Parking Functions of Fixed Displacement

Lucas Chaves Meyles, Richter Jordaan, Sam Sehayek, Ethan Spingarn

Summer@ICERM

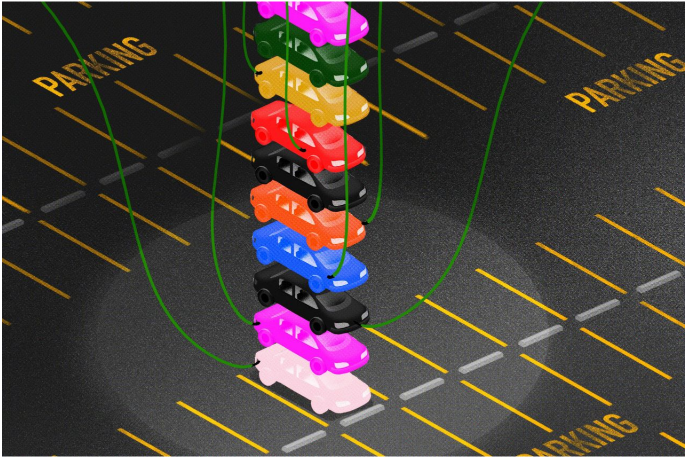
displacement@gmail.com

August 3, 2022


SOME MEN
JUST WANT TO WATCH THE WORLD BURN

Overview

Part I: Preliminaries



Displacement Preliminaries

The **displacement vector** for a parking function $a = (a_1, \ldots, a_n)$ is an n-tuple $(d_1, \ldots, d_n)$ where d_i is a nonnegative integer denoting the number of slots, starting with its preference, that car i passes over before it parks.

Displacement Preliminaries

The **displacement vector** for a parking function $a = (a_1, \ldots, a_n)$ is an n-tuple $(d_1, \ldots, d_n)$ where d_i is a nonnegative integer denoting the number of slots, starting with its preference, that car i passes over before it parks.

The parking function $(4, 1, 1, 3, 2)$ has displacement vector $(0, 0, 1, 0, 3)$.

Displacement Partition

The **displacement partition** is the integer partition comprising $d_1, d_2, \ldots, d_n$ rearranged into nonincreasing order with zeros removed.

Displacement Partition

The **displacement partition** is the integer partition comprising $d_1, d_2, \ldots, d_n$ rearranged into nonincreasing order with zeros removed.

The displacement partition of $(4, 1, 1, 3, 2)$ is $3 + 1$.

Partition-Preserving Parking Functions

What is the closest thing to increasing order that preserves its displacement partition?

Partition-Preserving Parking Functions

What is the closest thing to increasing order that preserves its displacement partition?

Partition-Preserving Parking Functions

A parking function $\alpha = (a_1, \ldots, a_n)$ is **partition-preserving** if the ith car parks in spot i.

Partition-Preserving Parking Functions

What is the closest thing to increasing order that preserves its displacement partition?

Partition-Preserving Parking Functions

A parking function $\alpha = (a_1, \ldots, a_n)$ is **partition-preserving** if the ith car parks in spot i.

A parking function $\alpha = (a_1, \ldots, a_n)$ is partition-preserving if and only if $a_i \leq i$.

Partition-Preserving Order

$$\alpha = (3, 1, 2, 3, 2)$$

Let $s = (3, 1, 2, 4, 5)$ be the permutation of $[n]$ which records where each car in β parks. Then,

$$\begin{pmatrix} 3 & 1 & 2 & 4 & 5 \\ 3 & 1 & 2 & 3 & 2 \end{pmatrix} \xrightarrow{\sigma} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 3 & 2 \end{pmatrix} = \beta'.$$

The preference of the car which parked at s_i gets moved to s_i.

Properties of Partition-Preserving

Theorem

Placing a parking function α into partition-preserving order α' does not alter the displacement partition λ of α.

Proof.

1. The parts of a displacement partition are all the differences $s_i - a_i$ with 0's removed.

$\square$

Properties of Partition-Preserving

Theorem

Placing a parking function α into partition-preserving order α' does not alter the displacement partition λ of α.

Proof.

1. The parts of a displacement partition are all the differences $s_i - a_i$ with 0's removed.
2. Because σ in the partition-preserving definition keeps s_i and a_i in the same column for all $i \in [n]$, $s_i - a_i = s_{\sigma(i)} - a_{\sigma(i)}$ for all $i \in [n]$.

$\square$

Properties of Partition-Preserving

Theorem

Placing a parking function α into partition-preserving order α' does not alter the displacement partition λ of α.

Proof.

1. The parts of a displacement partition are all the differences $s_i - a_i$ with 0's removed.
2. Because σ in the partition-preserving definition keeps s_i and a_i in the same column for all $i \in [n]$, $s_i - a_i = s_{\sigma(i)} - a_{\sigma(i)}$ for all $i \in [n]$.
3. This shows that α' has the same displacement partition as α.

$\square$

Partition-Preserving Examples

$$(3, 1, 2, 3, 2)$$
$$\lambda = 3 + 1$$

Partition-Preserving Examples

$$\begin{array}{c} (3,1,2,3,2) \\ \lambda = 3+1 \end{array} \xrightarrow{\sigma} \begin{array}{c} (1,2,3,3,2) \\ \lambda = 3+1 \end{array}$$

Partition-Preserving Examples

$$\begin{matrix}(3,1,2,3,2) \\ \lambda = 3+1\end{matrix} \xrightarrow{\sigma} \begin{matrix}(1,2,3,3,2) \\ \lambda = 3+1\end{matrix}$$

$$\begin{matrix}(4,1,4,1,4,1) \\ \lambda = 2+2+1+1\end{matrix}$$

Partition-Preserving Examples

$$\begin{matrix} (3,1,2,3,2) \\ \lambda = 3+1 \end{matrix} \xrightarrow{\sigma} \begin{matrix} (1,2,3,3,2) \\ \lambda = 3+1 \end{matrix}$$

$$\begin{matrix} (4,1,4,1,4,1) \\ \lambda = 2+2+1+1 \end{matrix} \xrightarrow{\sigma} \begin{matrix} (1,1,1,4,4,4) \\ \lambda = 2+2+1+1 \end{matrix}$$

Partition-Preserving Examples

$$\begin{array}{c}(3,1,2,3,2)\\ \lambda = 3+1\end{array} \xrightarrow{\sigma} \begin{array}{c}(1,2,3,3,2)\\ \lambda = 3+1\end{array}$$

$$\begin{array}{c}(4,1,4,1,4,1)\\ \lambda = 2+2+1+1\end{array} \xrightarrow{\sigma} \begin{array}{c}(1,1,1,4,4,4)\\ \lambda = 2+2+1+1\end{array}$$

$$\begin{array}{c}(2,3,2,6,1,1)\\ \lambda = 4+2\end{array}$$

Partition-Preserving Examples

$$\begin{array}{c} (3,1,2,3,2) \\ \lambda = 3+1 \end{array} \xrightarrow{\sigma} \begin{array}{c} (1,2,3,3,2) \\ \lambda = 3+1 \end{array}$$

$$\begin{array}{c} (4,1,4,1,4,1) \\ \lambda = 2+2+1+1 \end{array} \xrightarrow{\sigma} \begin{array}{c} (1,1,1,4,4,4) \\ \lambda = 2+2+1+1 \end{array}$$

$$\begin{array}{c} (2,3,2,6,1,1) \\ \lambda = 4+2 \end{array} \xrightarrow{\sigma} \begin{array}{c} (1,2,3,2,1,6) \\ \lambda = 4+2 \end{array}$$

Partition-Preserving Equivalence Classes

Theorem

Let α and β be two parking functions of length n. $\alpha \sim \beta$ if α and β have the same partition-preserving order is an equivalence relation on all parking functions of length n.

Partition-Preserving Equivalence Classes

111	112	113	121	122	123
		131	211	212	132
		311		221	213
					231
					312
					321

- The representative of each equivalence class is the one partition-preserving parking function.
- Note: Two parking functions can have the same preferences and displacement partition but be in different equivalence classes. For example, 11322 and 12213.

Prime Decomposition

We can uniquely partition any parking function into prime parking functions after placing it in partition-preserving order
$\alpha = (2, 3, 2, 6, 1, 1)$

1. Place α into partition-preserving order α'.
   $\alpha = (2, 3, 2, 6, 1, 1) \mapsto (1, 2, 3, 2, 1, 6)$.

Prime Decomposition

We can uniquely partition any parking function into prime parking functions after placing it in partition-preserving order
$\alpha = (2, 3, 2, 6, 1, 1)$

1. Place α into partition-preserving order α'.
   $\alpha = (2, 3, 2, 6, 1, 1) \mapsto (1, 2, 3, 2, 1, 6)$.
2. Find the indices where α' satisfies these two criteria:
   1. $a_i = i$
   2. For all $j > i$, $a_j \geq a_i$.

