenius characteristic of $\mathcal{DH}_n$ in terms of the *nabla operator* ∇, see Definition 6.15.

Definition 6.12 ([33])**.** Let λ be a partition of $n \in \mathbb{N}_{>0}$, and take P to be the Young diagram obtained from λ. Label the coordinates of the n boxes in P with $\{(p_1, q_1), \ldots, (p_n, q_n)\}$, with

p_i and q_i the row and column coordinates (beginning with 0). Let

$$\Delta_\lambda(x_1,\ldots,x_n,y_1,\ldots,y_n) = \det \begin{bmatrix} x_1^{p_1}y_1^{q_1} & x_2^{p_1}y_2^{q_1} & \cdots & x_n^{p_1}y_n^{q_1} \\ x_1^{p_2}y_1^{q_2} & x_2^{p_2}y_2^{q_2} & \cdots & x_n^{p_2}y_n^{q_2} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{p_n}y_1^{p_n} & x_2^{p_n}y_2^{q_n} & \cdots & x_n^{p_n}y_n^{q_n} \end{bmatrix}. \tag{6.4}$$

Then, the Garsia–Haiman module is defined as

$$\mathcal{H}_\lambda = \mathbb{C}\left[X_n, Y_n\right]/I_\lambda,$$

where I_λ is the ideal

$$I_\lambda = \langle p(X_n, Y_n) \mid p(\partial X_n, \partial Y_n)\Delta_\lambda = 0\rangle.$$

Garsia and Haiman [33] define $\mathcal{H}_\lambda$ as the quotient $\mathbb{C}\left[X_n, Y_n\right]/I_\lambda$, where I_λ is the ideal generated by all polynomials $p(X_n, Y_n)$ such that corresponding the polynomial of differential operators $p(\partial X_n, \partial Y_n) = p(\frac{\partial}{\partial x_1},\ldots,\frac{\partial}{\partial x_n},\frac{\partial}{\partial y_1},\ldots,\frac{\partial}{\partial y_n})$ satisfies $p(\partial X, \partial Y)\Delta_\lambda = 0$. Equivalently, we view as $\mathcal{H}_\lambda$ the space of all derivatives of Δ_λ.

Example 6.13. Let $\lambda = (3, 2, 1)$. Then, we find the corresponding Young diagram, with squares labeled by row and column index, as:

(0, 0)	(0, 1)	(0, 2)
(1, 0)	(1, 1)	
(2, 0)		

.

From here, we find Δ_λ as

$$\Delta_{(3,2,1)}(x_1,\ldots,x_n,y_1,\ldots,y_n) = \det \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \\ y_1^2 & y_2^2 & y_3^2 & y_4^2 & y_5^2 & y_6^2 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ x_1y_1 & x_2y_2 & x_3y_3 & x_4y_4 & x_5y_5 & x_6y_6 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 & x_5^2 & x_6^2 \end{bmatrix}.$$

Now, the corresponding Garsia–Haiman module $\mathcal{H}_{(3,2,1)}$ is the space of all derivatives of the determinant $\Delta_{(3,2,1)}$.

Garsia and Haiman [33] conjectured, and Haiman later proved [39], that $\dim \mathcal{H}_\lambda = n!$ as well as that the bigraded Frobenius characteristic of $\mathcal{H}_\lambda$ is $\tilde{H}\left[Z;q,t\right]$, the modified Macdonald polynomial, which we define next.

Definition 6.14 ([40])**.** Let μ be a partition of $n \in \mathbb{N}_{>0}$. Then, define as the *transformed* or *modified Macdonald polynomial* the function

$$\tilde{H}_\mu\left[Z;q,t\right] = t^{n(\lambda)} J_\lambda \left[\frac{Z}{1-\frac{1}{t}};q,t\right].$$

Here, $n(\mu) = \sum_{i=1}^{\ell(\mu)} \mu_i(i-1)$ and the polynomial $J_\mu\left[Z;q,t\right]$ is the *integral form Macdonald polynomial* defined by Macdonald in [53, Chapter VI, Section 8].

Haiman [40] provides the equivalent definition of $\tilde{H}_\mu$ as

$$\tilde{H}_\mu\left[Z;q,t\right] = \sum_\lambda \tilde{K}_{\lambda\mu}(q,t) s_\lambda(Z),$$

where s_λ is the Schur polynomial and

$$\tilde{K}_{\lambda\mu}(q,t) = t^{n(\mu)} K_{\lambda\mu}(q,t^{-1}),$$

where $K_{\lambda\mu}(q,t)$ is the Kostka–Macdonald coefficient, see [53, Chapter VI, Section 8].

Modified Macdonald polynomials, just as Macdonald polynomials themselves, form a basis for the space of symmetric functions. For a detailed background, we refer to [53].

Inspired by the fact that $\mathcal{H}_\lambda \subset \mathcal{DH}_n$ for all $\lambda \vdash n$ (see [74]), Haiman proved that

$$\mathcal{DH}_n\left[Z;q,t\right] = \sum_{\lambda\vdash n} t^{n(\lambda)} q^{n(\lambda')} \mathcal{C}_\lambda \tilde{H}_\lambda\left[Z;q,t\right], \tag{6.5}$$

where λ' is the *transpose* of a partition λ, obtained by transposing its corresponding Young diagram and $\mathcal{C}_\lambda$ comes from the expression of *elementary symmetric functions* in terms of $\tilde{H}_\lambda\left[Z;q,t\right]$, namely

$$e_n = \sum_{\lambda\vdash n} \mathcal{C}_\lambda \tilde{H}_\lambda\left[Z;q,t\right]. \tag{6.6}$$

Before stating a final result of Garsia and Haiman for $\mathcal{DH}_n\left[Z;q,t\right]$, we need to define the nabla operator.

Definition 6.15 ([40])**.** Let ∇ be the *nabla operator*, defined as

$$\nabla \tilde{H}_\lambda [Z; q, t] = t^{n(\lambda)} q^{n(\lambda')} \tilde{H}_\lambda [Z; q, t] .$$

Combining Definition 6.15 with (6.5) and (6.6) yields

$$\mathcal{DH}_n [Z; q, t] = \nabla e_n. \tag{6.7}$$

In the next section, we will present the shuffle conjecture, which gives a formula for ∇e_n summing over parking functions.

6.4.3 The shuffle conjecture

The famous shuffle conjecture expresses ∇e_n, and therefore $\mathcal{DH}_n [Z; q, t]$ in terms of parking functions and *quasisymmetric functions*. Despite the proof by Carlsson and Mellit [14], the shuffle conjecture is still widely referred to as a conjecture. We begin by stating a few necessary definitions.

The first important property of a parking function is its i-descent set.

Definition 6.16 (i-descent set)**.** Let $p \in \mathrm{PF}_n$. Then, we define as the i-descent set of a parking function the set

$$\mathrm{ides}(p) = \{ i \in [n] \mid i + 1 \text{ is left of } i \text{ in } \mathcal{O}(p) \} .$$

Next, we recall the following definition.

Definition 6.17 (Quasisymmetric functions)**.** Let $S = \{s_1, \ldots, s_{\#S}\} \subseteq [n-1]$ for some $n \in \mathbb{N}_{>0}$. Then the *fundamental quasisymmetric function* $F_{n,S}$ is

$$F_{n,S} = \sum z_{i_1} \cdots z_{i_n},$$

summing over all tuples $(i_1, \ldots, i_n)$ such that $i_1 \leq \cdots \leq i_n$ and $i_j < i_{j+1}$ if $j \in S$.

Finally, we provide the necessary definition of the *area* of a parking function.

Definition 6.18 (Area)**.** Let $p \in \mathrm{PF}_n$ and let d the Dyck path associated with p. Then, we define as area(p) the number of complete squares below d that are above the diagonal $y = x$.

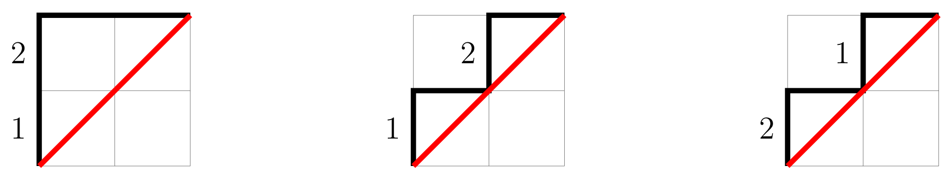
We can now state the shuffle conjecture:

Theorem 6.19 (The shuffle conjecture [37])**.**

$$\mathcal{DH}_n\left[Z; q, t\right] = \nabla e_n = \sum_{p \in \mathrm{PF}_n} t^{\mathrm{area}(p)} q^{\mathrm{dinv}(p)} F_{n,\mathrm{ides}(p)}.$$

Proof. See [14]. □

Example 6.20 ([74])**.** To illustrate the result, we set $n = 2$ and compute $\nabla e_2 = \mathcal{DH}_2\left[Z; q, t\right]$. First, we list the 3 elements in PF_2, together with their corresponding Dyck paths:



(a) $p_1 = (1, 1)$ (b) $p_2 = (1, 2)$. (c) $p_3 = (2, 1)$.

Figure 19: The elements in PF_n with their corresponding Dyck paths.

We compute the described statistics as

$$\mathrm{area}(p_1) = 1, \qquad \mathrm{dinv}(p_1) = 0, \qquad \mathrm{ides}(p_1) = \{1\}, \tag{6.8}$$

$$\mathrm{area}(p_2) = 0, \qquad \mathrm{dinv}(p_2) = 1, \qquad \mathrm{ides}(p_2) = \{1\}, \tag{6.9}$$

$$\mathrm{area}(p_3) = 0, \qquad \mathrm{dinv}(p_3) = 0, \qquad \mathrm{ides}(p_3) = \emptyset. \tag{6.10}$$

This yields

$$\nabla e_2 = \underbrace{tF_{2,\{1\}}}_{(6.8)} + \underbrace{qF_{2,\{1\}}}_{(6.9)} + \underbrace{F_{2,\emptyset}}_{(6.10)} = (q + t)F_{2,\{1\}} + F_{2,\emptyset}.$$

From (6.3) and (6.7), we know that we can express ∇e_n in terms of Schur functions. In turn, Schur functions can be defined in terms of fundamental quasisymmetric functions. Namely, for a partition $\lambda \vdash n$

$$s_\lambda = \sum_{T \in \mathrm{SYT}(\lambda)} F_{n,\mathrm{Des}(T)},$$

where $\mathrm{SYT}(\lambda)$ is the set of all *standard Young tableaux* (see [32]) and $\mathrm{Des}(T)$ is defined as the set of numbers i in T which appear in the same column or to the right of $i+1$. For the 2 tableaux corresponding to partitions of $n = 2$,

$$\begin{array}{|c|c|}\hline 1 & 2 \\ \hline \end{array} \text{ and } \begin{array}{|c|}\hline 1 \\ \hline 2 \\ \hline \end{array},$$

we get $s_{(2)} = F_{n,\emptyset}$ and $s_{(1,1)} = F_{n,\{1\}}$, which yields

$$\nabla e_2 = (q+t)s_{(1,1)} + s_{(2)}.$$

Remark. Let $p \in \mathrm{PF}_n$ for some $n \in \mathbb{N}_{>0}$. While the shuffle conjecture, in its original form, was stated in terms of the area of p, we observe, that

$$\mathrm{area}(p) = \mathrm{d}(p).$$

That is, the area of a parking function, counting the number of complete squares above $y = x$ in its corresponding Dyck path, is the equal to the *displacement* of the parking function, which we recall from Definition 6.8: Let $\pi = \mathcal{O}(p)$, then the displacment of p is

$$\mathrm{d}(p) = \sum_{i=1}^{n} (i - p_{\pi_i}).$$

This motivates a deeper exploration of the displacement in Section 6.5.

6.5 Displacement: EMD revisited

We use this section to show connections between the displacement or area statistics of parking functions and the EMD, which was the focus of the first part of this dissertation. The displacement of parking function requires some further examination. One important fact is the permutation–invariance of displacement, as for example shown in [22].

Lemma 6.21 ([22])**.** *Let $p = (p_1, \ldots, p_n) \in \mathrm{PF}_n$ and $\pi \in \mathrm{S}_n$. Then*

$$\mathrm{d}(p) = \mathrm{d}(\pi(p)),$$

where $\pi(p) = (p_{\pi(1)}, p_{\pi(2)}, \ldots, p_{\pi(n)}) \in \mathrm{PF}_n$.

Remark. As consequence of Lemma 6.21, the simplest formula to compute the displacement or area of a parking function $p \in \mathrm{PF}_n$ is

$$\mathrm{d}(p) = \frac{n(n+1)}{2} - \sum_{i=1}^{n} p_i.$$

Purely to establish a connection between the two parts of this dissertation, we proceed to highlight a way to compute the displacement of a parking function using the EMD.

When studying the displacement statistic, this property allows us to leave the context of parking functions to assume the perspective of histograms.

Definition 6.22. Let $\mathcal{H}(d, s)$ be the set of histograms with d units and s bins, for some $d, s \in \mathbb{N}_{>0}$. Now, for $n \in \mathbb{N}_{>0}$, we define

$$\varphi : \mathrm{PF}_n \to \mathcal{H}(n, n)$$

as

$$\varphi(p) = (\varphi_1(p), \ldots, \varphi_n(p)),$$

where

$$\varphi_i(p) = \sum_{k=1}^{n} \delta_{i,p_k}.$$

I.e., $\varphi_i(p)$ is equal to the number of cars in p that prefer spot i, making use of the indicator function Kronecker delta $\delta_{x,y}$.

Using Definition 6.22, we are able to create an injective map from the set of parking functions with n cars to the set of histograms with n bins and n units. Note, that the map φ, not unlike the the map d we defined as the displacement of parking functions, is invariant under the action of S_n. We require a further definition before formally stating the EMD between two parking functions.

Definition 6.23. Let $\lambda = (\lambda_1, \ldots, \lambda_n)$ be any histogram with n bins. Then, we define the *word* of λ as

$$w(\lambda) = \underbrace{1 \cdots 1}_{\lambda_1 \text{ times}} \underbrace{2 \cdots 2}_{\lambda_2 \text{ times}} \cdots \underbrace{n \cdots n}_{\lambda_n \text{ times}} .$$

Our main result will rely on this representation of a parking function as a histogram, as we will provide a formula for displacement in terms of a distance of histograms.
The metric that will provide a distance is the EMD.

Proposition 6.24. *Let $n \in \mathbb{N}_{>0}$. Then, for all $p \in \mathrm{PF}_n$:*

$$\mathrm{d}(p) = \mathrm{EMD}(\varphi(p), 1^{(n)}),$$

where $1^{(n)}$ is the histogram with n bins and one unit in each bin.

Proof. We recall the permutation-invariance of parking functions $p \in \mathrm{PF}_n$ as well as their displacement $\mathrm{d}(p)$. Together, they imply that the displacement of a parking function p is always equal to the displacement of the parking function p', which we define as the weakly increasing rearrangement of p. Additionally, note that the only weakly increasing parking function p_0 with $\mathrm{d}(p_0) = 0$ is the identity permutation, i.e. $p_0 = (1, 2, 3, 4, \dots, n)$, which also satisfies $\varphi(p_0) = 1^{(n)}$. Now, for any weakly increasing parking function $p' = (p'_1, \dots, p'_n)$, we have $\mathrm{d}(p') = \sum_{i=1}^{n} (i - p'_i)$.
Next, consider the problem from the perspective of the EMD, and write the parking function p' as well as the parking function that is the identity permutation p_0 as the two-row array

p'_1	p'_2	p'_3	$\cdots$	p'_n
1	2	3	$\cdots$	n

.

When constructing a transport plan from this two-row array, we notice that we will create a matrix T_{ij} that consists only of 0's and 1's. This simplifies the computation of the EMD to taking the sum of differences within the columns, i.e. $\sum_{i=1}^{n} |p'_i - i|$. Recall, that for any weakly increasing parking function p', we have $p'_i \leq i$. This allows us to simplify the last sum to $\sum_{i=1}^{n} (i - p'_i) = \mathrm{d}(p)$, which concludes the proof. □

Remark. The reader might recognize the construction of the transport plan T_{ij} as a special case of the *Robinson–Schenstedt–Knuth correspondence* (RSK) that was also used in [10, 28] and [27]. We refer to Fulton [32] for details on RSK.

However, in the case of parking functions, the value of the EMD can be computed in a more "direct" way, omitting the interpretation of parking functions as histograms: for a weakly increasing parking function p', note that

$$w(\varphi(p')) = p'. \tag{6.11}$$

Additionally, if $p \in \mathrm{PF}_n$ and p' its weakly increasing rearrangement, we note that (6.11) becomes

$$w(\varphi(p)) = p'.$$

The upshot is that the consecutive application of the maps w and φ is essentially sorting the parking function using the well–known algorithm *counting sort* [48, Section 5.2].

Alternatively, we can interpret the computation of the area statistic as an application of the EMD. To see this, we take $d = \mathrm{NENENNEENE}$ to be the Dyck path shown in Figure 20.

