Preferential and k-Zone Parking Functions

Parneet Gill, Christopher Soto, Pamela Vargas

Faculty Mentors: Rebecca E. Garcia, Pamela E. Harris, Dwight A. Williams II
Graduate Mentors: Carlos Martinez, Casandra Monroe

October 16, 2021

MSRI

Overview

- Introduction
- Relationships between Parking Functions
- Computational Results
- Other Results
- Summary

Parking Functions

- Imagine j cars enter a one-way street consisting of j parking spots and a list of parking preferences q.
- Each car entering has a preferred spot.
- If that spot is empty, then the car parks.
- If the spot is taken, then there are 5 variances of parking rules that the car can follow.

Motivating Questions

- What are the relationships between different parking functions?
- Are there any connections to other combinatorial objects?

Types of Parking Rules

G℮9 _H9℮ p9}c_j X S«jH§_nj I Checks only forward for available spots.

c] i 9q℮0 p9}c_j X S«jH§_nj I Checks c spots backwards one at a time.

c]µnj0 p9}c_j X S«jH§_nj I Checks back immediately c spots for availability.

p}0T0}0j§_9℮ p9}c_j X S«jH§_nj I Checks j _ spots back for an available spot.

^j®0} 0 p}0T0}0j§_9℮ p9}c_j X S«jH§_nj I Checks _ 1 spots back for an available spot.

Types of Parking Rules

Ge9 _H9e p9}c_jX S«jH§_nj I Checks only forward for available spots.

c] i 9qe0 p9}c_jX S«jH§_nj I Checks c spots backwards one at a time.

c]µnj0 p9}c_jX S«jH§_nj I Checks back immediately c spots for availability.

p}0T0}0j§_9e p9}c_jX S«jH§_nj I Checks j _ spots back for an available spot.

^j®0} 0 p}0T0}0j§_9e p9}c_jX S«jH§_nj I Checks _ 1 spots back for an available spot.

Classical Parking Functions

K0»j_§_nj

- Each car has a preferred spot which it goes to when entering the street.
- If parking spot is empty, car parks.
- Otherwise it continues down the street until it finds an empty spot to park in.

Consider the following parking preference vector $q = (2; 3; 1; 4)$:

i	p_i	Configuration			
1	2	___	c_1	___	___
2	3	___	c_1	c_2	___
3	1	c_3	c_1	c_2	___
4	4	c_3	c_1	c_2	c_4

Types of Parking Rules

G℮9 _H9℮ p9}c_jX S«jH§_nj I Checks only forward for available spots.

c] i 9q℮0 p9}c_jX S«jH§_nj I Checks c spots backwards one at a time.

c]µnj0 p9}c_jX S«jH§_nj I Checks back immediately c spots for availability.

p}0T0}0j§_9℮ p9}c_jX S«jH§_nj I Checks j _ spots back for an available spot.

^j®0} 0 p}0T0}0j§_9℮ p9}c_jX S«jH§_nj I Checks _ 1 spots back for an available spot.

k-Naples Parking Functions

K0»j_§_nj

- Each car prefers a spot (q), in which it attempts to park in.
- If spot empty, car parks.
- If occupied, car backs up checking c spots behind it's preferred spot <u>one at a time</u> and parks in first available.
- If there are no empty spots between q c and q , then car continues down the street and parks in first available.

Consider the following parking preference vector $q = (4; 4; 3; 2; 4)$:

i	p_i	k	Configuration				
1	4	2	___	___	___	c_1	___
2	4	2	___	___	c_2	c_1	___
3	3	2	___	c_3	c_2	c_1	___
4	2	2	c_4	c_3	c_2	c_1	___
5	4	2	c_4	c_3	c_2	c_1	c_5

Types of Parking Rules

Gℯ9 _H9ℯ p9}c_jX S«jH§_nj I Checks only forward for available spots.

c] i 9qℯ0 p9}c_jX S«jH§_nj I Checks c spots backwards one at a time.

c]µnj0 p9}c_jX S«jH§_nj I Checks back immediately c spots for availability.

p}0T0}0j§_9ℯ p9}c_jX S«jH§_nj I Checks j _ spots back for an available spot.

^j®0} 0 p}0T0}0j§_9ℯ p9}c_jX S«jH§_nj I Checks _ 1 spots back for an available. spot.

k-Zone Parking Functions

K0»j_§_nj

- Each car prefers a spot (q_i), in which it attempts to park in.
- If spot is empty, car parks.
- If occupied, car backs up immediately c spots behind it's preferred spot and then moves down the street if spot $q_i - c$ is taken.
- Parks in first available spot.