   (1, 2, 3, 2, 1 , 6)

Prime Decomposition

We can uniquely partition any parking function into prime parking functions after placing it in partition-preserving order

$\alpha = (2, 3, 2, 6, 1, 1)$

1. Place α into partition-preserving order α'.
   $\alpha = (2, 3, 2, 6, 1, 1) \mapsto (1, 2, 3, 2, 1, 6)$.
2. Find the indices where α' satisfies these two criteria:
   1. $a_i = i$
   2. For all $j > i$, $a_j \geq a_i$.

   (1, 2, 3, 2, 1 , 6)
3. Reorder α'
   (2, 3, 2 , 6 , 1, 1)

Prime Decomposition

We can uniquely partition any parking function into prime parking functions after placing it in partition-preserving order

$\alpha = (2, 3, 2, 6, 1, 1)$

1. Place α into partition-preserving order α'.
   $\alpha = (2, 3, 2, 6, 1, 1) \mapsto (1, 2, 3, 2, 1, 6)$.
2. Find the indices where α' satisfies these two criteria:
   1. $a_i = i$
   2. For all $j > i$, $a_j \geq a_i$.

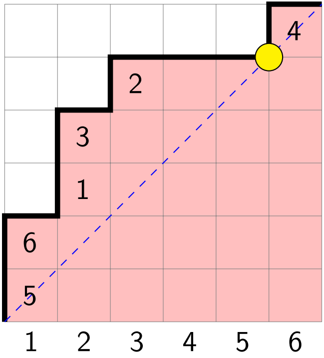
   $(\boxed{1, 2, 3, 2, 1}, \boxed{6})$
3. Reorder α'
   $(\boxed{2, 3, 2}, \boxed{6}, \boxed{1, 1})$

Key Insight

Displacement *always* occurs within the component prime parking functions!

Prime Decomposition Visualization

$$\alpha = (2, 3, 2, 6, 1, 1)$$



Prime Decomposition Example

$$(5, 3, 3, 1, 6, 5, 1)$$

$$\downarrow$$

$$(1, 1, 3, 3, 5, 6, 5)$$

Prime Decomposition Example

$$(5, 3, 3, 1, 6, 5, 1)$$

$$\downarrow$$

$$(1,\ 1,\ 3,\ 3,\ 5,\ 6,\ 5)$$

Prime Decomposition Example

$$(5, 3, 3, 1, 6, 5, 1)$$

$$\downarrow$$

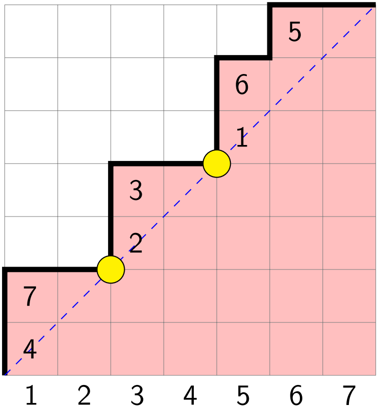
$$(1, 1, 3, 3, 5, 6, 5)$$

$$\downarrow$$

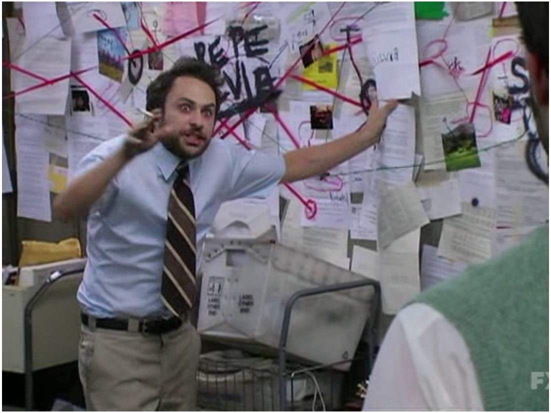
$$(5, 3, 3, 1, 6, 5, 1)$$

Prime Decomposition Visualization

$$\alpha = (\boxed{5}, \boxed{3,\ 3}, \boxed{1}, \boxed{6,\ 5}, \boxed{1})$$



Part II: Direct Enumerations



when the theorem statement has so many cases that it's 2 pages long

Direct Enumeration

Original motivating question: how many parking functions of length n with a fixed displacement partition are there?

Direct Enumeration

Original motivating question: how many parking functions of length n with a fixed displacement partition are there?

Our approach:

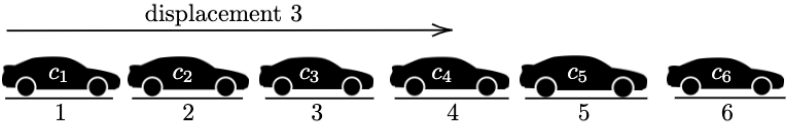
- understand what kinds of patterns give rise to the displacement partition
- count the ways to rearrange terms inside and outside of the pattern while preserving the displacement partition

1 Displaced Car

Suppose a single car is displaced by $k > 0$ slots.

1 Displaced Car

Suppose a single car is displaced by $k > 0$ slots.
Example: The parking function $(1, 2, 3, 1, 5, 6)$.



The parking function has length $n = 6$ and a single car c_4 displaced by $k = 3$ slots.

1 Displaced Car

In general, what does this kind of parking function look like?

1 Displaced Car

In general, what does this kind of parking function look like?

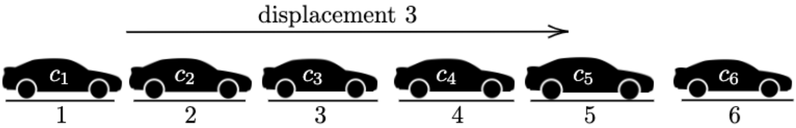
Theorem (CM, J, S)

In a parking function with displacement partition $k > 0$, the pattern of the partition-preserving prime that produces the displacement is $(\underbrace{1, 2, \ldots, k}_{\text{k terms}}, 1)$

Given the pattern, how many ways can we rearrange terms while preserving the displacement partition?

1 Displaced Car

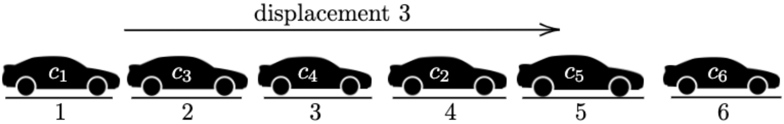
We can choose where to place the pattern inside the parking function.
Example: parking function $(1, 2, 3, 4, 2, 6)$



The starting point of the parking function's pattern shifts from slot 1 to 2.

1 Displaced Car

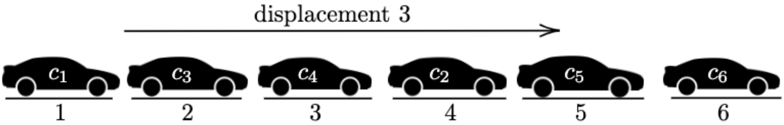
We can permute certain undisplaced cars inside the pattern.
Example: parking function $(1, 4, 2, 3, 2, 6)$



We can permute the preferences of c_2, c_3, and c_4, which provide the "buildup" for the displacement.

1 Displaced Car

We can permute certain undisplaced cars inside the pattern.
Example: parking function $(1, 4, 2, 3, 2, 6)$



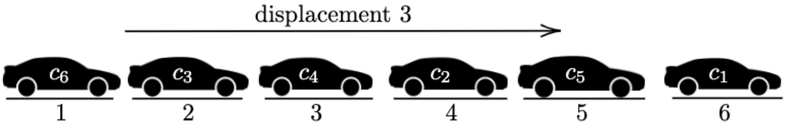
We can permute the preferences of c_2, c_3, and c_4, which provide the "buildup" for the displacement.

Question

Why is the position of the displaced car c_5 fixed?

1 Displaced Car

We can permute entries outside of the pattern.
Example: parking function $(6, 4, 2, 3, 2, 1)$



We permute the preferences of c_1 and c_6.

1 Displaced Car

Theorem (CM, J, S)

The number of parking functions of length n with displacement partition $k > 0$ is

$$\frac{n!}{k+1}(n-k).$$

1 Displaced Car

Corollary

The number of parking functions of length n with one car displaced,

$$\sum_{k=1}^{n-1} \frac{(n-k)n!}{k+1}$$

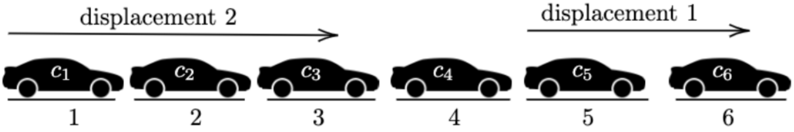
is equal to the Second-order Eulerian number $A(n, n-2)$.