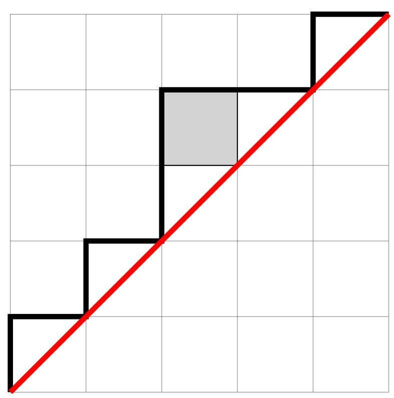


Figure 20: Dyck path $d = \mathrm{NENENNEENE}$ with highlighted area.

Clearly, we have $\mathrm{area}(d) = 1$. In the same way as before, we can construct a histogram from a Dyck path by viewing east steps as bins containing the north steps that are left of it in d. We take h to be the histogram corresponding to d so we have $h = (1, 1, 2, 0, 1)$. We can again obtain the area of d by computing

$$\mathrm{EMD}(h, 1^{(n)}) = 1.$$

After presenting the rich background and different areas of mathematics that are related to parking functions, we will provide further contributions to the topic in the remainder of this part.

7 COUNTING PARKING SEQUENCES AND PARKING ASSORTMENTS THROUGH PERMUTATIONS

In this chapter, we focus on two generalizations of parking functions: *parking sequences* and *parking assortments*. Both of these provide a more realistic aspect to the "parking" context by allowing cars to have different lengths. The lengths of cars $c_1, \ldots, c_n$ are stored in the *length vector* $y = (y_1, \ldots, y_n) \in \mathbb{N}_{>0}^n$, with car c_i requiring y_i consecutive empty spots to park.

Allowing cars to have different lengths opens up a further possibility for a potential collision: a car c_i can find an empty spot j, but it can only park if all of the spots $j, \ldots, j + y_i - 1$ are empty. That is, there must be a "gap large enough" to fit car c_i. The difference in parking sequences and parking assortments lies in the "parking rule" provided for the case that any of the spots $j + 1, \ldots, j + y_i - 1$ are occupied. Under the parking sequence rule, a car c_i has to leave the street if it finds an empty spot which is part of a gap too small for its length y_i, immediately ruling out the preference vector p as a parking sequence. If all cars are able to park, we call p a parking sequence, and denote the set of parking sequences for fixed $n \in \mathbb{N}_{>0}$ and $y \in \mathbb{N}_{>0}^n$ by $\mathrm{PS}_n(y)$.

Under the parking assortment rule, a car in this situation is allowed to continue searching along the street for a gap large enough in which to park. The pereference vector p is only ruled out if there is a car that cannot find a gap to park in on or after its preference. We call p a parking assortment if all cars are able to park; the set of parking assortments is denoted by $\mathrm{PA}_n(y)$.

The results of this chapter are part of a collaboration with Spencer J. Franks, Pamela E. Harris, Kimberly Harry and Megan Vance, published in *Enumerative Combinatorics and Applications* [30]. We provide enumerative counts for both parking sequences and parking assortments under fixed $n \in \mathbb{N}_{>0}$ and length vector $y \in \mathbb{N}_{>0}^n$. In our technical approach, we partition the sets of parking sequences and parking assortments into fibers of the outcome map $\mathcal{O}^{-1}(\pi)$ for all possible parking orders $\pi \in \mathrm{S}_n$. Thus, we will be able to count all preferences that yield a specific outcome, then sum over all possible outcomes to receive a enumerative formulas for $\#\mathrm{PS}_n(y)$ and $\#\mathrm{PA}_n(y)$.

7.1 Parking sequences

Parking sequences were first introduced by Ehrenborg and Happ [20] in 2016. The parking sequence rule describes the parking procedure as follows: n cars $c_1, \ldots, c_n$ with respective lengths $y = (y_1, \ldots, y_n)$ try to park on a one-way street with $m = \sum_{i=1}^{n} y_i$ spots. Each car c_i has a preferred spot p_i, which it attempts to park in first. If p_i is occupied, c_i continues driving down the street until it finds the first unoccupied spot s. If all spots $s + y_i - 1$ are unoccupied, c_i is able to park. However, if any of the spots $s+1, \ldots, s+y_i-1$ are occupied, c_i does not fit in the gap at s and has to leave the street. If all cars are able to park, we call the preference vector $p = (p_1, \ldots, p_n)$ a parking sequence for the length vector y. Figure 21 illustrates examples of preference lists which are parking sequences and preference lists which are not parking sequences for the length vector $y = (1, 2, 1)$.

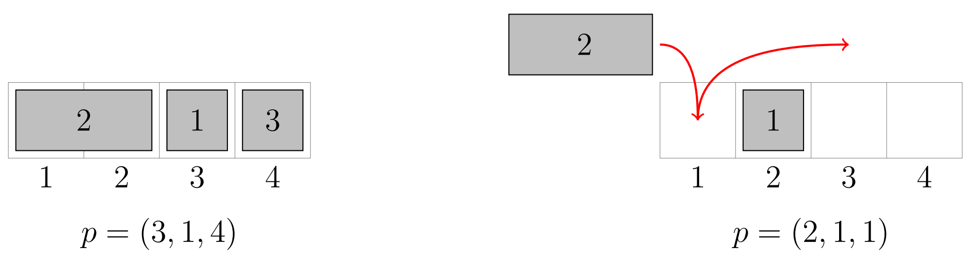


Figure 21: Note $p = (3, 1, 4)$ is a parking sequence for $y = (1, 2, 1)$ in which car 1 of length 1 parks in spot 3, car 2 of length 2 parks in spots 1 and 2, and car 3 of length 1 parks in spot 4. On the other hand, $p = (2, 1, 1)$ is not a parking sequence for y, since car 2 collides with car 1 when attempting to park.

Remark. If we choose as length vector $y = (1, 1, \ldots, 1)$, then

$$\#\mathrm{PS}_n(y) = (n+1)^{n-1},$$

as $\mathrm{PS}_n((1, 1, \ldots, 1)) = \mathrm{PF}_n$.

For a given length y, Ehrenborg and Happ established a first formula for $\#\mathrm{PS}_n(y)$:

$$\#\mathrm{PS}_n(y) = (y_1 + n)(y_1 + y_2 + n - 1) \cdots (y_1 + \cdots + y_{n-1} + 2). \tag{7.1}$$

The proof for (7.1) constructs a "circular street" on which cars park, an argument also used by Pollak (reported for example by Riordan [59]).

Given a parking sequence p for y, the result of the parking experiment yields a permutation $\pi = \pi_1\pi_2\cdots\pi_n$ of $[n]$, written in one-line notation, which denotes the order in which the cars park on the street. For example, the parking order in Figure 21 yields the permutation $\pi = 213$. Note that π corresponds to the order in which the cars park, not the order in which they arrive. Namely, for each $j \in [n]$, $\pi_j = i$ denotes that car i is the jth car parked on the street. In this work, we are interested in determining an alternative way of counting the number of parking sequences for y, by keeping track of those that park the cars in the order π. To this effect, we let S_n denote the set of permutations on $[n]$ and for a fixed y we define the outcome map $\mathcal{O}_{\mathrm{PS}_n(y)} : \mathrm{PS}_n(y) \to \mathrm{S}_n$ by $\mathcal{O}_{\mathrm{PS}_n(y)}(p) = \pi = \pi_1\pi_2\cdots\pi_n$; and given $\pi \in \mathrm{S}_n$, we study the fibers of the outcome map:

$$\mathcal{O}^{-1}_{\mathrm{PS}_n(y)}(\pi) = \left\{p \in \mathrm{PS}_n(y) : \mathcal{O}_{\mathrm{PS}_n(y)}(p) = \pi\right\}.$$

Using the outcome map, we will now show our first result, an enumerative formula for $\#\mathrm{PS}_n(y)$.

Theorem 7.1. *Fix $y = (y_1, y_2, \ldots, y_n) \in \mathbb{N}^n_{>0}$ and $\pi = \pi\pi\cdots\pi \in \mathrm{S}_n$. Then*

$$\#\mathcal{O}^{-1}_{\mathrm{PS}_n(y)}(\pi) = \prod_{i=1}^{n}\left(1 + \sum_{k\in\mathcal{L}(y,\pi_i)} y_k\right),$$

where $\mathcal{L}(y,\pi_i) = \emptyset$ if $i = 1$ or if $\pi_{i-1} > \pi_i$, otherwise $\mathcal{L}(y,\pi_i) = \{\pi_t, \pi_{t+1}, \ldots, \pi_{i-1}\}$ with $\pi_t\pi_{t+1}\ldots\pi_i$ being the longest subsequence of π such that $\pi_k < \pi_i$ for all $t \le k < i$.

Proof. Let $n \in \mathbb{N}_{>0}$, $y \in \mathbb{N}^n_{>0}$ and $p \in \mathrm{PS}_n(y)$, with $\pi = \pi_1\cdots\pi_n = \mathcal{O}_{\mathrm{PS}_n(y)}(p)$. Define as $\mathrm{Pref}_{\mathrm{PS}_n(y)}(\pi_i)$ the set of all values that $p_i \in p$ could have assumed without changing

$\mathcal{O}_{\mathrm{PS}_n(y)}(p)$. Then, from the independence of preferences, we get

$$\#\mathcal{O}^{-1}_{\mathrm{PS}_n(y)}(\pi) = \prod_{i=1}^{n} \#\mathrm{Pref}_{\mathrm{PS}_n(y)}(\pi_i).$$

Now, consider $\mathrm{Pref}_{\mathrm{PS}_n(y)}(\pi_i)$. The only spots car π_i can prefer are:

- the spot π_i occupies in π and
- any consecutive number of spots to the immediate left of π_i that is completely occupied by cars that arrived on the street before π_i.

The second point consists of exactly the elements of $\mathcal{L}(y, \pi_i)$; adding 1 to account for the potential of car π_i being lucky completes the result. □

We will illustrate this rather technical result with an example.

Example 7.2. Let $y = (1, 6, 5, 5, 3, 2, 2)$ and consider the parking order described by the permutation $\pi = 2457361$. We consider cars as they parked on the street from left to right in order to determine the preferences for each car so that the parking process results in the cars parking in the order π:

- Car 2 is parked first in the sequence of cars. Since there are no cars parked to the left of car 2, there is only 1 spot car 2 could have preferred, precisely where it is parked. Hence, $\mathrm{Pref}_{\mathrm{PS}_7(y)}(\pi_1) = \{1\}$.
- Car 4 is parked second in the sequence of cars. Since car 2 parked to the left of and earlier than car 4, car 4 could have preferred the spot it parked in or any of the spots occupied by car 2. Thus, $\mathrm{Pref}_{\mathrm{PS}_7(y)}(\pi_2) = \{1, 2, 3, 4, 5, 6, 7\}$.
- Car 5 is parked third in the sequence of cars. Since car 2 and car 4 parked to the left of and earlier than car 5, car 5 could have preferred the spot it parked in or any of the spots occupied by cars 2 or 4. Thus, $\mathrm{Pref}_{\mathrm{PS}_7(y)}(\pi_3) = \{1, 2, \ldots, 11, 12\}$.

- Car 7 is parked fourth in the sequence of cars. Since cars 2, 4, and 5 parked to the left of and earlier than car 7, car 7 could have preferred the spot it parked in or any of the spots occupied by cars 2, 4, or 5. Thus, $\mathrm{Pref}_{\mathrm{PS}_7(y)}(\pi_4) = \{1, 2, \ldots, 14, 15\}$.

- Car 3 is parked fifth in the sequence of cars. Since car 7 parked to the left of car 3 but entered the street after car 3, car 3 could not have preferred any spots to the left of where car 3 parked. Thus, $\mathrm{Pref}_{\mathrm{PS}_7(y)}(\pi_5) = \{17\}$.

- Car 6 is parked sixth in the sequence of cars. Since car 3 parked to the left of and earlier than car 6, car 6 could have preferred the spot it parked in or any of the spots occupied by car 3. Moreover, as the next car to the left of car 3 is car 7, which arrived after car 6, then car 6 could not have preferred any of the spots car 7 parks in or those before car 7. Thus, $\mathrm{Pref}_{\mathrm{PS}_7(y)}(\pi_6) = \{17, 18, \ldots, 22\}$.

- Car 1 is parked seventh in the sequence of cars. Since car 6 parked to the left of car 1 but entered the street after car 1, car 1 could not have preferred any spots to the left of where car 1 parked. Thus, $\mathrm{Pref}_{\mathrm{PS}_7(y)}(\pi_7) = \{24\}$.

These computations show that

$$\mathcal{O}^{-1}_{\mathrm{PS}_7(y)}(\pi) = \prod_{i=1}^{n} \mathrm{Pref}_{\mathrm{PS}_6(y)}(\pi_i)$$

and hence

$$\#\mathcal{O}^{-1}_{\mathrm{PS}_7(\boldsymbol{y})}(\sigma) = \prod_{i=1}^{7} \#\mathrm{Pref}_{\mathrm{PS}_7(y)}(\sigma_i) = 1 \cdot 7 \cdot 12 \cdot 15 \cdot 1 \cdot 6 \cdot 1 = 7560.$$

By using the result from Theorem 7.1 on the entire group S_n, we obtain a count for all parking sequences of a given y and n.

Corollary 7.3. *For a fixed $n \in \mathbb{N}_{>0}$ and $y \in \mathbb{N}^n_{>0}$, the number of parking sequences is*

$$\#\mathrm{PS}_n(y) = \sum_{\pi \in \mathrm{S}_n} \prod_{i=1}^{n} \left(1 + \sum_{k \in \mathcal{L}(y, \pi_i)} y_k \right).$$

7.2 Parking assortments

In this section, we provide a similar formula for parking assortments. We begin by recalling the parking assortment rule: n cars $c_1, \ldots, c_n$ of varying lengths $y_1, \ldots, y_n$ enter a one–way street from the left. Each car c_i has a preferred spot p_i, which attempts to park in first. If spot p_i is occupied or any of the spots $p_i + y_i - 1$ are occupied, i.e. the "gap" at p_i is not large enough for c_i to park, the car continues driving down the street until it either finds a sufficiently sized gap to park in or arrives at the end of the street, at which point it fails to park. If all cars are able to park, we call the preference vector p a parking assortment of n cars with lengths y. The set of all parking assortments for fixed n and y is called $\mathrm{PA}_n(y)$.

We illustrate the parking assortment rule, and especially the difference to parking sequences in Figure 22.

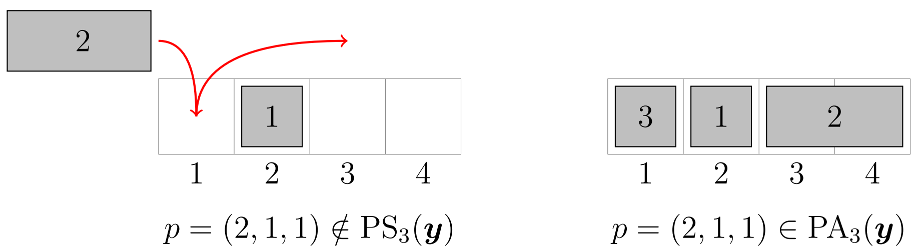


Figure 22: Let $y = (1, 2, 1)$. In Figure 21 we showed $(2, 1, 1) \notin \mathrm{PS}_3(y)$. However, under the parking assortment rule: car 1 parks in spot 2. Car 2 attempts to park in spot 1, unable to fit there, it continues down the street, parking in spot 3 (occupying spots 3 and 4). Car 3 finds spot 1 available; able to fit, it parks there. Thus, $p \in \mathrm{PA}_3(y)$.

Remark. For any $n \in \mathbb{N}_{>0}$ and $y \in \mathbb{N}_{>0}^n$ we observe that

$$\mathrm{PS}_n(y) \subseteq \mathrm{PA}_n(y).$$

When $y = (1, 1, \ldots, 1)$, we have $\mathrm{PA}_n(y) = \mathrm{PF}_n = \mathrm{PS}_n(y)$.

The following definitions set some necessary notation for our enumerative results.

Definition 7.4. For each $i \in [n]$ and $\pi = \pi_1\pi_2 \cdots \pi_n \in \mathrm{S}_n$, we let $\mathrm{Pref}_{\mathrm{PA}_n(y)}(\pi_i)$ be the set of possible preferences for car π_i so that it is the ith car to park on the street when using the parking assortment parking rule. We let $\#\mathrm{Pref}_{\mathrm{PA}_n(y)}(\pi_i)$ denote the cardinality of the set.

In the next technical definition, we fix a subword which ends at π_i, and further partition it into smaller subwords, so that in each smaller subword either all elements are smaller than or all are greater than π_i. The upshot is that for a car π_i, we need to find the rightmost contiguous block of cars that is parked to the left of π_i and arrived after π_i. This is, at the time car π_i enters the street, is a gap large enough for π_i to park in.