2 Displaced Cars

Now suppose the displacement partition is $k + \ell$, where $1 \leq \ell \leq k$. What does this kind of parking function look like?

2 Displaced Cars

Now suppose the displacement partition is $k + \ell$, where $1 \leq \ell \leq k$.
What does this kind of parking function look like?
Case 1: The displacements happen in disjoint primes.



The parking function $(1, 2, 1, 4, 5, 5)$ has two cars displaced in disjoint primes.

2 Displaced Cars

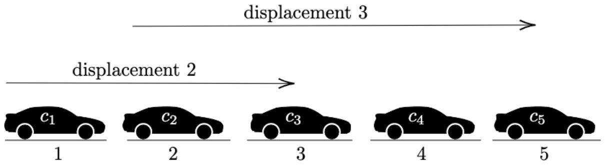
Case 2: The displacements happen in a single prime.

2 Displaced Cars

Case 2: The displacements happen in a single prime.

Displacement Order

The **displacement order** of a parking function with m displaced cars is an m-tuple whose i-th entry is the value of the i-th displacement that arises.



The parking function $(1, 2, 1, 4, 2)$ has displacement order $(2, 3)$. Both of its displacements happen in a single prime.

2 Displaced Cars

As before, we find the pattern for each case and then count the number of rearrangements that preserve the displacement partition.

2 Displaced Cars

As before, we find the pattern for each case and then count the number of rearrangements that preserve the displacement partition.

Theorem (CM, J, S)

The number of parking functions of length n with displacement partition $k+\ell$ where $1 \leq \ell < k$ is

$$\frac{n!(n-k)(k-\ell)}{(k+1)(\ell+1)} + \sum_{i=2}^{\ell+1} \frac{n!(k+i-1)!(n-k-i+1)}{(\ell+1)(k+i)!} + \sum_{i=1}^{\ell} \frac{n!(k+i)!(n-k-i)}{(k+i+1)!(k+1)}$$
$$+2k!\ell!\binom{n-(k+\ell)}{2}\binom{n}{k+1}\binom{n-(k+1)}{\ell+1}(n-(k+\ell+2))!$$

for $n \geq k+\ell+2$.

2 Displaced Cars

As before, we find the pattern for each case and then count the number of rearrangements that preserve the displacement partition.

Theorem (CM, J, S)

The number of parking functions of length n with displacement partition $k+\ell$ where $1 \leq \ell < k$ is

$$\frac{n!(n-k)(k-\ell)}{(k+1)(\ell+1)} + \sum_{i=2}^{\ell+1} \frac{n!(k+i-1)!(n-k-i+1)}{(\ell+1)(k+i)!} + \sum_{i=1}^{\ell} \frac{n!(k+i)!(n-k-i)}{(k+i+1)!(k+1)}$$
$$+2k!\ell!\binom{n-(k+\ell)}{2}\binom{n}{k+1}\binom{n-(k+1)}{\ell+1}(n-(k+\ell+2))!$$

for $n \geq k+\ell+2$.

be honest, you did not read that formula

3 Displaced Cars

Let's now only consider *prime* parking functions. Suppose the displacement partition of a prime parking function is $k + \ell + m$ where $1 \leq m \leq \ell \leq k$.
What does this kind of parking function look like?

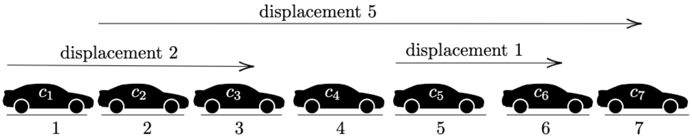
3 Displaced Cars

Let's now only consider *prime* parking functions. Suppose the displacement partition of a prime parking function is $k+\ell+m$ where $1 \leq m \leq \ell \leq k$.

What does this kind of parking function look like?

Observation

When more than 2 cars are displaced in a prime parking function, it's possible that not all of the displacements overlap.



For the prime parking function $(1,2,3,1,4,5,5,2)$, the second displacement does not overlap the first.

3 Displaced Cars

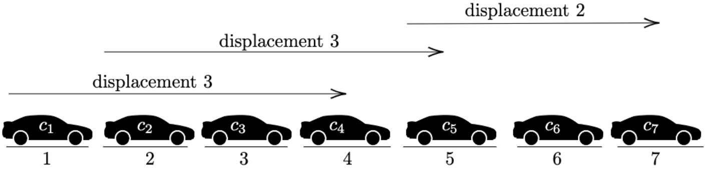
Definition

Let a prime parking function where each displacement overlaps the previous one be called **strongly grouped**.

3 Displaced Cars

Definition

Let a prime parking function where each displacement overlaps the previous one be called **strongly grouped**.



The prime parking function $(1, 2, 3, 1, 2, 6, 5)$ is strongly grouped.

3 Displaced Cars

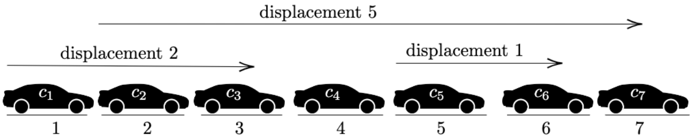
Definition

If a prime parking function is not strongly grouped, call it **weakly grouped**.

3 Displaced Cars

Definition

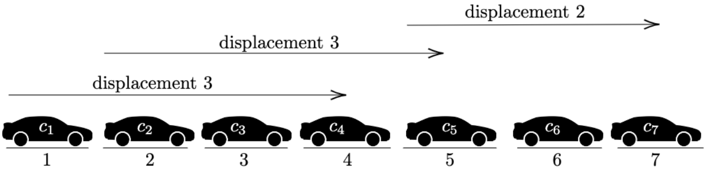
If a prime parking function is not strongly grouped, call it **weakly grouped**.



The prime parking function $(1, 2, 3, 1, 4, 5, 5, 2)$ is weakly grouped.
The second displacement does not overlap the first.

3 Displaced Cars

Strongly grouped primes with 3 displaced cars have patterns similar to those for 2 displaced cars, with one added subpattern for the last displacement.



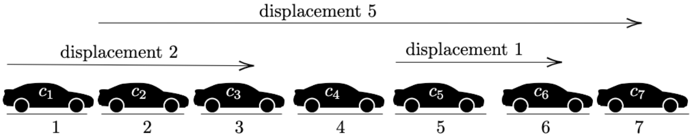
The first two displacements form a strongly grouped prime parking function. The third displacement can overlap anywhere.

3 Displaced Cars

Can we connect weakly grouped primes with 3 displaced cars to known results for 2 displaced cars?

3 Displaced Cars

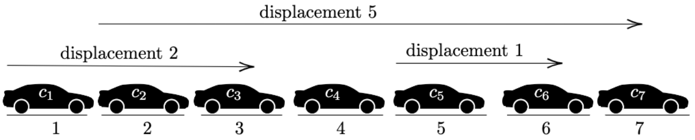
Can we connect weakly grouped primes with 3 displaced cars to known results for 2 displaced cars?



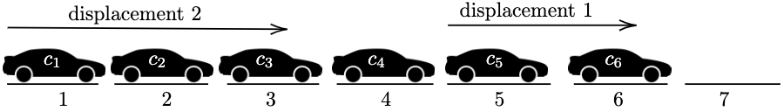
Let's see what the parking function looks like before the last car parks.

3 Displaced Cars

Can we connect weakly grouped primes with 3 displaced cars to known results for 2 displaced cars?



Let's see what the parking function looks like before the last car parks.



3 Displaced Cars

the greatest discovery in the history of math (CM, J, S)

Removing the last entry of a weakly grouped prime parking functions with 3 displaced cars gives a parking function with 2 displaced cars parked in disjoint primes!

3 Displaced Cars

the greatest discovery in the history of math (CM, J, S)

Removing the last entry of a weakly grouped prime parking functions with 3 displaced cars gives a parking function with 2 displaced cars parked in disjoint primes!

This allowed us compute the number of prime parking functions of length n for displacement partition $k + \ell + m$ for $1 \leq \ell \leq k$.

Beyond 3 Displaced Cars

We want to recursively count the number of prime parking functions with $m > 2$ displaced cars.

- We found a recursive formula for the number of strongly grouped prime parking functions with m displaced cars.
- The number of weakly grouped prime parking functions reduces to an enumerative formula for $m - 1$ displaced cars.

Beyond 3 Displaced Cars

We want to recursively count the number of prime parking functions with $m > 2$ displaced cars.

- We found a recursive formula for the number of strongly grouped prime parking functions with m displaced cars.
- The number of weakly grouped prime parking functions reduces to an enumerative formula for $m - 1$ displaced cars.

big slay (CM,J,S)

We can recursively build up the enumeration of prime parking functions with any number of displaced cars!

However, the formulas sum over all displacement orders, so they can get quite clunky.

Part III: The Algorithm

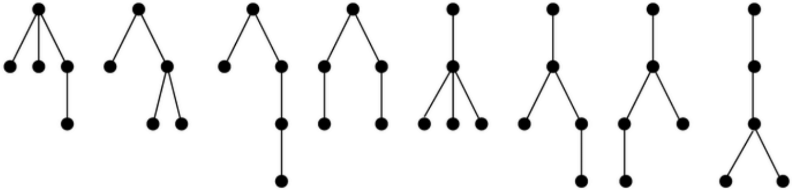


Photo Credit: Joe Sawada

Displacement Vectors Revisited

In General

DV : {Parking Functions} $\to$ {Displacement Vectors}, $pf \mapsto v$ is a many-to-one surjection.

Displacement Vectors Revisited

In General

DV : {Parking Functions} $\to$ {Displacement Vectors}, $pf \mapsto v$ is a many-to-one surjection.

How many displacement vectors of length n?
Hint: ith entry is between 0 and $i-1$.

Displacement Vectors Revisited

In General

DV : {Parking Functions} $\to$ {Displacement Vectors}, $pf \mapsto v$ is a many-to-one surjection.

How many displacement vectors of length n?
Hint: ith entry is between 0 and $i-1$.

How many partition preservers of length n?
Recall definition: ith entry is between 1 and i.

Displacement Vectors Revisited

In General

DV : {Parking Functions} $\to$ {Displacement Vectors}, $pf \mapsto v$ is a many-to-one surjection.

How many displacement vectors of length n?
Hint: ith entry is between 0 and $i-1$.

How many partition preservers of length n?
Recall definition: ith entry is between 1 and i.

Restricted to Partition Preserving

DV : {Partition Preserving Parking Functions} $\to$ {Displacement Vectors}, $pf \mapsto v$ is a bijection.

How to Count: a Brief Tutorial

In Displacement Vector Land

$$n = 6,\ \lambda = 4 + 2 + 1 + 1$$

How to Count: a Brief Tutorial

In Displacement Vector Land

$$n = 6,\ \lambda = 4 + 2 + 1 + 1$$

4 can only go in the 5th and 6th indices:

$$(\_, \_, \_, \_, 4, \_) \quad (\_, \_, \_, \_, \_, 4)$$

How to Count: a Brief Tutorial

In Displacement Vector Land

$$n = 6,\ \lambda = 4 + 2 + 1 + 1$$

2 can only go in the 3rd, 4th, 5th, and 6th indices:

$$\begin{array}{ccc} (\_,\_,2,\_,4,\_) & (\_,\_,\_,2,4,\_) & (\_,\_,\_,\_,4,2) \\ (\_,\_,2,\_,\_,4) & (\_,\_,\_,2,\_,4) & (\_,\_,\_,\_,2,4) \end{array}$$

How to Count: a Brief Tutorial

In Displacement Vector Land

$$n = 6,\ \lambda = 4 + 2 + 1 + 1$$

1 can go anywhere except the first index and fill in the rest with 0's:

$$\begin{array}{lll}
(0,1,2,1,4,0) & (0,1,1,2,4,0) & (0,1,1,0,4,2) \\
(0,1,2,1,0,4) & (0,1,1,2,0,4) & (0,1,1,0,2,4) \\
(0,1,2,0,4,1) & (0,1,0,2,4,1) & (0,1,0,1,4,2) \\
(0,1,2,0,1,4) & (0,1,0,2,1,4) & (0,1,0,1,2,4) \\
(0,0,2,1,4,1) & (0,0,1,2,4,1) & (0,0,1,1,4,2) \\
(0,0,2,1,1,4) & (0,0,1,2,1,4) & (0,0,1,1,2,4)
\end{array}$$

How to Count: a Brief Tutorial Part 2 the Squeakquel

How many parking functions have this partition preserving order?

$$(1,2,1,4,2)$$
$$d(1,2,1,4,2)=(0,0,2,0,3)$$

Which preferences must come before which other preferences?

How to Count: a Brief Tutorial Part 2 the Squeakquel

How many parking functions have this partition preserving order?

$$(1, 2, 1, 4, 2)$$
$$d(1, 2, 1, 4, 2) = (0, 0, 2, 0, 3)$$

Which preferences must come before which other preferences?

$$\begin{array}{cccc} 2 & 2 & 2 & 2 \\ | & | & | & | \\ 1 & 2 & 1 & 4 \end{array}$$

How to Count: a Brief Tutorial Part 2 the Squeakquel

How many parking functions have this partition preserving order?

$$(1, 2, 1, 4, 2)$$
$$d(1, 2, 1, 4, 2) = (0, 0, 2, 0, 3)$$

Which preferences must come before which other preferences?

$$\begin{array}{cccc} 2 & 2 & 2 & 2 \\ | & | & | & | \\ 1 & 2 & 1 & 4 \end{array} \qquad \begin{array}{cc} 1 & 1 \\ | & | \\ 1 & 2 \end{array}$$

How to Count: a Brief Tutorial Part 2 the Squeakquel

How many parking functions have this partition preserving order?

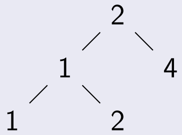
$$(1, 2, 1, 4, 2)$$
$$d(1, 2, 1, 4, 2) = (0, 0, 2, 0, 3)$$

Which preferences must come before which other preferences?

$$\begin{array}{cccccccc} 2 & 2 & 2 & 2 & \quad & \quad & 1 & 1 \\ | & | & | & | & & & | & | \\ 1 & 2 & 1 & 4 & & & 1 & 2 \end{array}$$

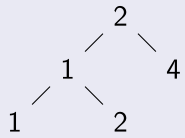
This is a partial ordering on the set of preferences.

Call in lattice 007 because we have a wild poset on the loose



This poset encodes all of the necessary orderings amongst preferences to maintain the displacement partition.

Call in lattice 007 because we have a wild poset on the loose



The linear extensions of this poset are

2	2	2	2	2	2	2	2
4	1	1	1	4	1	1	1
1	4	2	2	1	4	1	1
2	2	4	1	1	1	4	2
1	1	1	4	2	2	2	4

A Characterization of Parking Functions of Fixed Displacement

Partition preserving order partitions parking functions and the equivalence class representatives are partition preserving parking functions.

A Characterization of Parking Functions of Fixed Displacement

Partition preserving order partitions parking functions and the equivalence class representatives are partition preserving parking functions.

Partition preserving parking functions are the same as displacement vectors.

A Characterization of Parking Functions of Fixed Displacement

Partition preserving order partitions parking functions and the equivalence class representatives are partition preserving parking functions.

Partition preserving parking functions are the same as displacement vectors.

We can count displacement vectors. We can also count equivalence class sizes. Bam!

A Neat Formula

$$PF(n, \lambda) = \sum_{v \in V(n,\lambda)} L(v)$$

On efficiency

An Interesting Pattern

	[0]	[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, 1, 1, 1]	[1, 1, 1, 1, 1, 1, 1, 1]
1	1	0	0	0	0	0	0	0	0
2	2	1	0	0	0	0	0	0	0
3	6	6	1	0	0	0	0	0	0
4	24	36	14	1	0	0	0	0	0
5	120	240	150	30	1	0	0	0	0
6	720	1800	1560	540	62	1	0	0	0
7	5040	15120	16800	8400	1806	126	1	0	0
8	40320	141120	191520	126000	40824	5796	254	1	0
9	362880	1451520	2328480	1905120	834120	186480	18150	510	1

An Interesting Pattern

of faces of the order 3 Permutohedron

	[0]	[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, 1, 1, 1]	[1, 1, 1, 1, 1, 1, 1, 1]
1	1	0	0	0	0	0	0	0	0
2	2	1	0	0	0	0	0	0	0
3	6	6	1	0	0	0	0	0	0
4	24	36	14	1	0	0	0	0	0
5	120	240	150	30	1	0	0	0	0
6	720	1800	1560	540	62	1	0	0	0
7	5040	15120	16800	8400	1806	126	1	0	0
8	40320	141120	191520	126000	40824	5796	254	1	0
9	362880	1451520	2328480	1905120	834120	186480	18150	510	1

An Interesting Pattern

of faces of the order 3 Permutohedron

	[0]	[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, 1, 1, 1]	[1, 1, 1, 1, 1, 1, 1, 1]
1	1	0	0	0	0	0	0	0	0
2	2	1	0	0	0	0	0	0	0
3	6	6	1	0	0	0	0	0	0
4	24	36	14	1	0	0	0	0	0
5	120	240	150	30	1	0	0	0	0
6	720	1800	1560	540	62	1	0	0	0
7	5040	15120	16800	8400	1806	126	1	0	0
8	40320	141120	191520	126000	40824	5796	254	1	0
9	362880	1451520	2328480	1905120	834120	186480	18150	510	1

# of faces of the order 4 Permutohedron

Part IV: The Permutohedron

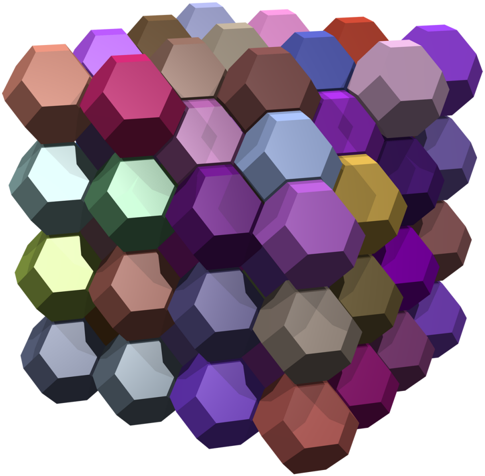


Photo Credit: Andrew Kepert

Unit Interval Parking Functions

A **unit interval parking function** is a parking function of length n where each car is displaced by at most 1 spot.