Definition 7.5. Let $\pi = \pi_1\pi_2\cdots\pi_n \in \mathrm{S}_n$. For $i \in [n]$ we define a partition $T(\pi_i)$ of the subword $\pi_1\pi_2\cdots\pi_{i-2}\pi_{i-1}$ into subwords as follows:

1. $T(\pi_1) = \emptyset$.

2. If $\pi_{i-1} > \pi_i$, then

$$T(\pi_i) = \begin{cases} \beta_\ell\alpha_\ell\cdots\beta_2\alpha_2\beta_1\alpha_1 & \text{if } \pi_1 < \pi_i \\ \alpha_{\ell+1}\beta_\ell\alpha_\ell\cdots\beta_2\alpha_2\beta_1\alpha_1 & \text{if } \pi_1 > \pi_i \end{cases} \tag{7.2}$$

   where

   - α_1 is the longest contiguous subword $\pi_s\pi_{s+1}\cdots\pi_{i-1}$ where $\pi_j > \pi_i$ for all $s \le j \le i-1$;
   - β_1 is the longest contiguous subword $\pi_t\pi_{t+1}\cdots\pi_{s-1}$ where $\pi_j < \pi_i$ for all $t \le j \le s-1$.

   We construct α_2 and β_2 in a similar fashion (from right to left) with the elements of $\pi_1\pi_2\cdots\pi_{t-1}$. Continue constructing subwords $\beta_k\alpha_k$, for $3 \le k \le \ell$, in this way until π_1 is included in one of them. If $\pi_1 > \pi_i$, then the leftmost subword is $\alpha_{\ell+1}$, otherwise the leftmost subword is β_ℓ.

3. If $\pi_{i-1} < \pi_i$, then

$$T(\pi_i) = \begin{cases} \alpha_\ell\beta_\ell\cdots\alpha_2\beta_2\alpha_1\beta_1 & \text{if } \pi_1 > \pi_i \\ \beta_{\ell+1}\alpha_\ell\beta_\ell\cdots\alpha_2\beta_2\alpha_1\beta_1 & \text{if } \pi_1 < \pi_i \end{cases} \tag{7.3}$$

   where

- β_1 is the longest contiguous subword $\pi_s\pi_{s+1}\cdots\pi_{i-1}$ where $\pi_j < \pi_i$ for all $s \le j \le i-1$;
- α_1 is the longest contiguous subword $\pi_t\pi_{t+1}\cdots\pi_{s-1}$ where $\pi_j > \pi_i$ for all $t \le j \le s-1$.

We construct α_2 and β_2 in a similar fashion (from right to left) with the elements of $\pi_1\pi_2\cdots\pi_{t-1}$. Continue constructing subwords $\alpha_k\beta_k$, for $3 \le k \le \ell$, in this way until π_1 is included in one of them. If $\pi_1 > \pi_i$, then the leftmost subword is $\beta_{\ell+1}$, otherwise the leftmost subword is α_ℓ.

We give some examples to illustrate the previous definition.

Example 7.6. If $\pi = 4123$, then

$$T(\pi_1) = \emptyset, \qquad T(\pi_2) = \underbrace{4}_{\alpha_1},$$

$$T(\pi_3) = \underbrace{4}_{\alpha_1}\underbrace{1}_{\beta_1} \text{ and } \qquad T(\pi_4) = \underbrace{4}_{\alpha_1}\underbrace{12}_{\beta_1}.$$

Example 7.7. If $\pi = 2457361$, then

$$T(\pi_1) = \emptyset, \qquad T(\pi_2) = \underbrace{2}_{\beta_1}, \qquad T(\pi_3) = \underbrace{24}_{\beta_1},$$

$$T(\pi_4) = \underbrace{245}_{\beta_1}, \qquad T(\pi_5) = \underbrace{2}_{\beta_1}\underbrace{457}_{\alpha_1}, \qquad T(\pi_6) = \underbrace{245}_{\beta_2}\underbrace{7}_{\alpha_1} \text{ and}$$

$$T(\pi_7) = \underbrace{245736}_{\alpha_1}.$$

Theorem 7.8. *Let $\pi = \pi_1\pi_2\cdots\pi_n \in \mathrm{S}_n$ and $y = (y_1, y_2, \ldots, y_n) \in \mathbb{N}^n_{>0}$. Fix $i \in [n]$ and partition $T(\pi_i)$ as in Definition 7.5. Then $\mathrm{Pref}_{\mathrm{PA}_n(y)}(\pi_i)$ has the following cardinalities:*

1. *if $i = 1$ or $\pi_{i-1} > \pi_i$, then $\#\mathrm{Pref}_{\mathrm{PA}_n(y)}(\pi_i) = 1$;*
2. *if $T(\pi_i) = \beta_1$, then $\#\mathrm{Pref}_{\mathrm{PA}_n(y)}(\pi_i) = 1 + \sum_{\pi_k \in \beta_1} y_{\pi_k}$;*

3. *otherwise*

$$\#\mathrm{Pref}_{\mathrm{PA}_n(y)}(\pi_i) = \begin{cases} 1 + \displaystyle\sum_{k=1}^{i-1} y_{\pi_k} & \textit{if } m(i) \textit{ does not exist} \\ \displaystyle\sum_{\pi_k \in \beta_{m(i)}\alpha_{m(i)-1}\beta_{m(i)-1}\cdots\alpha_1\beta_1\pi_i} y_{\pi_k} & \textit{if } m(i) \textit{ exists} \end{cases}$$

where

$$m(i) = \min\left\{1 \leq j \leq \ell : \sum_{\pi_k \in \alpha_j} y_{\pi_k} \geq y_{\pi_i}\right\}.$$

which denotes the closest gap to the left of π_i in which π_i could have parked.

Proof. We proceed by proving each case independently.

Case 1: If $i = 1$, then $\pi_i = \pi_1$ is the first car parked on the street, which implies that it must have preferred the first parking spot on the street. Hence $\#\mathrm{Pref}_{\mathrm{PA}_n(y)}(\pi_1) = 1$, as claimed. If $\pi_{i-1} > \pi_i$, this means that the car parked immediately to the left of π_i arrived after π_i. Hence car π_i can only prefer the spot it parked in, as otherwise it would have parked elsewhere. This implies $\#\mathrm{Pref}_{\mathrm{PA}_n(y)}(\pi_i) = 1$, as claimed.

Case 2: If $T(\pi_i) = \beta_1$, then $\pi_j < \pi_i$ for all $j \in [i-1]$. Thus all of the cars parked left of π_i arrived and parked before π_i. Hence π_i could prefer all of the spots cars $\pi_1, \pi_2, \ldots, \pi_{i-1}$ occupy, as well as the spot in which π_i ultimately parks. This implies $\#\mathrm{Pref}_{\mathrm{PA}_n(y)}(\pi_i) = 1 + \sum_{\pi_j \in \beta_1} y_{\pi_j}$.

Case 3: Note that $\pi_{i-1} < \pi_i$ (as otherwise this would be Case 1). Furthermore, we can assume that α_1 exists (as otherwise this would be Case 2). Hence, by Definition 7.5, we have

$$T(\pi_i) = \pi_1\pi_2\cdots\pi_{i-2}\pi_{i-1} = \begin{cases} \alpha_\ell\beta_\ell\cdots\alpha_2\beta_2\alpha_1\beta_1 & \text{if } \pi_1 > \pi_i \\ \beta_{\ell+1}\alpha_\ell\beta_\ell\cdots\alpha_2\beta_2\alpha_1\beta_1 & \text{if } \pi_1 < \pi_i \end{cases}$$

where β_1 is the longest contiguous subword consisting of $\pi_j < \pi_i$ and α_1 is the longest contiguous subword consisting of $\pi_j > \pi_i$. In either case, we note that by definition, each α_j denotes a set of cars parking contiguously on the street, arriving in the queue after car π_i and parking to the left of car π_i. The cars in the subwords α_j (for $1 \leq j \leq \ell$) create gaps in the street which π_i could potentially park in if they happen to be large enough.

That is, for any $j \in [\ell]$, if $\sum_{\pi_k \in \alpha_j} y_{\pi_k} \geq y_{\pi_i}$, then π_i

- cannot prefer all of the spots occupied by the cars in α_j and
- cannot prefer any spots to the left of the spots occupied by the cars in α_j,

since then π_i would park either before or within the spots occupied by the cars in α_j. Both cases contradict the fact that π_i is the ith car parked on the street.

In fact, the only parking spots car π_i could prefer are

- the spots occupied by the cars in $\beta_{m(i)}\alpha_{m(i)-1}\beta_{m(i)-1}\cdots\alpha_1\beta_1$,
- the right-most $y_{\pi_i} - 1$ spots occupied by the cars in $\alpha_{m(i)}$, or
- the spot π_i parks in.

Note that this exhausts all of the possible preferences for π_i, as $\alpha_{m(i)}$ (by definition) is the closest gap in which π_i could park. Thus, the number of spots that car π_i can prefer is

$$\begin{aligned}\#\mathrm{Pref}_{\mathrm{PA}_n(\boldsymbol{y})}(\pi_i) &= 1 + (y_{\pi_i} - 1) + \sum_{\pi_k \in \beta_{m(i)}\alpha_{m(i)-1}\beta_{m(i)-1}\cdots\alpha_1\beta_1} y_{\pi_k} \\ &= \sum_{\pi_k \in \beta_{m(i)}\alpha_{m(i)-1}\beta_{m(i)-1}\cdots\alpha_1\beta_1\pi_i} y_{\pi_k}\end{aligned}$$

as claimed.

□

We can now formally state and prove the analogous result to Theorem 7.1 for parking assortments.

Theorem 7.9. *Fix $y = (y_1, y_2, \ldots, y_n) \in \mathbb{N}^n_{>0}$ and let $\pi = \pi_1\pi_2\cdots\pi_n \in \mathrm{S}_n$. Then*

$$\#\mathcal{O}^{-1}_{\mathrm{PA}_n(y)}(\pi) = \prod_{i=1}^{n} \#\mathrm{Pref}_{\mathrm{PA}_n(y)}(\pi_i),$$

where

$$\#\mathrm{Pref}_{\mathrm{PA}_n(y)}(\pi_i) = \begin{cases} 1 & \text{if } i = 1 \text{ or } \pi_{i-1} > \pi_i \\ 1 + \sum_{\pi_k \in \beta_1} y_{\pi_k} & \text{if } T(\pi_i) = \beta_1 \\ 1 + \sum_{k=1}^{i-1} y_{\pi_k} & \text{if } m(i) \text{ does not exist} \\ \sum_{\pi_k \in \beta_{m(i)}\alpha_{m(i)-1}\beta_{m(i)-1}\cdots\alpha_1\beta_1\pi_i} y_{\pi_k} & \text{if } m(i) \text{ exists} \end{cases} \tag{7.4}$$

with

$$m(i) = \min\left\{1 \le j \le \ell : \sum_{\pi_k \in \alpha_j} y_{\pi_k} \ge y_{\pi_i}\right\}.$$

again denoting the closest spot to the left of π_i in which π_i could have parked.

Proof. The theorem follows directly from independence of preferences and Theorem 7.8. □

Theorem 7.9 immediately implies the following result.

Corollary 7.10. *Fix $y = (y_1, y_2, \ldots, y_n) \in \mathbb{N}^n_{>0}$ and for any $\pi \in \mathrm{S}_n$, let $|\mathcal{O}^{-1}_{\mathrm{PA}_n(y)}(\pi)|$ be as given by Theorem 7.9. Then*

$$\#\mathrm{PA}_n(y) = \sum_{\pi \in \mathrm{S}_n} \#\mathcal{O}^{-1}_{\mathrm{PA}_n(y)}(\pi).$$

We set $n = 4$ and $y = (1, 2, 1, 2)$ to provide example values for $\#\mathcal{O}^{-1}_{\mathrm{PS}_n(y)}$ (Table 1) and $\#\mathcal{O}^{-1}_{\mathrm{PA}_n(y)}$ (Table 2) for all $\pi \in \mathrm{S}_4$.

π	$\#\mathcal{O}^{-1}_{\mathrm{PS}_4(y)}(\pi)$	π	$\#\mathcal{O}^{-1}_{\mathrm{PS}_4(y)}(\pi)$	π	$\#\mathcal{O}^{-1}_{\mathrm{PS}_4(y)}(\pi)$	π	$\#\mathcal{O}^{-1}_{\mathrm{PS}_4(y)}(\pi)$
1234	40	2134	20	3124	10	4123	8
1243	8	2143	4	3142	3	4132	2
1324	10	2314	15	3214	5	4213	4
1342	6	2341	12	3241	4	4231	3
1423	6	2413	6	3412	4	4312	2
1432	2	2431	3	3421	2	4321	1

Table 1: Cardinalities of the sets $\mathcal{O}^{-1}_{\mathrm{PS}_4((1,2,1,2))}(\pi)$ for each $\pi \in \mathrm{S}_4$.

π	$\#\mathcal{O}^{-1}_{\mathrm{PA}_4(y)}(\pi)$	π	$\#\mathcal{O}^{-1}_{\mathrm{PA}_4(y)}(\pi)$	π	$\#\mathcal{O}^{-1}_{\mathrm{PA}_4(y)}(\pi)$	π	$\#\mathcal{O}^{-1}_{\mathrm{PA}_4(y)}(\pi)$
1234	40	2134	20	3124	15	4123	12
1243	8	2143	4	3142	3	4132	2
1324	10	2314	15	3214	5	4213	4
1342	6	2341	12	3241	4	4231	3
1423	6	2413	6	3412	6	4312	3
1432	2	2431	3	3421	2	4321	1

Table 2: Cardinalities of the sets $\mathcal{O}^{-1}_{\mathrm{PA}_4((1,2,1,2))}(\pi)$ for each $\pi \in \mathrm{S}_4$.

8 LUCKY CARS AND THE QUICKSORT ALGORITHM

In this chapter, we reveal a connection between certain parking objects and the *Quicksort* algorithm. We show that the number of parking preferences with n cars of which $n-1$ are lucky is equal to the number of comparisons the Quicksort algorithm performs when sorting all $n!$ elements of S_n.

This chapter is based on a collaboration with Pamela E. Harris and J. Carlos Martínez Mori [41].

8.1 The Quicksort algorithm

A common student experience is to form a line where people are sorted by their height, say with the shortest student on the left and the tallest student on the right. One approach to sort the students is to have students join the line one at a time, each finding their place in line by *comparing* their height to that of those already in it. Thereby one at a time, students self-sort and the end result yields the desired student ordering.

As inconspicuous as this activity may seem, sorting algorithms and their use of computational resources (e.g., comparisons) are subject to vast amounts of scientific attention. One such algorithm, aptly called `Quicksort`, was developed by Tony Hoare—who would later become Sir Tony Hoare—in 1961 while he was a British Council exchange student visiting Moscow State University [29, 42, 45]. `Quicksort` is a classical sorting algorithm and a prime example of the *divide-and-conquer* paradigm. By now, its design and analysis is covered in most undergraduate textbooks on algorithm design (e.g., [17, 46, 48, 64], to list a few).

The algorithm first bipartitions the array elements by selecting a *pivot* element: elements less than or equal to the pivot element are assigned to a “left” array, whereas elements greater than the pivot element are assigned to a “right” array. In this way, given an array with $n \in \mathbb{N} := \{1, 2, 3, \ldots\}$ elements, the algorithm first makes $n-1$ pivot comparisons. Then,

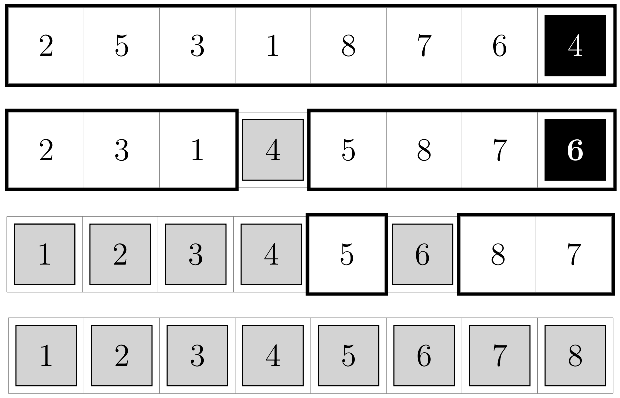


Figure 23: Using Quicksort to sort the permutation 25318764. The rightmost subarray element is chosen as the pivot, highlighted in black with white numerals four and six. The algorithm partitions the array into elements larger and smaller than the pivot. If the subarray is length 3 or smaller, it sorts by brute force. Otherwise, it recurses on the resulting subarrays, marked with thick black lines. Correctly ordered elements are highlighted in gray.

the algorithm recurses *separately* on the left and right arrays to obtain their respective sorted versions[1].

In Figure 23, we illustrate an execution of the algorithm as described in [46, Chapter 13.5]. (A more detailed description, with precise array-indexing and swap operations, can be found in [17, Chapter 7]).