Unit Interval Parking Functions

A **unit interval parking function** is a parking function of length n where each car is displaced by at most 1 spot.

$$(3, 1, 5, 3, 1, 6)$$

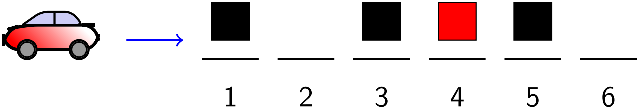
$\longrightarrow$ ___ ___ ___ ___ ___ ___

1 2 3 4 5 6

Unit Interval Parking Functions

A **unit interval parking function** is a parking function of length n where each car is displaced by at most 1 spot.

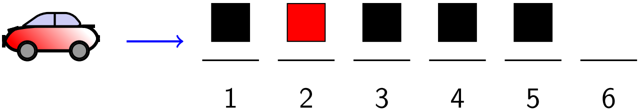
$$(3, 1, 5, 3, 1, 6)$$



Unit Interval Parking Functions

A **unit interval parking function** is a parking function of length n where each car is displaced by at most 1 spot.

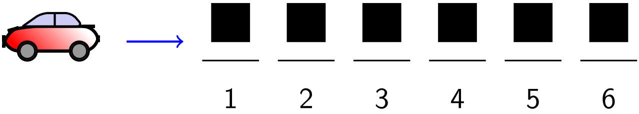
$$(3, 1, 5, 3, 1, 6)$$



Unit Interval Parking Functions

A **unit interval parking function** is a parking function of length n where each car is displaced by at most 1 spot.

$$(3, 1, 5, 3, 1, 6)$$

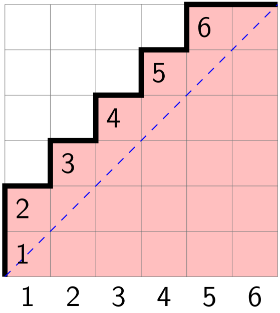


Unique Prime Unit Interval PFs

Theorem

Let α be a unit interval prime parking function of length n. Then, $\alpha = (1, 1, 2, 3, \ldots, n-2, n-1)$.

$$\alpha = (1, 1, 2, 3, 4, 5)$$



Permutohedron Bijection

Theorem (CM, J, S)

Parking functions of length n that displace k cars by 1 are in bijection with the k faces of the permutohedron of order n.

Permutohedron Bijection

Theorem (CM, J, S)

Parking functions of length n that displace k cars by 1 are in bijection with the k faces of the permutohedron of order n.

Vertices of the permutohedron $\Longleftrightarrow$ Parking Functions with 0 displacement

Permutohedron Bijection

Theorem (CM, J, S)

Parking functions of length n that displace k cars by 1 are in bijection with the k faces of the permutohedron of order n.

Vertices of the permutohedron $\Longleftrightarrow$ Parking Functions with 0 displacement

Edges of the Permutohedron $\Longleftrightarrow$ Parking Functions with 1 displacement

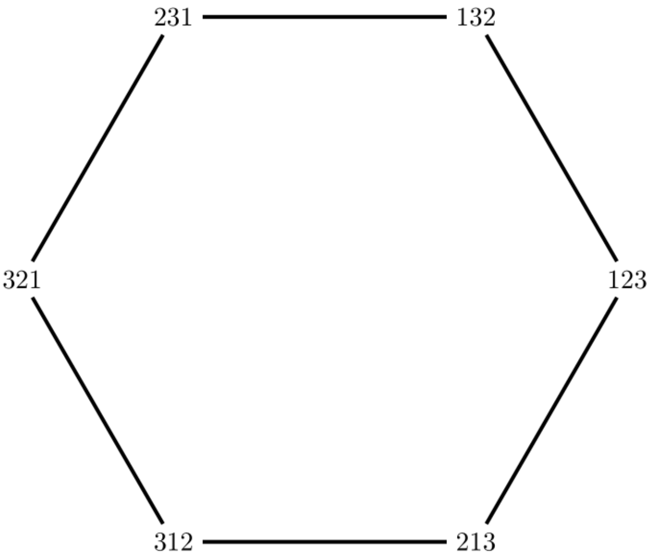
Permutohedron Bijection

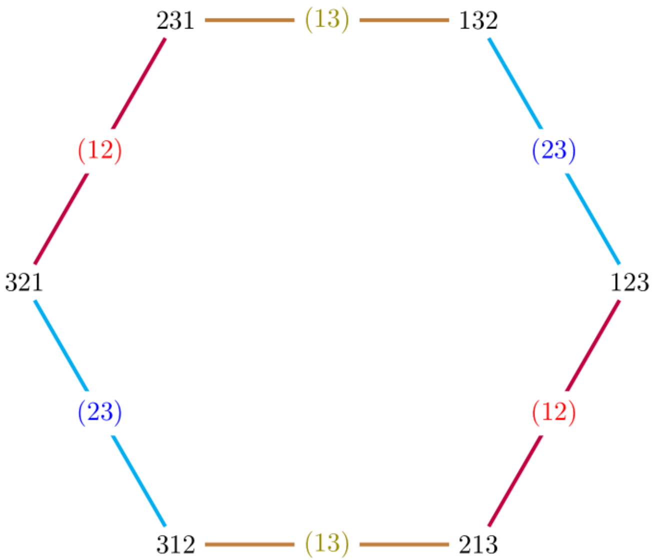
Theorem (CM, J, S)

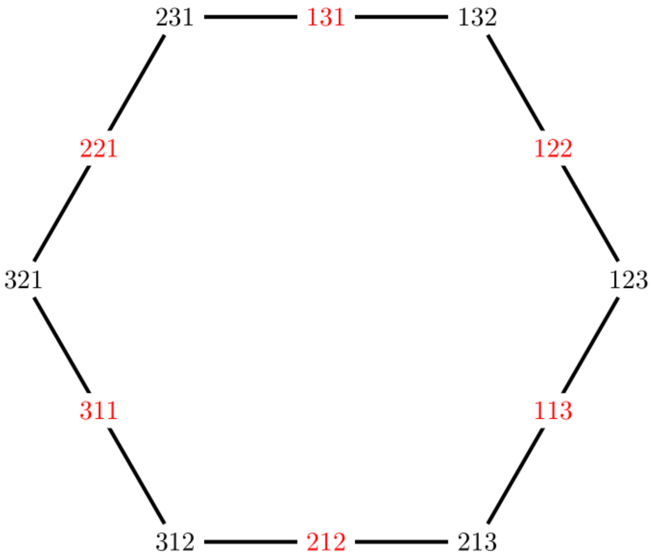
Parking functions of length n that displace k cars by 1 are in bijection with the k faces of the permutohedron of order n.

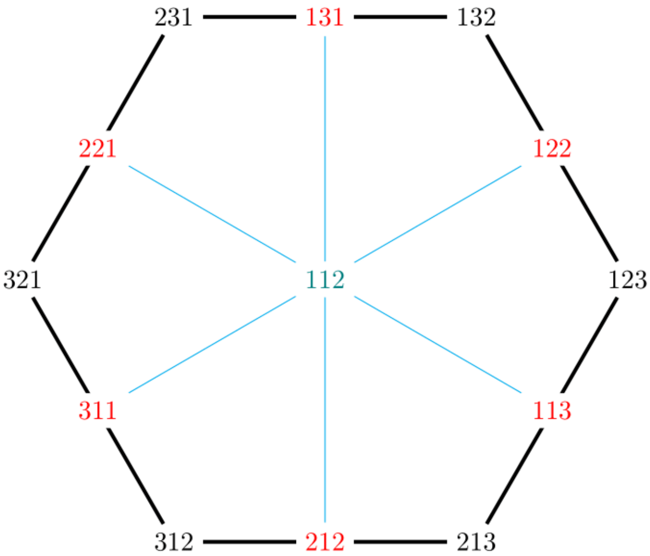
Vertices of the permutohedron $\Longleftrightarrow$ Parking Functions with 0 displacement
Edges of the Permutohedron $\Longleftrightarrow$ Parking Functions with 1 displacement
Faces of the Permutohedron $\Longleftrightarrow$ Parking Functions with $1+1$ displacement

$$\vdots$$


231
132
321
123
312
213


231
(13)
132
(12)
(23)
321
123
(23)
(12)
312
(13)
213


231
131
132
221
122
321
123
311
113
312
212
213

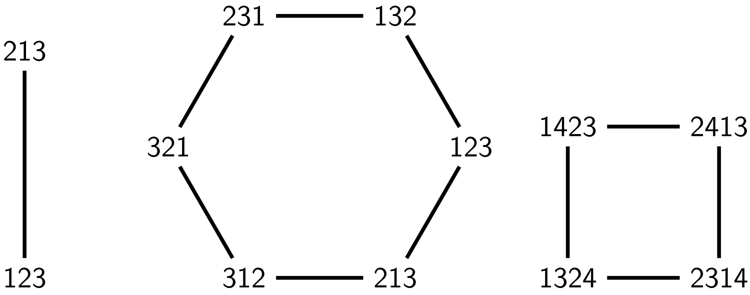

231
131
132
221
122
321
112
123
311
113
312
212
213

Key Insight

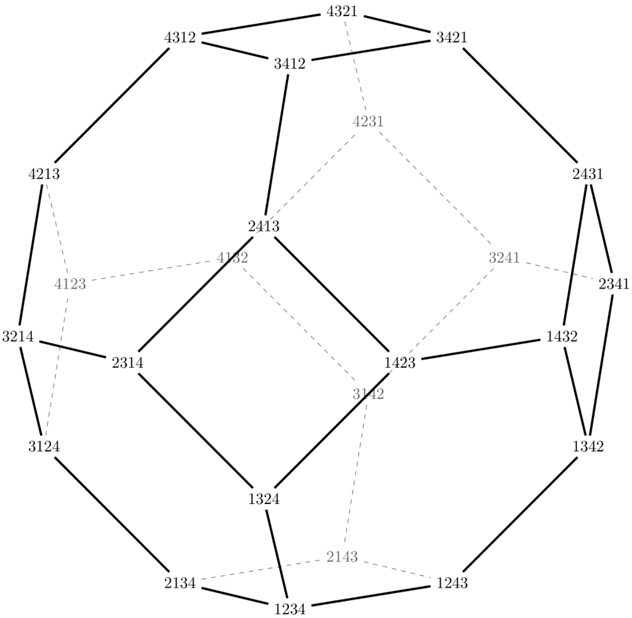
The faces of the permutohedron of order n are composed of products of lower dimensional permutohedra.

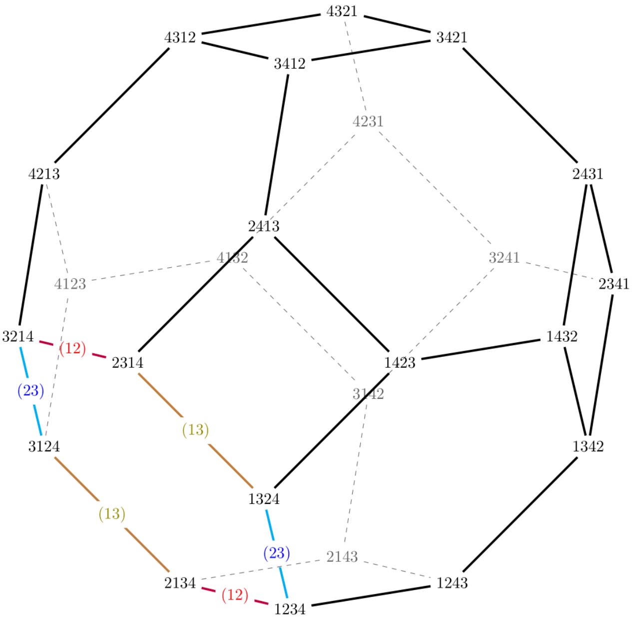
Key Insight

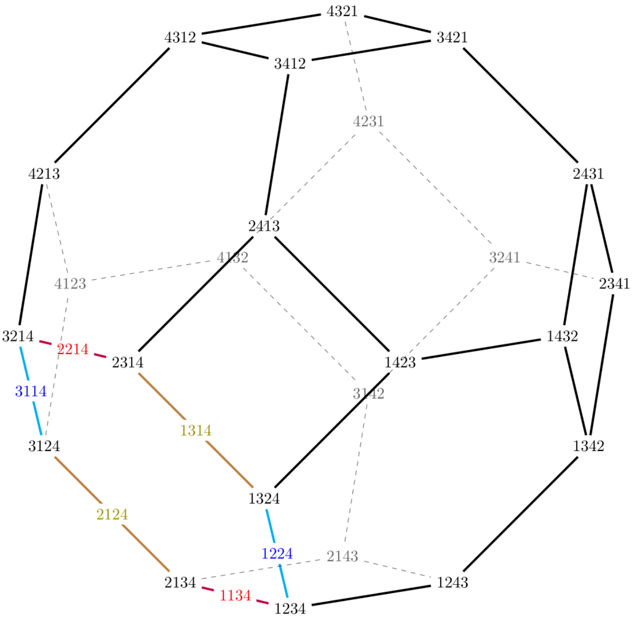
The faces of the permutohedron of order n are composed of products of lower dimensional permutohedra.

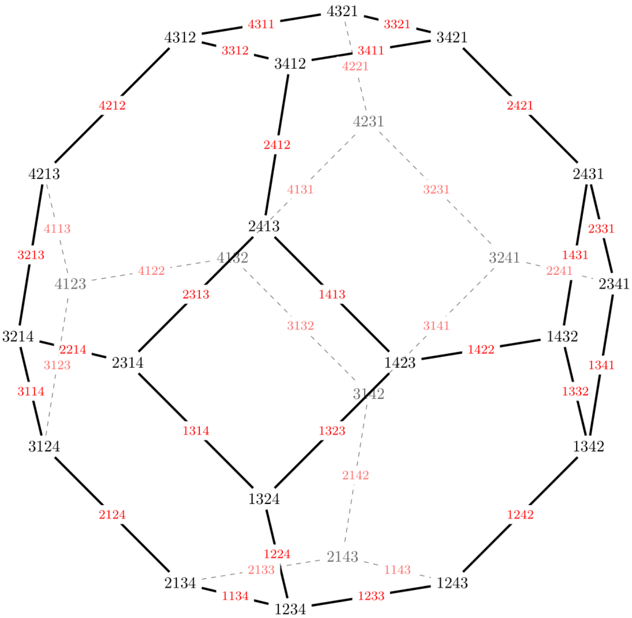


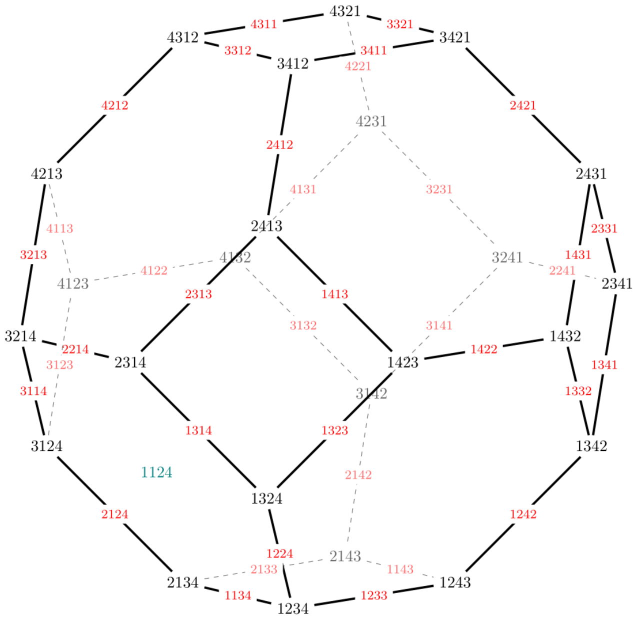
$P(2) \times P(1)$ $P(3)$ $P(2) \times P(2)$