Given an array of size n, the *overall* number of pivot comparisons made by `Quicksort` (including all recursive calls) crucially depends on the choice of pivot elements. For example, if the algorithm happens to repeatedly select the smallest element as the pivot element, then the input array of any particular recursive call is only one unit smaller than that of the preceding call. It follows that, in the worst case, the algorithm makes $\sum_{i=1}^{n}(n-i) = \frac{n(n-1)}{2} = O(n^2)$ pivot comparisons[2]. However, if we assume the original array is ordered uniformly at random (i.e., it is in any given ordering with probability $1/n!$), the expected number of pivot comparisons is $O(n \log n)$. This bound can be obtained as follows: Let Q_n be the expected number of pivot comparisons performed by `Quicksort` given an array of

[1]In practice, `Quicksort` is implemented in conjunction with non-recursive sorting algorithms that work well on small arrays. This and other practical optimizations of the algorithm are described by Sedgewick in [63].

[2]Given $f, g : \mathbb{N} \to \mathbb{N}$, we say $f = O(g)$ if there exist $k \in \mathbb{N}$ and $C \in \mathbb{R}_{>0}$ such that $f(n) \leq C \cdot g(n)$ for all $n \geq k$.

size n ordered uniformly at random. Then, regardless of the pivot selection strategy (e.g., selecting the right-most element), *any* choice of pivot element is equally likely to be the kth smallest and we have

$$Q_n = (n-1) + \frac{1}{n}\left(\sum_{k=1}^{n} Q_{k-1} + Q_{n-k}\right). \tag{8.1}$$

This recursive relation unravels (see [13, Chapter 4.7] for details) to obtain

$$Q_n = 2(n+1)H_n - 4n = O(n \log n), \tag{8.2}$$

where H_n is the nth harmonic number, i.e. $H_n \coloneqq \sum_{i=1}^{n} \frac{1}{i}$.

Note that while Q_n is not an integer sequence, $A_n = n!Q_n$ is an integer sequence and denotes the *total* number of comparisons performed by `Quicksort` given *all* possible orderings of an array of size n, (see [43, A288964]).

In fact, by the definition of expectation, A_n is given by

$$A_n = n!Q_n = n!\left[2(n+1)H_n - 4n\right]. \tag{8.3}$$

8.2 Relation to parking objects

To state our main result, we first recall the *lucky* statistic on parking functions: as stated in Definition 6.7, for $p \in \mathrm{PF}_n$ we have

$$\mathrm{L}(p) = \#\left\{\pi_i \in \mathcal{O}(p) \mid i = p_{\pi_i}\right\}, \tag{8.4}$$

i.e., $\mathrm{L}(p)$ is the number of cars that park on their preferred spot. For this section, we will slightly extend this definition to include arbitrary n-tuples of $[n]$. We still call a car "lucky" if it parks on its preferred spot – cars that are unable to park are therefore not considered lucky.

Definition 8.1. Let $n \in \mathbb{N}_{>0}$. Then, for $\alpha \in [n]^n$, we define

$$\mathrm{L}'(\alpha) = \#\left\{i \in \alpha \mid i \text{ is lucky}\right\}.$$

The lucky statistic was studied by Gessel and Seo [34], who gave the following generating function

$$\sum_{p\in \mathrm{PF}_n} q^{\mathrm{L}(p)} = q\prod_{i=1}^{n}(i+(n-i+1)q). \tag{8.5}$$

In what follows, we let

$$\mathcal{L}_n = \{\alpha \in [n]^n : \mathrm{L}'(\alpha) = n-1\}$$

and give a formula for $L_n = \#\mathcal{L}_n$; the number of preference lists in $[n]^n$ with $n-1$ lucky cars. Note that unlike (8.5), where the sum is over parking functions of length n, $\mathcal{L}_n$ is not restricted to elements of PF_n.

Our main result shows that $L_n = A_n$, where A_n is given by (8.3) and counts the total number of comparisons performed by `Quicksort` given all possible orderings of an array of size n.

Theorem 8.2. *If $n \geq 2$, then $L_n = A_n = n!\,Q_n$.*

Proof. It suffices to show that both A_n and L_n satisfy the second order recurrence relation

$$f_n = 2nf_{n-1} - n(n-1)f_{n-2} + 2(n-1)! \quad \text{with } f_0 = f_1 = 0. \tag{8.6}$$

Algebraic manipulations involving telescoping sums show that (8.1) implies $nQ_n = (n+1)Q_{n-1}+2(n-1)$ with $Q_0 = 0$. As $A_n = n!Q_n$ we have that $A_n = (n+1)A_{n-1}+2(n-1)(n-1)!$ with $A_0 = 0$. Then,

$$\begin{aligned} A_n &= 2nA_{n-1} - (n-1)A_{n-1} + 2(n-1)(n-1)! \\ &= 2nA_{n-1} - (n-1)(nA_{n-2} + 2(n-2)(n-2)!) + 2(n-1)(n-1)! \\ &= 2nA_{n-1} - n(n-1)A_{n-2} + 2(n-1)! \end{aligned}$$

with $A_0 = A_1 = 0$, as desired.

We now show that L_n, which is the number of preference lists in $[n]^n$ with exactly $n-1$ lucky cars, also satisfies (8.6). Note that each $\alpha \in \mathcal{L}_n$ contains exactly one duplicate entry;

all other entries are pairwise distinct. Hence, there is exactly one pair of cars that share a preference; call this pair of cars the *competing cars*. This also implies that there is exactly one parking spot that is not preferred by any car; call this parking spot the *undesired spot*. Note that

$$L_n = \#N_n + \#M_n, \tag{8.7}$$

where

$$N_n = \{\alpha \in \mathcal{L}_n : \text{car 1 is not a competing car}\}, \text{ and}$$
$$M_n = \{\alpha \in \mathcal{L}_n : \text{car 1 is a competing car}\}.$$

We begin by enumerating the elements of N_n. To do so we use the following fact. If car 1 is not a competing car and prefers spot a_1, then the remaining $n-1$ cars behave as if car 1 preferred spot n. Namely, by shifting the preferences so that $a_i = a_i - \mathbf{1}_{\{a_i > a_1\}}$ for each $i \in [n] \setminus \{1\}$, where $\mathbf{1}$ denotes the indicator function, we note there are L_{n-1} preference lists that satisfy the required condition. As there are n options for the preference a_1 of car 1 we have that

$$\#N_n = nL_{n-1}. \tag{8.8}$$

Now, if $\alpha = (a_1, a_2, \ldots, a_n) \in M_n$, then car 1 is a competing car. This implies that there exists an index $2 \leq j \leq n$ such that $a_1 = a_j$, while all other entries are pairwise distinct as well as distinct from this value. We now consider the cases $j = 2$ and $3 \leq j \leq n$ separately:

- If $j = 2$, then $a_1 = a_2$. If $a_1 = a_2 = n$, then cars $3, 4, \ldots, n$ have pairwise distinct preferences among the first $n-1$ parking spots. Hence, if $a_1 = a_2 = n$, there are $(n-1)!$ possibilities for the remaining entries $a_3, a_4, \ldots, a_n$ in α. If $a_1 = a_2 = k < n$, then spot $k+1$ is the undesired spot and cars $3, 4, \ldots, n$ have pairwise distinct preferences among the $n-1$ parking spots $\{1, 2, \ldots, k-1, k+2, \ldots, n\}$. Hence, if $a_1 = a_2 = k < n$, there are $(n-1)!$ possibilities for the remaining entries $a_3, a_4, \ldots, a_n$ in α. Together, these mutually exclusive cases contribute $2(n-1)!$ toward the total count.

- If $3 \leq j \leq n$, then car 2 is not a competing car. Note that there are n possible preferred spots for car 2. Moreover, the preference list $(a_1, a_3, a_4, \ldots, a_n)$ contains the two competing cars (with the first car being in the competing pair), and all of the remaining $n-1$ cars have preferences in the set $[n] \setminus \{a_2\}$. By shifting the preferences so that $a_i = a_i - \mathbf{1}_{\{a_i > a_2\}}$ for each $i \in [n] \setminus \{2\}$, we note that the total number of such preferences for cars $1, 3, 4, \ldots, n$ is given by $\#M_{n-1}$. This case contributes $n \cdot \#M_{n-1}$ toward the total count.

Therefore,

$$\#M_n = n \cdot \#M_{n-1} + 2(n-1)!. \tag{8.9}$$

Substituting (8.8) and (8.9) into (8.7) yields

$$L_n = nL_{n-1} + n \cdot \#M_{n-1} + 2(n-1)!. \tag{8.10}$$

Subtracting $2nL_{n-1}$ from both sides of (8.10) yields

$$\begin{aligned} L_n - 2nL_{n-1} &= -nL_{n-1} + n \cdot \#M_{n-1} + 2(n-1)! \\ &= -n((n-1)L_{n-2} + \#M_{n-1}) + n \cdot \#M_{n-1} + 2(n-1)! \\ &= -n(n-1)L_{n-2} + 2(n-1)! \end{aligned}$$

which upon rearranging is the desired recurrence relation

$$L_n = 2nL_{n-1} - n(n-1)L_{n-2} + 2(n-1)!$$

with $L_0 = L_1 = 0$. □

8.3 Summing over parking orders

Similarly to the results shown in Chapter 7, we can employ the technique of summing over all potentially resulting parking orders, which we also refer to as “counting through permutations”, to our newly restricted parking objects, as well as to parking functions with an arbitrary set of lucky cars. We use this section to provide formulas for these techniques.

8.3.1 All cars but one are lucky

We begin with an alternative counting formula for the parking objects examined previously in Section 8.2. For preference lists $\alpha \in [n]^n$ such that $n-1$ cars are lucky and only one car is unlucky (and might not park at all), counting through permutations reduces to:

Theorem 8.3. *For all $n \in \mathbb{N}_{>0}$,*

$$\#\mathcal{L}_n = \sum_{\pi \in \mathrm{S}_n} \sum_{i=1}^{n} \left(S_i(\pi, i-1) + S_i(\pi, n) \right),$$

where $S_i(\pi, j)$ is the longest contiguous sequence $\pi_{j-\#S_i(\pi,j)} \cdots \pi_{j-1}\pi_j$ such that $\pi_k < \pi_i$ for all $\pi_k \in S_i(\pi, j)$.

Proof. Every lucky car can only have 1 preference. Therefore, each car only contributes to $\#\mathcal{L}_n$ in the case that it is the *unlucky* car. The unlucky car has 2 options:

1. It parks. Then it must have preferred a spot from a number of contiguously occupied spots to the left of its parking spot. The maximum possible for any permutation π is captured in $S_i(\pi, i-1)$.

2. It does not park. Then, it must have preferred a spot from a number of contiguously occupied spots to the very end of the street. The number of spots the unlucky car can choose from is $S_i(\pi, n)$.

Adding the two options yields the number of preferences each car can have. Since all other cars are lucky, we cannot multiply the numbers of preferences here. Instead summing over each car and each permutation provides the result. □

8.3.2 Arbitrary cars are lucky

To conclude this chapter, we will provide an enumerative formula for the number of parking functions with a fixed set $\mathfrak{L}$ of lucky cars. For a given set $\mathfrak{L} \subseteq [n]$, we define as $\mathrm{PF}_n(\mathfrak{L})$ the set of parking functions with lucky cars $\mathfrak{L}$.

We begin by stating a necessary definition.

Definition 8.4. Let $\pi = \pi_1 \cdots \pi_n \in \mathrm{S}_n$ be a parking order and $\mathfrak{L} \subseteq [n]$. Then, for each car π_i, the number of spots π_i could have preferred such that the lucky cars are exactly $\mathfrak{L}$ is:

$$\mathrm{Pref}_{\mathfrak{L}}(\pi, \pi_i) = \begin{cases} 0 & \text{if } \pi_i = \pi_1 \text{ or } \pi_i = 1 \\ \# S_i(\pi, i-1) & \text{else,} \end{cases}$$

where $S_i(\pi, j)$ is the longest contiguous sequence $\pi_{j-\#S_i(\pi,j)} \cdots \pi_{j-1}\pi_j$ such that $\pi_k < \pi_i$ for all $\pi_k \in S_i(\pi, j)$.

Theorem 8.5. *Let $n \in \mathbb{N}_{>0}$ and $\mathfrak{L} \subseteq [n]$. Then, the number of parking functions with lucky cars $\mathfrak{L}$ is found by*

$$\#\mathrm{PF}_n(\mathfrak{L}) = \delta_{\mathfrak{L},[n]} \cdot n! + \sum_{\pi \in \mathrm{S}_n} \prod_{\pi_i \in [n] - \mathfrak{L}} \mathrm{Pref}_{\mathfrak{L}}(\pi, \pi_i),$$

with $\delta_{x,y}$ the Kronecker delta.

Proof. We follow the standard approach to find the number of preference lists that result in each possible outcome $\pi \in \mathrm{S}_n$. Since lucky cars can only have a single preference, they do not contribute to the number of preference lists for each outcome. Now, we are only counting the preferences of the unlucky cars $[n] - \mathfrak{L}$, which can prefer as many spots as there are cars parked contiguously to the left of their parking spot. Since we are summing over all permutations, we have to handle the special case that $\pi_1 \notin \mathfrak{L}$ – Definition 8.4 rectifies and results in 0 potential preferences. Additionally, we need to consider the case that $\mathfrak{L} = [n]$ – the resulting parking functions are exactly the permutations, which we are conditionally adding through the Kronecker delta. □

Example 8.6. Consider $n = 4$ and $\mathfrak{L} = \{1, 3\}$ – that is, we have 4 cars $c_1, \ldots, c_4$ and cars c_3 and c_3 are the only lucky cars, i.e. the only cars parking in their preferred spots. Since $\mathfrak{L} \neq [n]$, the Kronecker delta is 0 and we are left to sum over parking outcomes $\pi \in \mathrm{S}_n$. For each parking order, we need only consider the unlucky cars $\{2, 4\} = [n] - \mathfrak{L}$, since a lucky car can only have 1 preferred spot. We list some parking orders $\pi \in \mathrm{S}_4$, showing how many possible preferences lead to the outcome π.

- $\pi = 1234$:
  - For car 2, we have $\mathrm{Pref}_{\mathfrak{L}}((1234), 2) = \#S_2((1234), 1) = 1$.
  - For car 4, we have $\mathrm{Pref}_{\mathfrak{L}}((1234), 4) = \#S_4((1234), 4) = 3$.
- $\pi = 3214$
  - For car 2, we have $\mathrm{Pref}_{\mathfrak{L}}((4132), 2) = \#S((4132), 2) = 0$, making the entire product over potential preferences for each car 0. We notice that car 2 is parked to the right of a car that arrived after car 2 – therefore, it can only prefer the spot it parks on, which implies that car 2 is lucky. However, since $2 \notin \mathfrak{L}$, this parking order is impossible to achieve, and therefore there are 0 preference lists yielding this outcome under $\mathfrak{L} = \{1, 3\}$.
- $\pi = 2134$
  - For car 2, we have $\mathrm{Pref}_{\mathfrak{L}}((4132), 2) = 0$, because we see that $2 = \pi_1$. The car parking first on the street must always be lucky, therefore $\pi = 2134$ is also an impossible outcome under $\mathfrak{L} = \{1, 3\}$.

Next, consider $\mathfrak{L} = \{2, 4\}$ and any outcome $\pi \in \mathrm{S}_4$. Since $1 \notin \mathfrak{L}$, for each $\pi_i = 1$, car c_1 will contribute to the result. However, since car c_1 always parks on an entirely empty street, it is always a lucky car. Therefore, we receive $\mathrm{Pref}_{\mathfrak{L}}(\pi, 1) = 0$, as stated in Definition 8.4.

In the next chapter, we use the intersection of a subset of parking functions and *Fubini rankings* to examine *Boolean algebras* in the weak order of S_n.

9 BOOLEAN INTERVALS IN THE WEAK ORDER OF S_n

In the final chapter of this work, we provide an enumeration and characterization of *Boolean intervals* in the weak order of S_n through a connection to *unit fubini rankings* – which are a subset of parking functions. This chapter is based on joint work with Jennifer Elder, Pamela E. Harris and J. Carlos Martínez Mori. With slight changes to notation, the remainder of the chapter is available as a preprint [21].

9.1 Background

A poset is called *Boolean* if it is isomorphic to the poset of subsets of a set I ordered by inclusion. The term *Boolean poset* is inherited from *Boolean algebras*, given that one of the most familiar examples of a Boolean algebra is the power set 2^I. If $\#I = k < \infty$, then a Boolean poset is a distributive lattice, making it a ranked poset. Hence, we let B_k denote a Boolean poset of rank k.

Boolean posets appear frequently in combinatorics, especially as intervals (subposets) within larger structures. In these cases, they are referred to as *Boolean intervals*. One notable example is that of Boolean intervals in the *right weak (Bruhat) order lattice* on the symmetric group S_n [7, 71, 72, 73], where $n \in \mathbb{N}_{>0}$. The weak order lattice on S_n, denoted $W(\mathrm{S}_n)$, is constructed by the simple transpositions $s_i = (i, i+1)$ for $i \in [n-1]$, where cover relations arise from the (right hand side) application of a single simple transposition. Therefore, simple transpositions are also referred to as generators. Figure 24 highlights a B_3 interval in $W(\mathrm{S}_6)$.