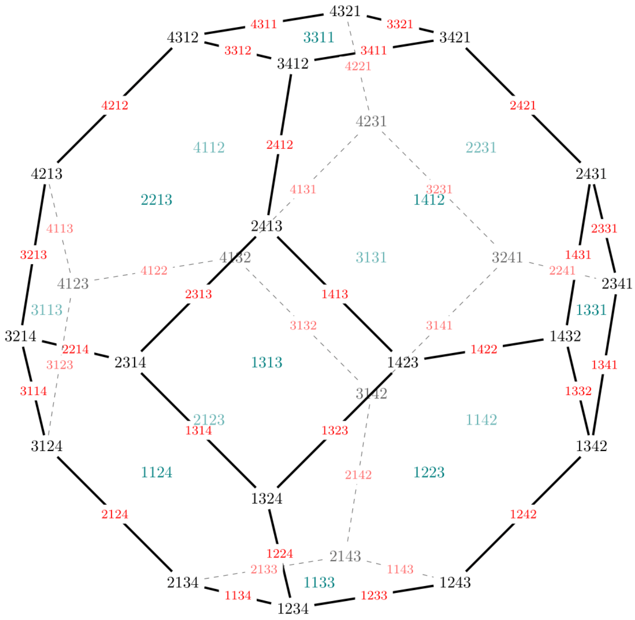

4321
4312
3421
3412
4231
4213
2431
2413
4132
3241
4123
2341
3214
1432
2314
1423
3142
3124
1342
1324
2143
2134
1243
1234

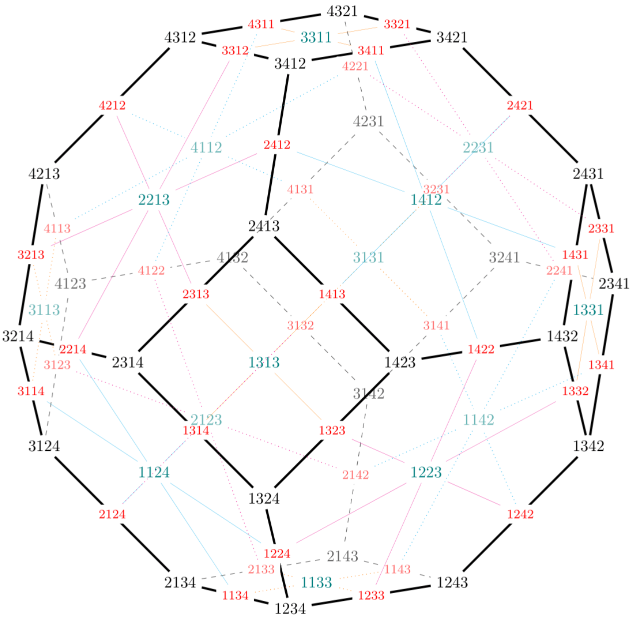

4321
4312
3421
3412
4231
4213
2431
2413
4132
3241
4123
2341
3214
(12)
2314
1432
1423
(23)
3142
(13)
3124
1342
(13)
1324
(23)
2143
2134
1243
(12)
1234

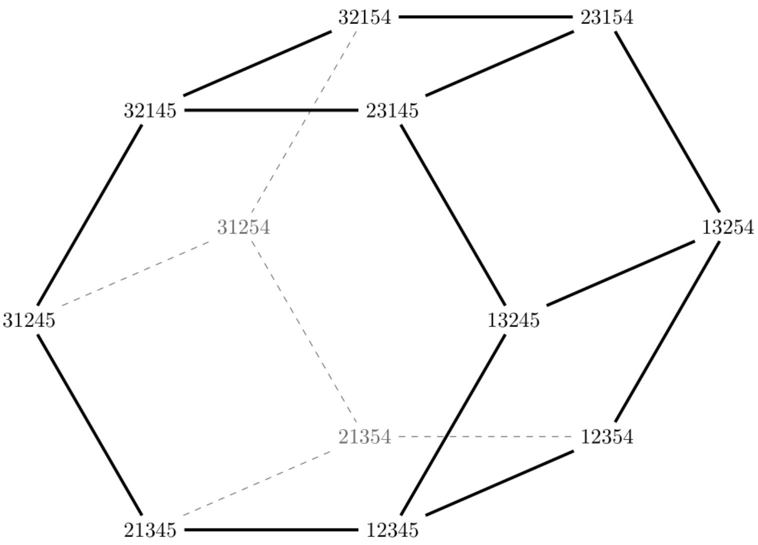

4321
4312
3421
3412
4231
4213
2431
2413
4132
3241
4123
2341
3214
1432
2214
2314
1423
3114
3142
1314
3124
1342
2124
1324
1224
2143
2134
1243
1134
1234

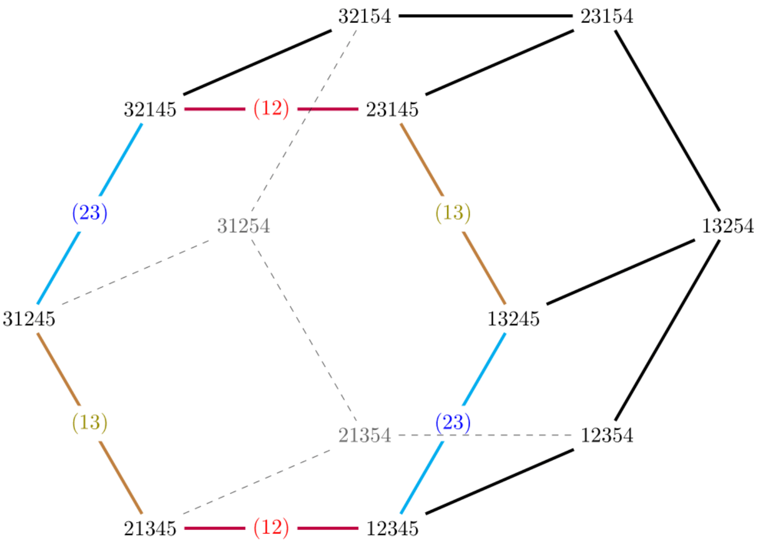

4321
4311
3321
4312
3421
3312
3411
3412
4221
4212
2421
4231
2412
4213
2431
4131
3231
2413
4113
2331
3213
4132
3241
1431
4122
2241
4123
2341
2313
1413
3132
3141
3214
1432
2214
1422
2314
1423
3123
1341
3114
3142
1332
1314
1323
3124
1342
2142
2124
1242
1324
1224
2143
2133
1143
2134
1243
1134
1233
1234

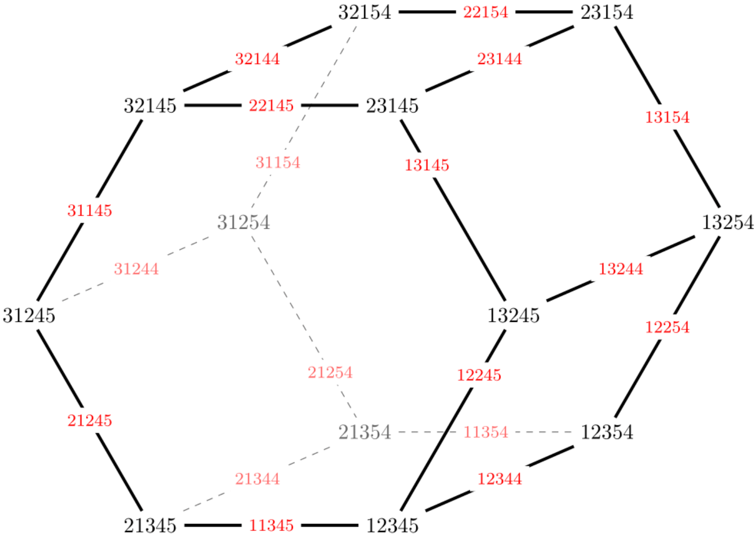

4321
4311
3321
4312
3312
3411
3421
3412
4221
4212
2421
4231
2412
4213
2431
4131
3231
4113
2413
2331
3213
4132
3241
1431
4122
2241
2341
4123
2313
1413
3132
3141
3214
2214
1432
1422
3123
2314
1423
1341
3114
3142
1332
1314
1323
3124
1342
1124
2142
1324
2124
1242
1224
2143
2133
1143
2134
1243
1134
1233
1234