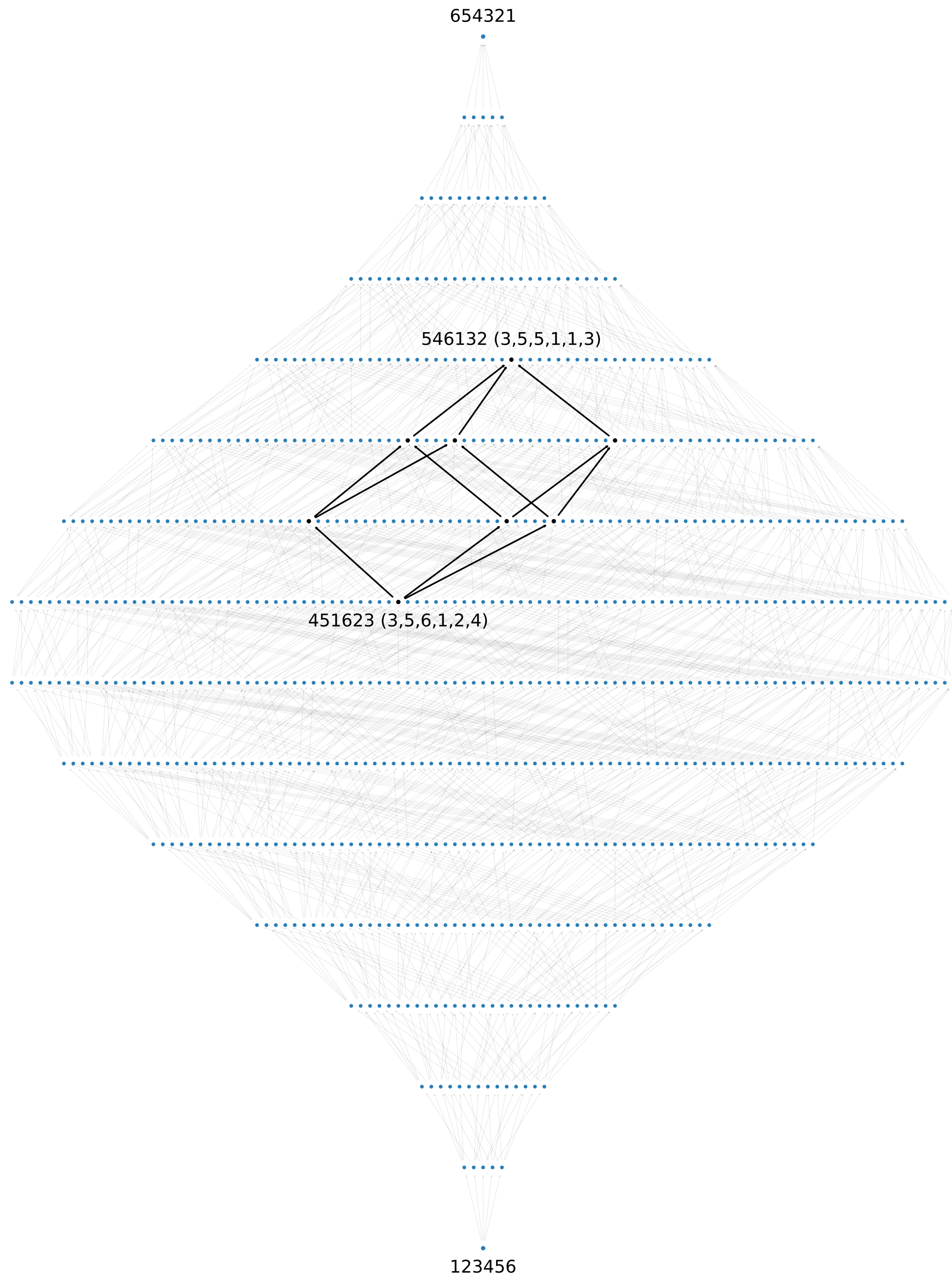


Figure 24: Illustration of $W(\mathrm{S}_6)$. A Boolean interval B_3 with minimal element 451623 and maximal element 546132, written in one-line notation, is highlighted. The decorators $(3,5,6,1,2,4)$ and $(3,5,5,1,1,3)$ indicate the unit Fubini rankings associated with the minimal and maximal Boolean subintervals of B_3 rooted at 451623, respectively.

Tenner established that Boolean posets appear as intervals $[v,w]$ in the weak order if and only if $v^{-1}w$ is a permutation composed of only commuting generators [73, Corollary 4.4]. We recall that generators s_i and s_j commute whenever $|i-j|>1$. We provide more background on the weak order lattice and Boolean intervals in Section 9.1.1. Tenner also established that Boolean intervals with a generator as minimal element are enumerated by products of at most two Fibonacci numbers [73, Proposition 5.9].

9.1.1 Boolean intervals in the weak order lattice

Boolean posets are constructed by subsets of a set I ordered by inclusion. Figure 25 illustrates some small examples.

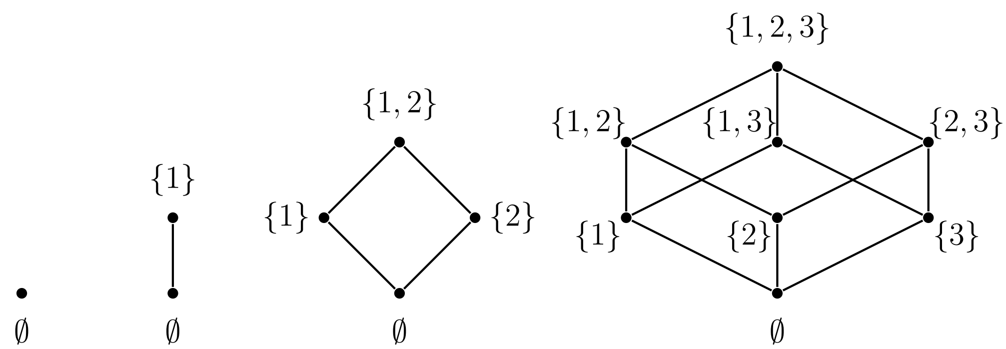


Figure 25: Boolean posets B_0, B_1, B_2, and B_3 of rank 0, 1, 2, and 3, respectively.

We recall the definition of the *descent* and *ascent* set, which plays a key role in the proof of Theorem 9.25.

Definition 9.1. For a permutation $\sigma = \sigma_1\sigma_2 \cdots \sigma_n \in \mathrm{S}_n$, the *ascent set* of σ is given by

$$\mathrm{Asc}(\sigma) = \{i \in [n-1] \mid \sigma \text{ has an ascent at } i\}.$$

Similarly, the *descent set* of σ is given by

$$\mathrm{Des}(\sigma) = \{i \in [n-1] \mid \sigma \text{ has a descent at } i\}$$

The *right weak (Bruhat) order*, denoted $W(\mathrm{S}_n)$, is a partial order on S_n. Its cover relations are defined by the application of a single simple (adjacent) transposition on the right hand side. That is, $\tau \lessdot \sigma$ if and only if $\tau s_i = \sigma$ for some $i \in \mathrm{Des}(\sigma)$. In general, if $\tau \leq \sigma$, then there exists a collection $s_{i_1}, \ldots, s_{i_k}$ of simple transpositions such that $\tau s_{i_1} \ldots s_{i_k} = \sigma$.

Note that $W(\mathrm{S}_n)$ is a bounded lattice for all $n \geq 2$ [68]. In one-line notation, its minimal element is $12\cdots n$ while its maximal element is $n(n-1)\cdots 21$. Figure 26 illustrates $W(\mathrm{S}_4)$ with its elements written in one-line notation.

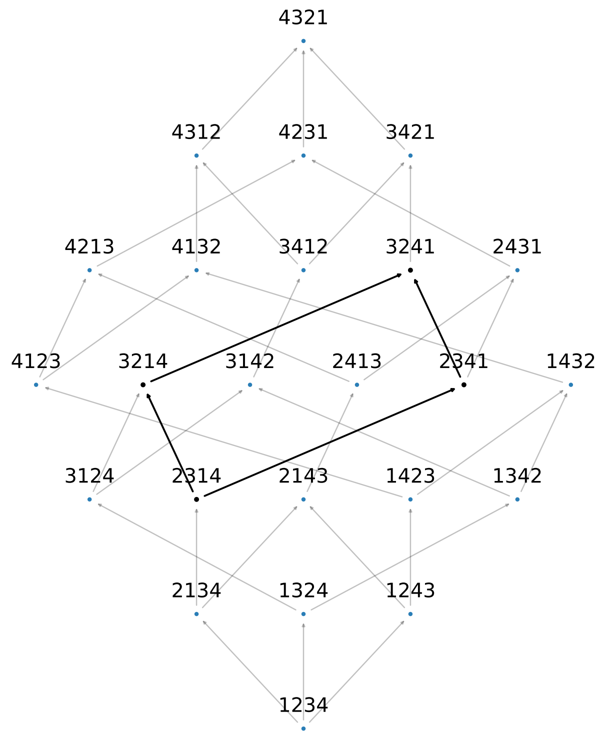


Figure 26: Illustration of $W(\mathrm{S}_4)$ with a highlighted Boolean interval B_2.

Remark. In a similar way, we can define the *left weak (Bruhat) order*, where $\tau \leq \sigma$ if and only if there exists a collection $s_{k_1} \ldots, s_{k_m}$ of simple transpositions such that $\sigma = s_{k_1} \ldots s_{k_m}\tau$. The two weak orders are distinct, but isomorphic under the map $\sigma \mapsto \sigma^{-1}$.

A subset $[\sigma, \tau] \subseteq W(\mathrm{S}_n)$ is an interval if $\sigma \leq \tau$ and $\pi \in [\sigma, \tau]$ whenever $\sigma \leq \pi \leq \tau$. As noted earlier in this chapter, Tenner established that Boolean intervals in $W(\mathrm{S}_n)$ have the structure $[v, w]$ if and only if $v^{-1}w$ is a permutation composed of only commuting generators [73, Corollary 4.4].

Example 9.2. In Figure 26, if $\pi \in \mathrm{S}_4$, then the interval $[\pi, \pi]$ is a Boolean interval of rank zero. In addition, all intervals $[\pi, \pi s_i]$ where $i \in \mathrm{Asc}(\pi)$ are Boolean intervals of rank one.

Finally, if $\mathrm{Asc}(\pi) = \{1, 3\}$, then the interval $[\pi, \pi s_1 s_3]$ is a Boolean interval of rank two. For example, the interval $[2314, 3241]$, which is highlighted in Figure 26, is one of the six Boolean intervals of rank two in $W(\mathrm{S}_4)$.

9.1.2 Unit interval parking functions and Fubini rankings

Hadaway and Harris introduced unit interval parking functions, which are a subset of parking functions in which cars park exactly at their preferred spot or one spot away [35]. For example, $(1, 2, 3, 4, 5)$, $(1, 1, 3, 4, 5)$, $(1, 1, 2, 4, 5)$ are unit interval parking functions (of length 5), whereas $(1, 1, 1, 1, 1)$ is a parking function but not a unit interval parking function. Let UPF_n denote the set of unit interval parking functions of length n. Hadaway and Harris established that the number of unit interval parking functions of length n is given by the Fubini numbers, also known as the ordered Bell numbers (OEIS A000670). That is,

$$\#\mathrm{UPF}_n = \mathrm{Fub}_n = \sum_{k=1}^{n} k!\, S(n, k), \tag{9.1}$$

where $S(n, k)$ are Stirling numbers of the second kind (OEIS A008277), which count the number of set partitions of $[n]$ with k non-empty parts.

To establish their result, Hadaway and Harris proved that the set of unit interval parking functions is in bijection with the set of *Fubini rankings*. A Fubini ranking of length n is a tuple $r = (r_1, r_2, \ldots, r_n) \in [n]^n$ that records a valid ranking over n competitors with ties allowed (i.e., multiple competitors can be tied and have the same rank). However, if k competitors are tied and rank ith, the $k - 1$ subsequent ranks $i + 1, i + 2, \ldots, i + k - 1$ are disallowed. For example, if two competitors are tied and rank first, the second rank is disallowed and the next available rank is the third[1]. Similarly, $(1, 1, 3, 3, 5)$, $(1, 2, 3, 4, 5)$, $(1, 1, 1, 1, 1)$, $(3, 1, 5, 1, 3)$ are all Fubini rankings (of length 5) while $(3, 1, 5, 1, 2)$ is not, as

[1] One noteworthy instance of this took place at the men's high jump event at the Summer 2020 Olympics [16]. In this competition, Mutaz Essa Barshim of Qatar and Gianmarco Tamberi of Italy led the final round. Both athletes cleared 2.37 meters but neither of them cleared 2.39 meters. Upon being presented the option of a "jump-off" to determine the sole winner, they agreed to instead share the gold medal. The next best rank was held by Maksim Nedasekau of Belarus, who obtained the bronze medal.

competitors 2 and 4 are tied and rank first, implying no competitor can rank second. Let FR_n denote the set of Fubini rankings of length n. Cayley [15] showed that $\#\mathrm{FR}_n = \mathrm{Fub}_n$, as in (9.1).

Note that by the definition of Fubini ranking, any rearrangement of a Fubini ranking is itself a Fubini ranking; as long as the distribution of ranks does not change, which competitor holds which rank is immaterial. In other words, Fubini rankings are invariant under permutations. As we reference this fact in a later section, we state it formally below.

Proposition 9.3. *Fubini rankings are invariant under permutations.*

9.2 Unit Fubini rankings

Despite the fact that the sets FR_n and UPF_n are in bijection, their intersection $\mathrm{FR}_n \cap \mathrm{UPF}_n$ is non-trivial for all $n > 1$. For example, $(1, 1, 2)$ is a unit interval parking function but not a Fubini ranking, $(1, 1, 1)$ is a Fubini ranking but not a unit interval parking function, while $(1, 1, 3)$ is both a Fubini ranking and a unit interval parking function.

Henceforth, we refer to the elements in $\mathrm{FR}_n \cap \mathrm{UPF}_n$ as *unit Fubini rankings*, and we denote this set by UFR_n. Note that elements in UFR_n are Fubini rankings with the additional constraint that ranks are shared by at most two competitors. Table 3 gives the cardinality of UFR_n for small values of n, agreeing with OEIS A080599, which Stanley identifies as the number of Boolean intervals in $W(\mathrm{S}_n)$. His remark motivates this work.

n	1	2	3	4	5	6	7
$\#\mathrm{UFR}_n$	1	3	12	66	450	3690	35280

Table 3: The number of unit Fubini rankings with $1 \leq n \leq 7$ competitors.

The following definition and result are due to Bradt, Elder, Harris, Rojas Kirby, Reutercrona, Wang, and Whidden [11], who gave a complete characterization of unit interval parking functions.

Definition 9.4 ([11]). Let $\alpha = (a_1, a_2, \ldots, a_n) \in \mathrm{UPF}_n$ and $\alpha' = (\alpha'_1, \alpha'_2, \ldots, \alpha'_n)$ be its weakly increasing rearrangement. Let $i_1, i_2, \ldots, i_m \in [n]$ be the increasing sequence of indices satisfying $\alpha'_{i_j} = i_j$. The partition of α' as the concatenation $b_1|b_2|\ldots|b_m$ where $b_j = (\alpha'_{i_j}, \alpha'_{i_j+1}, \ldots, \alpha'_{i_{j+1}-1})$ is called the *block structure* of α. Each part b_j for $j \in [m]$ is called a *block* of α.

Next, we state the characterization of unit parking functions by Bradt et al. [11, Theorem 2.9].

Theorem 9.5. *Given $\alpha = (a_1, \ldots, a_n) \in \mathrm{UPF}_n$, let α' be its weakly increasing rearrangement and $\alpha' = \pi_1 \,|\, \pi_2 \,|\, \ldots \,|\, \pi_m$ be the block structure of α (as in Definition 9.4).*

1. *There are*

$$\binom{n}{\#\pi_1, \ldots, \#\pi_m} \tag{9.2}$$

   possible rearrangements σ of α such that σ is still a unit interval parking function.

2. *A rearrangement σ of α is in UPF_n if and only if the entries in σ respect the relative order of the entries in each of the blocks $\pi_1, \pi_2, \ldots, \pi_m$.*

For our purposes, we only need the following result, which follows from Theorem 9.5.