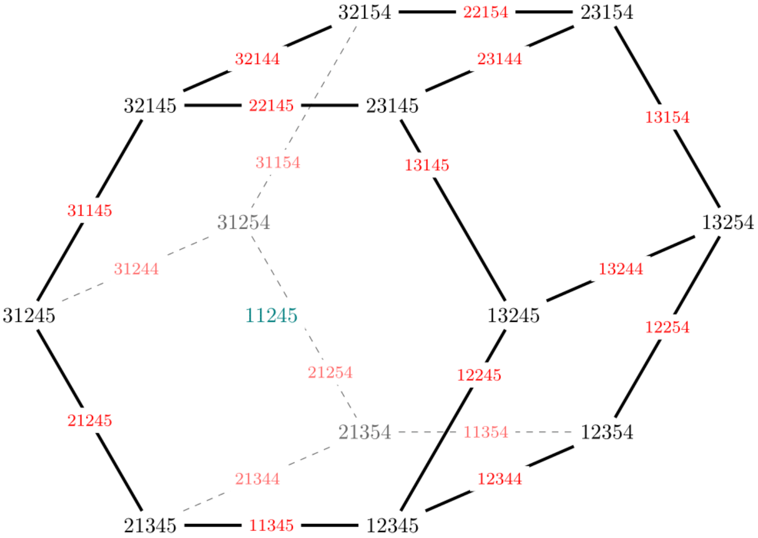

4321
4311
3321
4312
3311
3421
3312
3411
3412
4221
4212
2421
4231
4112
2231
2412
4213
2431
4131
3231
2213
1412
4113
2331
2413
3213
1431
4132
3131
3241
4122
2241
4123
2341
2313
1413
3113
1331
3132
3141
3214
1432
2214
1422
2314
1313
1423
3123
1341
3114
1332
3142
2123
1142
1314
1323
3124
1342
1124
1223
2142
1324
2124
1242
1224
2143
2133
1143
2134
1133
1243
1134
1233
1234

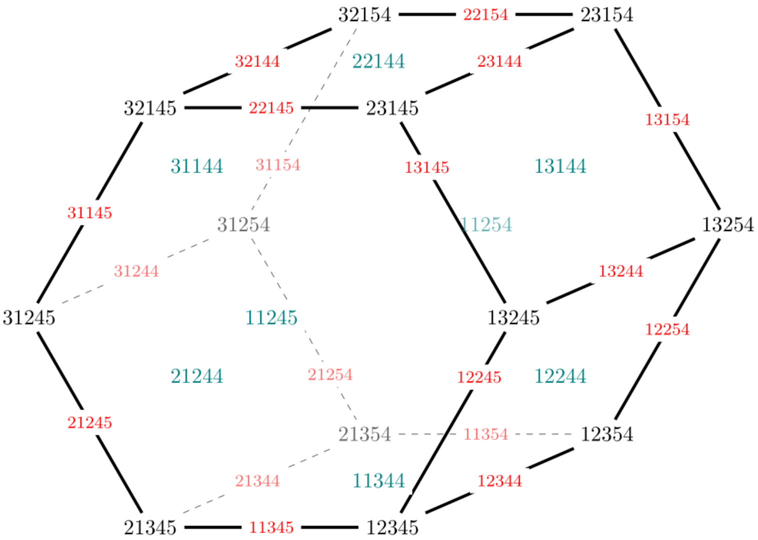

4321
4311
3321
4312
3311
3421
3312
3411
3412
4221
4212
2421
4231
4112
2231
2412
4213
2431
4131
3231
2213
1412
2413
4113
2331
3213
4132
3131
3241
1431
4122
2241
4123
2341
2313
1413
3113
1331
3132
3141
3214
1432
2214
1422
2314
1313
1423
3123
1341
3114
3142
1332
2123
1142
1314
1323
3124
1342
1124
1223
2142
1324
2124
1242
1224
2143
2133
1143
2134
1133
1243
1134
1233
1234

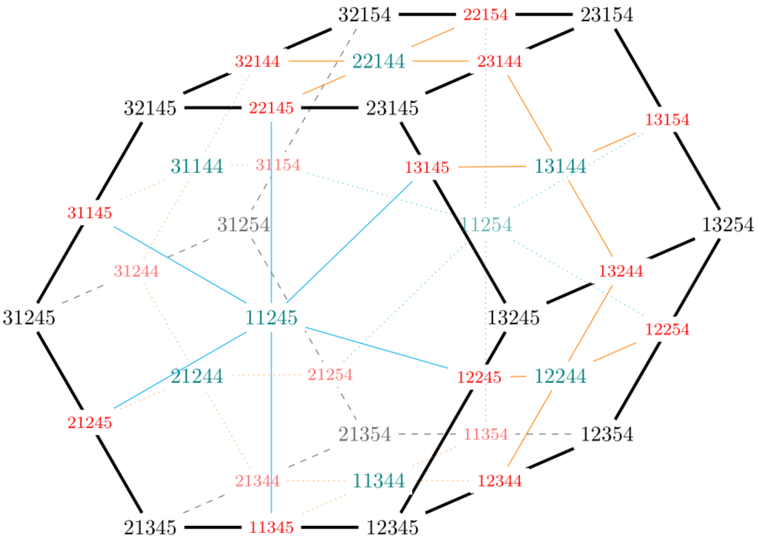

32154
23154
32145
23145
31254
13254
31245
13245
21354
12354
21345
12345

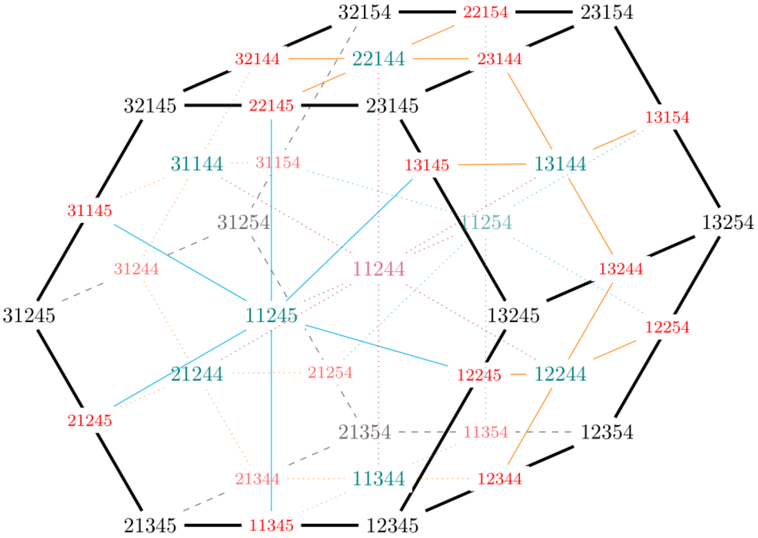

32154
23154
32145
(12)
23145
(23)
31254
(13)
13254
31245
13245
(13)
21354
(23)
12354
21345
(12)
12345


32154
22154
23154
32144
23144
32145
22145
23145
13154
31154
13145
31145
31254
13254
31244
13244
31245
13245
12254
21254
12245
21245
21354
11354
12354
21344
12344
21345
11345
12345


32154
22154
23154
32144
23144
32145
22145
23145
13154
31154
13145
31145
31254
13254
31244
13244
31245
11245
13245
12254
21254
12245
21245
21354
11354
12354
21344
12344
21345
11345
12345


32154
22154
23154
32144
22144
23144
32145
22145
23145
13154
31144
31154
13145
13144
31145
31254
11254
13254
31244
13244
31245
11245
13245
12254
21244
21254
12245
12244
21245
21354
11354
12354
21344
11344
12344
21345
11345
12345


32154
22154
23154
32144
22144
23144
32145
22145
23145
13154
31144
31154
13145
13144
31145
31254
11254
13254
31244
13244
31245
11245
13245
12254
21244
21254
12245
12244
21245
21354
11354
12354
21344
11344
12344
21345
11345
12345


32154
22154
23154
32144
22144
23144
32145
22145
23145
13154
31144
31154
13145
13144
31145
31254
11254
13254
31244
11244
13244
31245
11245
13245
12254
21244
21254
12245
12244
21245
21354
11354
12354
21344
11344
12344
21345
11345
12345

References

Ira Gessel (2006).
The refinement of Cayley's Formula for Trees.
Brandeis University ark:/13030/m5kq312b.

Yasmin Aguillon, Pamela Harris, et al.
On Parking Functions and the Tower of Hanoi.
To appear.

Catherine Yan (2015).
Parking Functions, Handbook of Enumerative Combinatorics.
vol. CRC Press Boca Raton, FL

References

Permutohedron Tesselation, Andrew Kepert,
https://hexnet.org/content/truncated-octahedron?layout=print.

Nonisomorphic rooted plane trees with 5 nodes, Joe Sawada,
https://www.researchgate.net/figure/Nonisomorphic-rooted-plane-trees-with-5-nodes$_fig7_2$20390379.

Thank You!

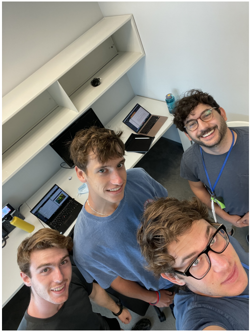


Figure: First selfie

Figure: Second selfie

Figure: Third selfie