Corollary 9.6. *Let $\alpha \in \mathrm{UPF}_n$ and $b_1 \,|\, b_2 \,|\, \cdots \,|\, b_m$ be its block structure. For each $j \in [m]$, let i_j be the minimal element of b_j. Consider any $j \in [m-1]$. If $\#b_j = 1$, then $b_j = (i_j)$ and $i_{j+1} = i_j + 1$. Otherwise, if $\#b_j = 2$, then $b_j = (i_j, i_j)$ and $i_{j+1} = i_j + 2$. Otherwise, $\#b_j \geq 3$, $b_j = (i_j, i_j, \underbrace{i_j + 1, i_j + 2, \ldots, i_j + \#b_j - 2}_{\#b_j - 2 \text{ terms}})$ and $i_{j+1} = i_j + \#b_j$.*

We now give a characterization of unit Fubini rankings based on their block structure. We employ this technical result in our proof of Theorem 9.20.

Theorem 9.7. *Let $\alpha \in \mathrm{UPF}_n$ and $b_1 \,|\, b_2 \,|\, \cdots \,|\, b_m$ be its block structure. Then, $\alpha \in \mathrm{UFR}_n$ if and only if $\#b_j \leq 2$ for each $j \in [m]$.*

Proof. First, suppose $\#b_j \leq 2$ for each $j \in [m]$. We need to show that $\alpha \in \mathrm{UFR}_n$. To do this, it suffices to show that for each pair b_j, b_{j+1} of consecutive blocks with $j \in [m-1]$, there being competitors whose ranks correspond to the block b_j does not disallow there being a competitor whose rank is the minimal element of block b_{j+1}. Consider any such pair b_j, b_{j+1} of consecutive blocks and let i_j and i_{j+1} be the minimal elements of blocks b_j and b_{j+1}, respectively. If $\#b_j = 1$, then by Corollary 9.6 we know that $b_j = (i_j)$ and $i_{j+1} = i_j + 1$, so there being a competitor whose rank is i_j does not disallow there being a competitor whose rank is $i_{j+1} = i_j + 1$. If $\#b_j = 2$, then by Corollary 9.6 we know that $b_j = (i_j, i_j)$ and $i_{j+1} = i_j + 2$, so there being two competitors whose ranks are both i_j does not disallow there being a competitor whose rank is $i_{j+1} = i_j + 2$.

Now, suppose $\#b_j = k > 2$ for some $j \in [m]$. We need to show that $\alpha \notin \mathrm{UFR}_n$. Let i_j be the minimal element of block b_j so that, by Corollary 9.6, $b_j = (i_j, i_j, i_j + 1, \ldots, i_j + k)$. Note that, in b_j, i_j appears twice while $i_j + 1$ appears once. Therefore, similarly in α, i_j appears twice while $i_j + 1$ appears once. This implies that $\alpha \notin \mathrm{UFR}_n$, since there being two competitors whose ranks are both i_jth disallows the subsequent rank $i_j + 1$, which some competitor supposedly holds. □

As a corollary, we give an inequality description of unit Fubini rankings.

Corollary 9.8. *Let $\alpha = (a_1, a_2, \ldots, a_n) \in [n]^n$ and $\alpha' = (a'_1, a'_2, \ldots, a'_n)$ be its weakly increasing rearrangement. Then, $\alpha \in \mathrm{UFR}_n$ if and only if $c_i \leq a'_i \leq i$ for each $i \in [n]$, where*

$$c_i = \begin{cases} 1, & \text{if } i = 1 \\ i, & \text{if } a'_{i-1} = i - 2 \text{ and } 2 \leq i \leq n \\ i - 1, & \text{else.} \end{cases}$$

Proof. First, let $\alpha \in \mathrm{UFR}_n$. Then, by Theorem 9.7, the block structure $b_1 \mid b_2 \mid \cdots \mid b_m$ of α satisfies $\#b_j \leq 2$ for each $j \in [m]$. This implies that $c_i \leq a'_i \leq i$ for each $i \in [n]$.

Now, let $\alpha \in [n]^n$ such that $c_i \leq a'_i \leq i$ for all $i \in [n]$. This implies that each number

$i \in [n]$ occurs at most twice in α. Moreover, if $i \in [n]$ occurs twice, then the next smallest number, if any, is $i+2$. This implies that the block structure $b_1 \mid b_2 \mid \cdots \mid b_m$ of α satisfies $\#b_j \leq 2$ for each $j \in [m]$. By Theorem 9.7, this implies $\alpha \in \mathrm{UFR}_n$. □

We now take a quick aside to provide a connection between unit Fubini rankings and the Fibonacci numbers (OEIS A000045), defined by $F_{n+1} = F_n + F_{n-1}$ for $n \geq 2$ and $F_1 = F_2 = 1$.

Theorem 9.9. *Let* $\mathrm{UFR}_n^{\uparrow}$ *be the set of weakly increasing unit Fubini rankings of length* n. *Then, for* $n \geq 1$ *we have*

$$\#\mathrm{UFR}_n^{\uparrow} = F_{n+1},$$

where F_{n+1} *is the* $(n+1)$*th Fibonacci number.*

Proof. We show that $\#\mathrm{UFR}_n^{\uparrow}$ satisfies the same recurrence relation as the Fibonacci numbers. That is, we show $\#\mathrm{UFR}_n^{\uparrow} = \#\mathrm{UFR}_{n-1}^{\uparrow} + \#\mathrm{UFR}_{n-2}^{\uparrow}$, $\#\mathrm{UFR}_2^{\uparrow} = 2$, and $\#\mathrm{UFR}_1^{\uparrow} = 1$. By Theorem 9.7, the block structure of any unit Fubini ranking has blocks of size at most two. Moreover, for any $n \in \mathbb{N}_{>0}$, each $\alpha \in \mathrm{UFR}_n$ satisfies $\#\{i \in [n] : a_i = n\} \leq 1$. That is, no two competitors can tie and rank nth over n competitors. Therefore, to compute $\#\mathrm{UFR}_n^{\uparrow}$, we need only consider forming a block of size two in which 2 participants tie and rank $n-1$ to any $\beta \in \mathrm{UFR}_{n-2}^{\uparrow}$, or appending a block of size one with rank n to any $\gamma \in \mathrm{UFR}_{n-1}^{\uparrow}$. These cases are disjoint and exhaustive, and therefore give the required recursion relation. To conclude, we note that $\#\mathrm{UFR}_1^{\uparrow} = \#\{(1)\} = 1$ and $\#\mathrm{UFR}_2^{\uparrow} = \#\{(1,1),(1,2)\} = 2$. □

Lastly, we describe a set of functions on unit Fubini rankings used in future sections to establish Theorem 9.20.

Definition 9.10. For each $i \in [n-1]$ define $\delta_i : \mathrm{UFR}_n \to \mathrm{UFR}_n$ given by

$$\delta_i(\alpha) = \begin{cases} \alpha, & \text{if } \#\{j : a_j = i-1\} = 2 \text{ or } \#\{j : a_j = i\} = 2 \text{ or } \#\{j : a_j = i+1\} = 2 \\ \widehat{\alpha}(i), & \text{otherwise;} \end{cases} \tag{9.3}$$

where $\widehat{\alpha}(i)$ is obtained from α by decreasing the singular occurrence of $i+1$ to i.

For example, if $\alpha = (1,3,5,3,6,1,7)$, then $\delta_i(\alpha) = \alpha$, for $1 \leq i \leq 4$, while

- $\delta_5(\alpha) = \widehat{\alpha}(5) = (1,3,5,3,5,1,7)$, because 4, 5, and 6 each occur only once in α and
- $\delta_6(\alpha) = \widehat{\alpha}(6) = (1,3,5,3,6,1,6)$, because 5, 6, and 7 each occur only once in α.

One can readily confirm that all of the tuples above are in UFR_7. This motivates the next result.

Lemma 9.11. *The functions δ_i for $i \in [n-1]$ are well-defined.*

Proof. Let $\alpha \in \mathrm{UFR}_n$ and let $b_1 \,|\, b_2 \,|\, \cdots \,|\, b_m$ be its block structure. Consider any fixed but arbitrary $i \in [n-1]$. We need to show that $\delta_i(\alpha) \in \mathrm{UFR}_n$. There are two possibilities.
Case 1: Suppose $\delta_i(\alpha) = \alpha$. The claim holds since $\alpha \in \mathrm{UFR}_n$, by assumption.
Case 2: Suppose $\delta_i(\alpha) = \widehat{\alpha}(i)$. By definition of δ_i this means that each of $i-1$, i, and $i+1$, whenever they appear in α, in fact appear exactly once. In addition, by Corollary 9.8, if $i+2 \leq n$, then $i+2$ appears at least once in α. Note the only change that δ_i makes to obtain $\widehat{\alpha}(i)$ from α occurs at the value $i+1$, which is decreased to i; all other entries of α remain unchanged. Therefore, the only change that δ_i makes to the block structure $b_1 \,|\, b_2 \,|\, \cdots \,|\, b_m$ is that the singleton block containing (i) and the (adjacent) singleton block containing $(i+1)$ are turned into a single block of size 2 containing (i,i). Then, Corollary 9.6 guarantees that $\widehat{\alpha}(i) \in \mathrm{UPF}_n$ while, in turn, Theorem 9.7 guarantees that $\widehat{\alpha}(i) \in \mathrm{UFR}_n$, as claimed. □

Next we show that the functions of Definition 9.10 commute whenever their domain is restricted to the set of permutations and are applied on nonconsecutive indices.

Theorem 9.12. *Let $i, j \in [n-1]$ be nonconsecutive. If $\pi \in \mathrm{S}_n$, then $\delta_i(\delta_j(\pi)) = \delta_j(\delta_i(\pi))$.*

Proof. Fix any pair of nonconsecutive integers $i, j \in [n-1]$. Without loss of generality, let $i < j$. By Lemma 9.3, it suffices to consider only the identity permutation $\pi = 12\cdots n$. Note that the block structure of π is $b_1 \,|\, b_2 \,|\, \ldots \,|\, b_n$ with singleton blocks $b_i = (i)$ for each $i \in [n]$.

Note that f $\delta_i(\pi)$ has the block structure $1\,|\,2\,|\,\cdots\,|\,i-1\,|\,i\,i\,|\,i+2\,|\,\cdots\,|\,n-1\,|\,n$. Then, since $i<j$, $\delta_j(\delta_i(\pi))$ has the block structure

$$1\,|\,2\,|\,\cdots\,|\,i-1\,|\,i\,i\,|\,i+2\,|\,\cdots\,|\,j-1\,|\,j\,j\,|\,j+2\,|\,\cdots\,|\,n-1\,|\,n.$$

Note if $i+2=j$, then the block structure would be

$$1\,|\,2\,|\,\cdots\,|\,i-1\,|\,i\,i\,|\,j\,j\,|\,j+2\,|\,\cdots\,|\,n-1\,|\,n.$$

On the other hand, $\delta_j(\pi)$ has the block structure

$$1\,|\,2\,|\,\cdots\,|\,j-1\,|\,j\,j\,|\,j+2\,|\,\cdots\,|\,n-1\,|\,n.$$

Then, since $i<j$, $\delta_i(\delta_j(\pi))$ has the block structure

$$1\,|\,2\,|\,\cdots\,|\,i-1\,|\,i\,i\,|\,i+2\,|\,\cdots\,|\,j-1\,|\,j\,j\,|\,j+2\,|\,\cdots\,|\,n-1\,|\,n.$$

Again, if $i+2=j$, then the block structure would be

$$1\,|\,2\,|\,\cdots\,|\,i-1\,|\,i\,i\,|\,j\,j\,|\,j+2\,|\,\cdots\,|\,n-1\,|\,n.$$

Therefore, for $\pi=12\cdots n$, then $\delta_i(\delta_j(\pi))=\delta_j(\delta_i(\pi))$. Finally, note that for any $\pi\neq 12\cdots n$, the blocks (i,i) and (j,j) will be in the positions where the consecutive blocks $\cdots\,|\,i\,|\,i{+}1\,|\,\cdots$ and $\cdots\,|\,j\,|\,j+1\,|\,\cdots$ originally appeared, respectively. □

Remark. In Theorem 9.12, it is important that i and j are nonconsecutive. To see this, let $\pi\in \mathrm{S}_n$ and $j=i+1$. Then, the block structure of π changes in the following way upon application of δ_{i+1} followed by δ_i:

$$\delta_i(\delta_{i+1}(\pi))=\delta_i(\cdots\,|\,i-1\,|\,j\,j\,|\,i+2\,|\,\cdots)=\cdots\,|\,i-1\,|\,j\,j\,|\,i+2\,|\,\cdots\,. \tag{9.4}$$

On the other hand, the block structure of π changes in the following way upon application of δ_i followed by δ_{i+1}:

$$\delta_{i+1}(\delta_i(\pi))=\delta_{i+1}(\cdots\,|\,i-1\,|\,i\,i\,|\,i+2\,|\,\cdots)=\cdots\,|\,i-1\,|\,i\,i\,|\,i+2\,|\,\cdots\,. \tag{9.5}$$

Equations (9.4) and (9.5) show that $\delta_{i+1}(\delta_i(\pi))\neq\delta_i(\delta_{i+1}(\pi))$.

We now generalize the composition of the functions of Definition 9.10 to subsets consisting of nonconsecutive integers.

Definition 9.13. Let $I = \{i_1, i_2, \ldots, i_k\} \subset [n-1]$ be a set of pairwise nonconsecutive integers satisfying $i_1 < i_2 < \cdots < i_k$. If $\pi \in \mathrm{S}_n$, then we define the composition

$$\delta_I(\pi) \coloneqq \delta_{i_1} \circ \delta_{i_2} \circ \cdots \circ \delta_{i_k}(\alpha). \tag{9.6}$$

If $I = \emptyset$, then $\delta_I = \mathrm{Id}$ is the identity map on S_n.

Next we show that the composition defined in Equation (9.6) can be done in any order.

Corollary 9.14. *Let $I = \{i_1, i_2, \ldots, i_k\} \subseteq [n-1]$ be a set of nonconsecutive integers. If $\pi \in \mathrm{S}_n$, then the composition $\delta_I(\pi) \in \mathrm{UFR}_n$.*

Proof. Upon repeated application, Theorem 9.12 implies that if $I = \{i_1, i_2, \ldots, i_k\} \subset [n-1]$ consists of pairwise nonconsecutive integers and $\pi \in \mathrm{S}_n$, then the composition

$$\delta_{i_1} \circ \delta_{i_2} \circ \cdots \circ \delta_{i_k}(\pi) \tag{9.7}$$

is commutative. □

9.3 Bijection

By Theorem 9.7, $\mathrm{UFR}_n \subseteq \mathrm{UPF}_n$, hence, we can treat unit Fubini rankings as parking functions. We define the outcome map $\mathcal{O} : \mathrm{UFR}_n \to \mathrm{S}_n$ by $\mathcal{O}(\alpha) = \pi = \pi_1\pi_2\cdots\pi_n$ where $\pi \in \mathrm{S}_n$ is written in one-line notation and denotes the order in which the cars park on the street. That is, if $j \in [n]$, then $\pi_j = i$ denotes that car i is the jth car parked on the street. Given $\pi \in \mathrm{S}_n$, we define the fiber of the outcome map:

$$\mathcal{O}^{-1}(\pi) = \{\alpha \in \mathrm{UFR}_n : \mathcal{O}(\alpha) = \pi\}.$$

Proposition 9.15. *Since no car can park in more than one spot, $\mathcal{O}$ is a well-defined map.*

In what follows, we write both Fubini rankings and permutations in one-line notation. We now provide some initial technical results.

Lemma 9.16. *Let $\pi \in \mathrm{S}_n$. Then $\alpha = \pi^{-1}$ is the unique permutation with outcome π.*

Proof. Let $\pi = \pi_1 \cdots \pi_n \in \mathrm{S}_n$. Suppose that $\pi_i = j$. That is, car j parked in spot i. Because we wish to find the permutation with parking outcome π, this means we will restrict to car j having preference i. That is, we need the jth entry of α to be equal to i in order to produce the outcome π. Note, that $\pi_j^{-1} = i$. Since this was an arbitrary entry in π, we have that $\alpha = \pi^{-1}$, as desired. We note that since permutation inverses are unique, there is only one permutation $\alpha \in \mathcal{O}^{-1}(\pi)$. □

Next, we provide the connection between the elements in $\mathcal{O}^{-1}(\pi)$ and the set $\mathrm{Asc}(\pi)$.

Lemma 9.17. *Let $\pi = \pi_1\pi_2 \cdots \pi_n \in \mathrm{S}_n$. If $j \in \mathrm{Asc}(\pi)$, $\pi_{j+1} = i$, and $\alpha = (a_1, a_2, \ldots, a_n) \in \mathcal{O}^{-1}(\pi)$, then $a_i \in \{j, j+1\}$.*

Proof. Assume $j \in \mathrm{Asc}(\pi)$, which implies that $\pi_j < \pi_{j+1}$. This means that car $\pi_{j+1} = i$ arrived after car π_j and is parked immediately to the right of π_j. Under unit interval parking rule, there are only two ways in which car i can park in spot $j + 1$, either spot $j + 1$ was its preference and that spot was available, or its preference was the spot j, which it found occupied by car π_j. Thus $a_i \in \{j, j + 1\}$ as desired. These are the only preferences which ensure α is a unit Fubini ranking and which would result in car i parking in spot $j+1$, which is required so that α has outcome π. □

Proposition 9.18. *Let $\pi = \pi_1\pi_2 \cdots \pi_n \in \mathrm{S}_n$ and $\alpha = \pi^{-1} \in \mathcal{O}^{-1}(\pi)$. Then*

$$\mathcal{O}^{-1}(\pi) = \{\delta_I(\alpha) : I \subseteq \mathrm{Asc}(\pi) \textit{ with nonconsecutive entries}\}.$$

Before we prove Proposition 9.18, we illustrate the effect of δ_I on a permutation π, when I is a subset of nonconsecutive elements from $\mathrm{Asc}(\pi)$.

Example 9.19. Fix $\pi = 412356$ and note $\text{Asc}(\pi) = \{2,3,4,5\}$. Then $\alpha = \pi^{-1} = 234156$ is the unique permutation in $\mathcal{O}^{-1}(\pi)$. Observe that the only possible subsets of $\text{Asc}(\pi) = \{2,3,4,5\}$ consisting of nonconsecutive integers are: $\emptyset$, $\{2\}$, $\{3\}$, $\{4\}$, $\{5\}$, $\{2,4\}$, $\{2,5\}$, and $\{3,5\}$. Note, that

$$\delta_\emptyset(\alpha) = 234156 \qquad \delta_{\{2\}}(\alpha) = 224156 \qquad \delta_{\{3\}}(\alpha) = 233156 \qquad \delta_{\{4\}}(\alpha) = 234146$$
$$\delta_{\{5\}}(\alpha) = 234155 \qquad \delta_{\{2,4\}}(\alpha) = 224146 \qquad \delta_{\{2,5\}}(\alpha) = 224155 \qquad \delta_{\{3,5\}}(\alpha) = 233155.$$

Straightforward computations establish that the results are unit Fubini rankings with outcome π. Moreover, Theorem 9.12 and the subsequent remark establish that if we take any subset of $\text{Asc}(\pi)$ that contains consecutive integers, the result would not yield any unit Fubini rankings not found in the above list. Together this confirms that for any subset of $\text{Asc}(\pi)$ consisting of nonconsecutive integers, $\delta_I(\alpha) \in \mathcal{O}^{-1}(\pi)$.

Now note that $\delta_1(\alpha) = 134156$ and $\mathcal{O}(134156) = 142356 \neq \pi$. Hence $\delta_1(\alpha) \notin \mathcal{O}^{-1}(\pi)$. Establishing that $\delta_j(\alpha) \notin \mathcal{O}^{-1}(\pi)$ when $j \in \text{Des}(\pi)$.

Proof of Proposition 9.18. It suffices to show

1. $\mathcal{O}^{-1}(\pi) \subseteq \{\delta_I(\pi^{-1}) : I \subseteq \text{Asc}(\pi) \text{ with nonconsecutive entries}\}$ and

2. $\{\delta_I(\pi^{-1}) : I \subseteq \text{Asc}(\pi) \text{ with nonconsecutive entries}\} \subseteq \mathcal{O}^{-1}(\pi)$.

For (1): Let $\beta \in \mathcal{O}^{-1}(\pi)$ such that the block structure of β contains exactly 1 block of size two. Let the entries of that block be ii. We note that if i appears twice in β, then there must be an ascent at position i in π. We must also have that $\delta_i(\pi^{-1}) = \beta$. Therefore, $\beta \in \{\delta_I(\pi^{-1}) : I \subseteq \text{Asc}(\pi)$ with nonconsecutive entries for $I = \{i\}\}$.

Inductively, for any $\beta \in \mathcal{O}^{-1}(\pi)$ with k blocks of size 2, we can reconstruct the set I by looking at the entries in those k blocks. The indices in I must all be more than one unit away, are determined by the minimum element in each block of size two, and must have all come from the ascent set of π. Thus $\delta_I(\pi^{-1}) = \beta$, which means that $\beta \in \{\delta_I(\pi^{-1}) : I \subseteq \text{Asc}(\pi)$ with nonconsecutive entries for $I = \{i\}\}$.

For (2): Let $I = \{i_1, i_2, \ldots, i_k\} \subseteq \mathrm{Asc}(\pi)$ consist of nonconsecutive integers. Without loss of generality assume $i_1 < i_2 < \cdots < i_k$. By Corollary 9.14 we know $\delta_I(\pi^{-1}) \in \mathrm{UFR}_n$, and the block structure of $\delta_I(\pi^{-1})$ is as follows:

- For each $i \in I$, there is a block of size two containing both instances of i in $\delta_I(\pi^{-1})$, and
- for each $i \notin I$, there is a block of size one containing the only instance i in $\delta_I(\pi^{-1})$.

Since the entries in I are nonconsecutive, the block structure of $\delta_I(\pi^{-1})$ ensures that if $i \notin I$, car π_i with preference i parks in spot i, as needed to have outcome π. Moreover, if $i \in I$, then under $\delta_I(\pi^{-1})$, car π_i has preference i and parks in spot i, and car π_{i+1} has preference i and as $\pi_i < \pi_{i+1}$ it finds spot i occupied and parks in spot $i+1$, as needed to have outcome π. Thus establishing that $\mathcal{O}(\delta_I(\pi^{-1})) = \pi$, as desired. □

Tenner established that Boolean intervals in the weak order all have the form $[v, w]$ where $w = v \prod_{i \in I} s_i$ for some $I \subseteq \mathrm{Asc}(v)$ whose elements are nonconsecutive [73, Corollary 4.4]. We use this result in the proof of the following.

Theorem 9.20. *The set of unit Fubini rankings with $n-k$ distinct ranks is in bijection with the set of Boolean intervals in $W(\mathrm{S}_n)$ of rank k.*

Proof. Fix $\pi \in \mathrm{S}_n$. Let $\mathcal{B}_n$ be the set of all Boolean intervals in $W(\mathrm{S}_n)$, and $\mathcal{B}_n(\pi)$ denote the set of all Boolean intervals in $W(\mathrm{S}_n)$ with minimal element π. Define the map $\varphi_\pi : \mathcal{O}^{-1}(\pi) \to \mathcal{B}_n(\pi)$ defined by

$$\varphi_\pi(\beta) = [\pi, \pi \prod_{i \in I} s_i]$$

where $I \subseteq \mathrm{Asc}(\pi)$ of nonconsecutive integers is determined by $\beta = \delta_I(\pi^{-1})$. Namely, the set I consists of the repeated values in β, which is unique by Proposition 9.18. We begin by establishing that φ_π is a bijection.

The output $\varphi_\pi(\beta)$ is computed using the unique set I associated with each β, and hence is unique. Furthermore, the output $[\pi, \pi \prod_{i \in I} s_i] \in \mathcal{B}_n$ is a Boolean interval [73, Corollary 4.4]. Therefore φ_π is well-defined.

For injectivity: If $\varphi_\pi(\beta) = \varphi_\pi(\gamma) = [\pi, \pi \prod_{i \in I} s_i]$ for some (nonconsecutive) $I \subseteq \text{Asc}(\pi)$, then $\delta_I(\pi^{-1}) = \beta$ and $\delta_I(\pi^{-1}) = \gamma$. Therefore, $\beta = \gamma$.

For surjectivity: Every Boolean interval in $\mathcal{B}_n(\pi)$ has the form $[\pi, \pi \prod_{i \in I} s_i]$ where $I \subseteq \text{Asc}(\pi)$ consists of nonconsecutive integers [73, Corollary 4.4]. Then, by Proposition 9.18, we know that $\delta_I(\pi^{-1}) \in \mathcal{O}^{-1}(\pi)$. Then $\varphi_\pi(\delta_I(\pi^{-1})) = [\pi, \pi \prod_{i \in I} s_i]$.

Together, this establishes that the map φ_π is a bijection.

Now define $\phi : \text{UFR}_n \to \mathcal{B}_n$ by $\phi(\alpha) \coloneqq \varphi_\pi(\alpha)$ where $\mathcal{O}(\alpha) = \pi$. Note that since φ_π is a bijection for all π and since $\mathcal{O}$ is well-defined (Proposition 9.15), then ϕ is a bijection.

To conclude, we establish that φ_π preserves the statistic of $n-k$ distinct ranks in $\mathcal{O}^{-1}(\pi)$ and rank k in the Boolean interval. Let $\beta \in \text{UFR}_n$ such that $\mathcal{O}(\beta) = \pi$ where ties occur at ranks denoted by $r_1, r_2, \ldots, r_k$. Note, that β then has $n-k$ distinct ranks. Then, by Proposition 9.18, the set $I = \{r_1, r_2, \ldots, r_k\}$, is a subset of $\text{Asc}(\pi)$ consisting of k nonconsecutive integers, and $\delta_I(\pi^{-1}) = \beta$. Then $\varphi_\pi(\beta)$ corresponds uniquely to the rank k Boolean interval given by $[\pi, \pi \prod_{i \in I} s_i]$. □

9.4 Enumerations

In this section, we provide enumerative formulas for:

1. $f(n)$, the total number of Boolean intervals in $W(\mathrm{S}_n)$,

2. $f(n,k)$, the total number of rank k Boolean intervals in $W(\mathrm{S}_n)$, and

3. the number of Boolean intervals in $W(\mathrm{S}_n)$ with minimal element π .

To establish (1), we begin with an immediate consequence of Theorem 9.20.

Corollary 9.21. *The total number of Boolean intervals in $W(\mathrm{S}_n)$ is equal to the number of unit Fubini rankings of length n.*

By setting $q = 1$ into the exponential generating function [68, Exercise 3.185(h)]

$$F(x,q) = \sum_{n \geq 0} \sum_{k \geq 0} f(n,k) q^k \frac{x^n}{n!} = \frac{1}{1 - x - \frac{q}{2}x^2}, \tag{9.8}$$

Stanley [65] points out that the *total* number of Boolean intervals in $W(\mathrm{S}_n)$ (OEIS A080599) satisfies the recurrence relation

$$f(n+1) = (n+1)f(n) + \binom{n+1}{2} f(n-1), \tag{9.9}$$

where $f(0) = 1$ and $f(1) = 1$. In light of Corollary 9.21, we give a combinatorial proof of this result from the perspective of unit Fubini rankings.

Theorem 9.22. *Let $g(n+1)$ denote the number of unit Fubini rankings of length $n+1$. Then $g(n+1)$ satisfies the recursion*

$$g(n+1) = (n+1)g(n) + \binom{n+1}{2} g(n-1),$$

where $g(1) = 1$ and $g(2) = 3$.

Proof. Let α be unit Fubini ranking of length n. The block structure of an element in UFR_n means we have two options for the final block: it either ends in an $(n-1)(n-1)$ or an n. We have total freedom in the remaining positions. Thus there are two mutually exclusive cases to consider.

- The last block has the form $(n-1)(n-1)$: Then we may select one of the $g(n-1)$ unit Fubini rankings in UFR_{n-1}. Place the elements in the unit Fubini rankings in any of the $n+1$ possible spots for the unit Fubini ranking of length $n+1$. For each unit Fubini ranking in UFR_{n-1} there are
$$\binom{n+1}{n-1} = \binom{n+1}{2}$$
ways to do this.

- The last block has the form n: Then we may select one of the $g(n)$ unit Fubini rankings in UFR_n. Place the elements in the unit Fubini ranking in any of the $n+1$ possible spots for the unit Fubini ranking of length $n+1$. For each unit Fubini ranking in UFR_n there are
$$\binom{n+1}{n} = n+1$$
ways to do this.

The recursion follows from taking the sum of the counts in each case. The initial values arise from the fact that $\#\mathrm{UFR}_1 = \{(1)\}$, hence $g(1) = 1$, and $\#\mathrm{UFR}_2 = \{(1,1),(1,2),(2,1)\}$, hence $g(2) = 3$. $\square$

For (2), we begin with the following combinatorial proof.

Theorem 9.23. *Let $f(n,k)$ denote the number of Boolean intervals in $W(\mathrm{S}_n)$ of rank k. Then,*

$$f(n,k) = \frac{n!}{2^k}\binom{n-k}{k}. \tag{9.10}$$

Proof. Let $g(n,k)$ denote the number of unit Fubini rankings of length n which have $n-k$ distinct ranks. Note that Theorem 9.20 implies that $g(n,k) = f(n,k)$, hence it suffices to show that $g(n,k) = \frac{n!}{2^k}\binom{n-k}{k}$.

If $\alpha \in \mathrm{UFR}_n$ has $n-k$ distinct ranks, then its block structure has the form

$$b_1 \mid b_2 \mid \cdots \mid b_{n-k},$$

where exactly k of the blocks have size two and all remaining blocks have size one. To enumerate all such α, first select the indices of the blocks with size two. We can do this in $\binom{n-k}{k}$ ways. To enumerate, we begin by selecting the indices at which we place the repeated values within the blocks of size two. We do so iteratively by first selecting two indices among n where we will place the smallest repeated values of α. This can be done in $\binom{n}{2}$ ways. Then we repeat this process by selecting two indices among the remaining $n-2$ indices in which we place the next smallest repeated values of α. This can be done in $\binom{n-2}{2}$ ways. Through this process, the total ways in which we can place all repeated values in α is given by the product

$$\binom{n}{2}\binom{n-2}{2}\cdots\binom{n-2(k-1)}{2} = \prod_{i=0}^{k-1}\binom{n-2i}{2}.$$

Finally, we note that the values in the blocks of size one can appear in any order within the remaining available indices. We can do this in $(n-2k)!$ ways. Thus

$$g(n,k) = \binom{n-k}{k}(n-2k)!\prod_{i=0}^{k-1}\binom{n-2i}{2},$$

which simplifies to our desired result. $\square$

Remark. In the introduction we referenced OEIS A001286, a sequence known as the Lah numbers, which gives the values $f(n,1) = \frac{(n-1)n!}{2}$ for the number of B_1 in $W(\mathrm{S}_n)$. Theorem 9.23 implies that the Lah numbers also enumerate unit Fubini rankings with $n-1$ distinct ranks. Aguillon et al. [76] showed that the number of unit interval parking functions in which exactly $n-1$ cars park in their preference is also enumerated by the Lah numbers. This result was established via a bijection between those parking functions and ideal states in the game the Tower of Hanoi, which were enumerated by the Lah numbers.

We now prove that $g(n,k)$ has the same generating function as (9.8).

Theorem 9.24. *The exponential generating function for $g(n,k)$ has the closed form*

$$G(x,q) = \sum_{n\geq 0}\sum_{k\geq 0} g(n,k)q^k\frac{x^n}{n!} = \frac{1}{1-x-\frac{q}{2}x^2}.$$

Proof. From Theorem 9.23, we know that $g(n,k) = \frac{n!}{2^k}\binom{n-k}{k}$. Then

$$G(x,q) = \sum_{n\geq 0}\sum_{k\geq 0} g(n,k)q^k\frac{x^n}{n!} = \sum_{n\geq 0}\sum_{k\geq 0}\frac{1}{2^k}\binom{n-k}{k}q^k x^n. \tag{9.11}$$

Note that, for the purpose of counting objects, $\binom{n}{k} = 0$ whenever $k > n$ or n is negative. Setting $n = 0$ in Equation (9.11) yields

$$\sum_{k\geq 0}\frac{1}{2^k}\binom{-k}{k}q^k x^0 = 1 + \sum_{k\geq 1}\frac{1}{2^k}\binom{-k}{k}q^k = 1+0. \tag{9.12}$$

Substituting (9.12) into (9.11) gives

$$G(x,q) = 1 + \sum_{n\geq 1}\sum_{k\geq 1}\frac{1}{2^k}\binom{n-k}{k}q^k x^n. \tag{9.13}$$

Using the binomial identity $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$, (9.13) becomes

$$G(x,q) = 1 + \sum_{n\geq 1}\sum_{k\geq 1}\frac{1}{2^k}\left(\binom{n-k-1}{k} + \binom{n-k-1}{k-1}\right)q^k x^n, \tag{9.14}$$

which can be rewritten as

$$G(x,q) = 1 + \sum_{n\geq 1}\sum_{k\geq 1}\frac{1}{2^k}\binom{n-k-1}{k}q^kx^n + \sum_{n\geq 1}\sum_{k\geq 1}\frac{1}{2^k}\binom{n-k-1}{k-1}q^kx^n. \tag{9.15}$$

We note that the first set of summands in (9.15) simplifies in the following way:

$$\sum_{n\geq 1}\sum_{k\geq 1}\frac{1}{2^k}\binom{n-k-1}{k}q^kx^n = x\sum_{n\geq 1}\sum_{k\geq 1}\frac{1}{2^k}\binom{(n-1)-k}{k}q^kx^{n-1} \tag{9.16}$$

$$= x\sum_{n\geq 0}\sum_{k\geq 0}\frac{1}{2^k}\binom{n-k}{k}q^kx^n, \tag{9.17}$$

where the last equality in (9.16) follows from re-indexing with respect to n, and the fact that $\binom{n}{k} = 0$ whenever $k > n$.

We note that the second set of summands in (9.15) simplifies in the following way:

$$\sum_{n\geq 1}\sum_{k\geq 1}\frac{1}{2^k}\binom{n-k-1}{k-1}q^kx^n = \frac{q}{2}x^2\sum_{n\geq 1}\sum_{k\geq 1}\frac{1}{2^{k-1}}\binom{(n-2)-(k-1)}{k-1}q^{k-1}x^{n-2}$$

$$= \frac{q}{2}x^2\sum_{n\geq 0}\sum_{k\geq 0}\frac{1}{2^k}\binom{n-k}{k}q^kx^n, \tag{9.18}$$

where the last equality in (9.18) follows from re-indexing with respect to n and k.

Substituting (9.16) and (9.18) into (9.15) allows us to reassemble everything to arrive at

$$G(x,q) = 1 + xG(x,q) + \frac{q}{2}x^2G(x,q),$$

from which we arrive at

$$G(x,q) = \frac{1}{1-x-\frac{q}{2}x^2}. \qquad \square$$

We now present our final enumerative result settling (3), which further connects this work to Fibonacci numbers.

Theorem 9.25. *Let $\pi = \pi_1\pi_2\cdots\pi_n \in \mathrm{S}_n$ be in one-line notation and partition its ascent set $\mathrm{Asc}(\pi) = \{i \in [n-1] : \pi_i < \pi_{i+1}\}$ into maximal blocks $b_1, b_2, \ldots, b_k$ of consecutive entries.*

Then, the number of Boolean intervals $[\pi, w]$ *in* $W(\mathrm{S}_n)$ *with fixed minimal element* π *and arbitrary maximal element* w *(including the case* $\pi = w$*) is given by*

$$\prod_{i=1}^{k} F_{\#b_i+2},$$

where F_ℓ *is the* ℓ*th Fibonacci number, and* $F_1 = F_2 = 1$.

Proof. It is straightforward to prove that the number of ways to select nonconsecutive entries from the set $[n]$ is given by F_{n+2}. Thus, for each $i \in [k]$, the number of ways to select nonconsecutive elements from b_i is given by $F_{\#b_i+2}$. As the blocks $b_1, b_2, \ldots, b_k$ are pairwise disjoint, the total number of ways to select subsets from $\cup_{i=1}^{k} b_i$ consisting of nonconsecutive integers is given by $\prod_{i=1}^{k} F_{\#b_i+2}$, as desired. □

Among the many results established by Tenner concerning Boolean intervals in both the Bruhat order and in the weak order [73], we highlight the following.

Proposition 9.26. *[73, Proposition 5.9] Let* $i \in [n-1]$ *be fixed. The number of Boolean intervals in* $W(\mathrm{S}_n)$ *of the form* $[s_i, w]$ *is* $F_{i+1}F_{n-i+1}$*, where* F_i *is the* i*th Fibonacci number.*

Note that for any $i \in [n-1]$, we have that $\mathrm{Asc}(s_i) = [n] \setminus \{i\}$. Then $b_1 = [i-1]$ and $b_2 = \{i+1, i+2, \ldots, n-1\}$, and Theorem 9.25 implies that the number of Boolean intervals in $W(\mathrm{S}_n)$ with minimal element s_i is given by $F_{\#b_1+2} = F_{\#b_2+2} = F_{i+1}F_{n-i+1}$, recovering [73, Proposition 5.9].

We remark that in the statement of Theorem 9.25, we allow $[\pi, \pi]$ to be a Boolean interval. If we impose the condition that the maximal element w cannot be equal to the minimal element π, then we have the following.

Corollary 9.27. *Let* $\pi = \pi_1\pi_2\cdots\pi_n \in \mathrm{S}_n$ *be in one-line notation and partition its ascent set* $\mathrm{Asc}(\pi) = \{i \in [n-1] : \pi_i < \pi_{i+1}\}$ *into maximal blocks* $b_1, b_2, \ldots, b_k$ *of consecutive entries. Then, the number of Boolean intervals* $[\pi, w]$ *in* $W(\mathrm{S}_n)$ *with* $w \neq \pi$ *is given by*

$$\left(\prod_{i=1}^{k} F_{\#b_i+2}\right) - 1,$$

where F_ℓ *is the* ℓ*th Fibonacci number and* $F_1 = F_2 = 1$.

Proof. The result follows from Theorem 9.25, and noting that in creating a subset of $\mathrm{Asc}(\pi)$ consisting of nonconsecutive elements we cannot utilize the empty set. □

9.5 Future work

We gave a combinatorial proof of this result via the enumeration of unit Fubini rankings with $n-k$ distinct ranks. We wonder whether this new proof and combinatorial objects might shed light on how a symmetric group proof may be constructed.

As noted at the end of Section 9.4, Tenner has provided many results for intervals in the weak (Bruhat) order [73]. The paper also provides results on the Bruhat order, which leads us to wonder if there are other connections from Fubini rankings that can be used to count intervals in the Bruhat order. We also wonder if it may be possible to utilize unit Fubini rankings, or a slight generalization thereof, to enumerate Boolean intervals in Bruhat and weak orders of other Coxeter systems. To this end we state the following: How many Boolean intervals are there in the weak order of the hyperoctahedral group (type B Coxeter group)?

REFERENCES

[1] M. Abramson. A simple solution of Simon Newcomb's problem. *J. Combin. Theory Ser. A*, 18:223–225, 1975.

[2] A. Amanbayeva and D. Wang. The convex hull of parking functions of length n. *Enumer. Comb. Appl.*, 2(2):Paper No. S2R10, 10, 2022.

[3] W. W. Bein, P. Brucker, J. K. Park, and P. K. Pathak. A Monge property for the d-dimensional transport problem. *Discrete Appl. Math.*, 58(2):97–109, 1995.

[4] C. Benedetti, R. S. González D'León, C. R. H. Hanusa, P. E. Harris, A. Khare, A. H. Morales, and M. Yip. A combinatorial model for computing volumes of flow polytopes. *Trans. Amer. Math. Soc.*, 372(5):3369–3404, 2019.

[5] S. Billey, K. Burdzy, and B. Sagan. Permutations with given peak set. *J. Integer Seq.*, 16, 09 2012.

[6] A. Björner. Shellable and Cohen-Macaulay partially ordered sets. *Trans. Amer. Math. Soc.*, 260(1):159–183, 1980.

[7] A. Björner and T. Ekedahl. On the shape of Bruhat intervals. *Ann. of Math.*, 170:799–817, 2005.

[8] A. Björner and M. Wachs. On lexicographically shellable posets. *Trans. Amer. Math. Soc.*, 277(1):323–341, 1983.

[9] A. Björner and M. Wachs. Shellable nonpure complexes and posets. I. *Trans. Amer. Math. Soc.*, 348(4):1299–1327, 1996.

[10] R. Bourn and J. F. Willenbring. Expected value of the one-dimensional earth mover's distance. *Algebr. Stat.*, 11(1):53–78, 2020.

[11] S. A. Bradt, J. Elder, P. E. Harris, G. Rojas Kirby, E. Reutercrona, Y. (S.) Wang, and J. Whidden. Unit-interval parking functions and the r-Fubini numbers, 2023. Submitted.

[12] W. Bruns and H. J. Herzog. *Cohen-Macaulay Rings*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 1998.

[13] P. J. Cameron. *Combinatorics: topics, techniques, algorithms*. Cambridge University Press, 1994.

[14] E. Carlsson and A. Mellit. A proof of the shuffle conjecture. *J. Amer. Math. Soc.*, 31(3):661–697, 2018.

[15] A. Cayley. *On the analytical forms called Trees, with application to the theory of chemical combinations*, volume 9 of *Cambridge Library Collection - Mathematics*, page 427–460. Cambridge University Press, 2009.

[16] International Olympic Committee. Results of the 2020 olympics, men's high jump. `https://olympics.com/en/olympic-games/tokyo-2020/results/athletics/men-s-high-jump`. Online, accessed 06-12-2023.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2022.

[18] P. Diaconis and A. Hicks. Probabilizing parking functions. *Adv. in Appl. Math.*, 89:125–155, 2017.

[19] J. F. Dillon and D. P. Roselle. Simon Newcomb's problem. *SIAM J. Appl. Math.*, 17:1086–1093, 1969.

[20] R. Ehrenborg and A. Happ. Parking cars of different sizes. *Amer. Math. Monthly*, 123(10):1045–1048, 2016.

[21] J. Elder, P. E. Harris, J. Kretschmann, and J. C. Martínez Mori. Boolean intervals in the weak order of $\mathfrak{S}_n$. *arXiv preprint arXiv:2306.14734*, 2023.

[22] J. Elder, P. E. Harris, J. Kretschmann, and J. C. Martínez Mori. Cost-sharing in parking games. *arXiv preprint arXiv:2309.12265*, 2023.

[23] A. Erdélyi, W. Magnus, F. Oberhettinger, and F. G. Tricomi. *Higher transcendental functions. Vol. I*. Robert E. Krieger Publishing Co., Inc., Melbourne, Fla., 1981.

[24] W. Q. Erickson. A generalization for the expected value of the earth mover's distance. *Algebr. Stat.*, 12(2):139–166, 2021.

[25] W. Q. Erickson and J. Kretschmann. The structure and normalized volume of monge polytopes. *arXiv preprint arXiv:2311.07522*, pages 1–17, 2023.

[26] W. Q. Erickson and J. Kretschmann. The sum of all width-one matrices. *arXiv preprint arXiv:2303.08924 v1*, pages 1–14, 2023.

[27] W. Q. Erickson and J. Kretschmann. The sum of width-one tensors. *Enumer. Comb. Appl.*, 4(1):Paper No. S2R7, 2023.

[28] W. Q. Erickson and J. Kretschmann. The sum of all width-one matrices. *European J. Combin.*, 115:103799, 2024.

[29] M. Foley and C. A. R. Hoare. Proof of a recursive program: Quicksort. *Comput. J.*, 14(4):391–395, 1971.

[30] S. J. Franks, P. E. Harris, K. J. Harry, J. Kretschmann, and M. Vance. Counting parking sequences and parking assortments through permutations. *Enumer. Comb. Appl.*, 4(1):Paper No. S2R2, 10, 2023.

[31] A. Frohmader and H. Volkmer. 1-wasserstein distance on the standard simplex. *Algebr. Stat.*, 12(1):43–56, 2021.

[32] W. Fulton. *Young Tableaux: With Applications to Representation Theory and Geometry.* London Mathematical Society Student Texts. Cambridge University Press, 1996.

[33] A. M. Garsia and M. D. Haiman. A graded representation model for macdonald's polynomials. *Proc. Natl. Acad. Sci. USA*, 90(8):3607–3610, 1993.

[34] I. M. Gessel and S. Seo. A refinement of Cayley's formula for trees. *Electron. J. Combin.*, 11(2):Research Paper 27, 23, 2004/06.

[35] K. P. Hadaway. On combinatorical problems of generalized parking functions. Williams College, Honors Thesis, 2022.

[36] J. Haglund. The q, t-catalan numbers and the space of diagonal harmonics. *University lecture series*, 41, 2008.

[37] J. Haglund, M. D. Haiman, N. Loehr, J. Remmel, and A. Ulyanov. A combinatorial formula for the character of the diagonal coinvariants. *Duke Math. J.*, 126:10, 11 2003.

[38] M. D. Haiman. Conjectures on the quotient ring by diagonal invariants. *J. Algebraic Combin.*, 3:17–76, 1994.

[39] M. D. Haiman. Hilbert schemes, polygraphs and the macdonald positivity conjecture. *J. Amer. Math. Soc.*, 14(4):941–1006, 2001.

[40] M. D. Haiman. Vanishing theorems and character formulas for the hilbert scheme of points in the plane. *Invent. Math.*, 149:371–407, 2002.

[41] P. E. Harris, J. Kretschmann, and J. C. Martínez Mori. Lucky cars and the quicksort algorithm. *arXiv preprint arXiv:2306.13065 (To appear in The American Mathematical Monthly)*, pages 1–8, 2023.

[42] C. A. R. Hoare. Algorithm 64: quicksort. *Comm. Assoc. Comp. Mach.*, 4(7):321, 1961.

[43] OEIS Foundation Inc. The on-line encyclopedia of integer sequences: Entry a288964. `oeis.org/A288964`, 2017.

[44] S. B. Iyengar, G. J. Leuschke, A. Leykin, C. Miller, E. Miller, A. K. Singh, and U. Walther. *Twenty-four hours of local cohomology*, volume 87 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2007.

[45] C. B Jones and J. Misra. *Theories of programming: the life and works of Tony Hoare.* ACM, 2021.

[46] J. Kleinberg and E. Tardos. *Algorithm design.* Pearson Education India, 2006.

[47] J. Kline. Properties of the d-dimensional earth mover's problem. *Discrete Appl. Math.*, 265:128–141, 2019.

[48] D. Knuth. *The art of computer programming*, volume 3. Pearson Education, 1997.

[49] D. E. Knuth. Permutations, matrices, and generalized Young tableaux. *Pacific J. Math.*, 34:709–727, 1970.

[50] P. T. Komiske, E. M. Metodiev, and J. Thaler. Metric space of collider events. *Physical review letters*, 123(4):041801, 2019.

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

[52] J. Kretschmann. Earth mover's distance between grade distribution data with fixed mean. Master's thesis, The University of Wisconsin-Milwaukee, 2020.

[53] I. G. Macdonald. *Symmetric functions and Hall polynomials*. Oxford university press, 1998.

[54] P. A. MacMahon. *Combinatory analysis*. Chelsea Publishing Co., New York, 1960. Two volumes (bound as one).

[55] A. Melnyk, S. Knyazev, F. Vannberg, L. Bunimovich, P. Skums, and A. Zelikovsky. Using earth mover's distance for viral outbreak investigations. *BMC genomics*, 21(5):1–9, 2020.

[56] G. Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pages 666–704, 1781.

[57] M. Morales. Segre embeddings, Hilbert series and Newcomb's problem, 2014. arXiv:1306.6910.

[58] M. Petkovšek, H. S. Wilf, and D. Zeilberger. $A = B$. A.K. Peters, Ltd., Wellesley, MA, 1996. With a foreword by Donald E. Knuth.

[59] J. Riordan. Ballots and trees. *J. Combin. Theory*, 6(4):408–411, 1969.

[60] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40:99–121, 2000.

[61] R. Rudolf and G. J. Woeginger. The cone of monge matrices: extremal rays and applications. *Zeitschrift für Operations Research*, 42:161–168, 1995.

[62] scipy.optimize.linprog — scipy v1.7.1 manual. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html`. Accessed: 2023-09-28.

[63] R. Sedgewick. Implementing quicksort programs. *Comm. Assoc. Comp. Mach.*, 21(10):847–857, 1978.

[64] R. Sedgewick. *Algorithms*. Pearson Education India, 1988.

[65] R. P. Stanley. `https://mathoverflow.net/questions/253773/number-of-boolean-algebra-subintervals-in-weak-order-of-s-n/253781#253781`. Online, accessed 06-12-2023.

[66] R. P. Stanley. Hyperplane arrangements, parking functions and tree inversions. In *Mathematical essays in honor of Gian-Carlo Rota (Cambridge, MA, 1996)*, volume 161 of *Progr. Math.*, pages 359–375. Birkhäuser Boston, Boston, MA, 1998.

[67] R. P. Stanley. Parking functions. `https://math.mit.edu/~rstan/transparencies/parking.pdf`, 1999.

[68] R. P. Stanley. *Enumerative combinatorics. Volume 1*, volume 49 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, second edition, 2012.

[69] R. P. Stanley. *Catalan numbers*. Cambridge University Press, 2015.

[70] R. P. Stanley. *Algebraic combinatorics: walks, trees, tableaux, and more*. Undergraduate Texts in Mathematics. Springer, Cham, 2018. 2nd ed.

[71] B. E. Tenner. Reduced decompositions and permutation patterns. *J. Algebraic Combin.*, 24, 07 2005.

[72] B. E. Tenner. Pattern avoidance and the bruhat order. *J. Combin. Theory Ser. A*, 114(5):888–905, 2007.

[73] B. E. Tenner. Interval structures in the Bruhat and weak orders. *J. Comb.*, 13(1):135 – 165, 2022.

[74] S. van Willigenburg. The shuffle conjecture. *Bulletin of the American Mathematical Society*, 57(1):77–89, 2020.

[75] C. H. Yan. Parking functions. In *Handbook of enumerative combinatorics*, Discrete Math. Appl. (Boca Raton), pages 835–893. CRC Press, Boca Raton, FL, 2015.

[76] A. Yasmin, A. Dylan, P. E. Harris, S. Kotapati, J. C. Martínez Mori, C. D. Monroe, Z. Saylor, C. Tieu, and D. A. Williams. On parking functions and the tower of hanoi. *Amer. Math. Monthly*, 0(0):1–7, 2023.