Consider the following parking preference vector $q = (4; 4; 3; 2; 4)$:

i	p_i	k	Configuration				
1	4	2	__	__	__	c_1	__
2	4	2	__	c_2	__	c_1	__
3	3	2	__	c_2	c_3	c_1	__
4	2	2	c_4	c_2	c_3	c_1	__
5	4	2	c_4	c_2	c_3	c_1	c_5

k-Zone vs. k-Naples

Gn j a0H§«}0

c] i 9qe0 _ 9 «@ 0§ nT c]µn j 0u

For $c = 2$:

j	c-Naples	c-Zone
1	1	1
2	4	4
3	27	27
4	240	244
5	2,731	2,808
6	38,034	39,416
7	627,405	654,302

Non-Increasing Preference Vectors

Consider the following table of the total number of parking functions formulated from non-increasing preference vector of length j.

For $c = 2$:

j	c-Naples	c-Zone
1	1	1
2	3	3
3	10	10
4	34	34
5	117	117
6	407	407
7	1,430	1,430

Findings From the Table

¦ \0n}0 h

d0§ $q = (q_1, \dots, q_j) \in [j]^j$ *@0 9 jnj]_jH}09 _jX q}0T0}0jH0 ®0H§n}u*
¦ \0j q _ 9 c] i 9qe0 _T 9jL nje´ _T q _ 9 c]µnj0u

p}nnT

We prove this by induction on the total number of cars, from last to first, that switch their parking rule from c-Naples to c-Zone and vice versa.

Enumerating *k*-Zone

Consider the table enumerating the number of c-Zone parking functions of length j and fixed parameter c.

j	$c = 0$	$c = 1$	$c = 2$	$c = 3$	$c = 4$	$c = 5$	$c = 6$	$c = 7$
1	o							
2	3	V						
3	16	24	©¢					
4	125	203	244	©U£				
5	1,296	2,225	2,808	3,065	¨Jo©U			
6	16,807	30,067	39,416	44,424	46,296	V£J£U£		
7	262,144	484,071	654,302	757,919	805,543	821,023	P©¨JUV¨	
8	4,782,969	9,057,316	12,553,351	14,880,368	16,110,376	16,613,896	16,757,056	o£J¢¢¢J©o£

Findings From the Table (Continued)

Gn j a0H§«}0

^T j 2 9jL 0 c j 1J §\0j jµpS(j; c 1)j jµpS(j; c 2)j _ 0x«9e §nI

- ¦ \0 n}L0} nT §\0 9e§0}j9§_jX X}n«q 8_{j+1}
- i «h@0} nT [9h_e§nj_9j H´He0 nj §\0 Hnhqe0§0 X}9q\J b_j
- i «h@0} nT j0Hce9H0 °_§\ j L_ §_jH§ @09L Tn} j! @09L q0}h«§9§_nj u

Formula:

$$\frac{j!}{2} \tag{1}$$

$$1; 3; 12; 60; 360; 2520; 20160; 181440; \ldots$$

coinciding with the OEIS sequence A001710.

Types of Parking Rules

Gɐ9 _H9ɐ p9}c_jX S«jH§_nj I Checks only forward for available spots.

c] i 9qɐ0 p9}c_jX S«jH§_nj I Checks c spots backwards one at a time.

c]µnj0 p9}c_jX S«jH§_nj I Checks back immediately c spots for availability.

p}0T0}0j§_9ɐ p9}c_jX S«jH§_nj I Checks j _ spots back for an available spot.

^j®0} 0 p}0T0}0j§_9ɐ p9}c_jX S«jH§_nj I Checks _ 1 spots back for an available spot.

Preferential Parking Functions

K0»j_§_nj

- Each car prefers a spot ($q_$), in which it attempts to park in.
- If spot is empty, car parks.
- Otherwise car checks j _ spots behind $q_$, one by one.
- If all the j _ spots preceding $q_$ are taken, car continues down the street until it finds an available spot to park in.

Consider the following parking preference vector $q = (6; 6; 4; 3; 3; 3)$:

i	p_i	$n-i$	Configuration					
1	6	5	___	___	___	___	___	c_1
2	6	4	___	___	___	___	c_2	c_1
3	4	3	___	___	___	c_3	c_2	c_1
4	3	2	___	___	c_4	c_3	c_2	c_1
5	3	1	___	c_5	c_4	c_3	c_2	c_1
6	3	0	___	c_5	c_4	c_3	c_2	c_1

Types of Parking Rules

Ge9 _H9e p9}c_jX S«jH§_nj I Checks only forward for available spots.

c] i 9qe0 p9}c_jX S«jH§_nj I Checks c spots backwards one at a time.

c]µnj0 p9}c_jX S«jH§_nj I Checks back immediately c spots for availability.

p}0T0}0j§_9e p9}c_jX S«jH§_nj I Checks j _ spots back for an available spot.

^j®0} 0 p}0T0}0j§_9e p9}c_jX S«jH§_nj I Checks _ 1 spots back for an available spot.

Inverse Preferential Parking Functions

K0»j_§_nj

- Each car prefers a spot ($q_{\_}$), in which it attempts to park in.
- If spot is empty, car parks.
- If occupied, car <u>checks _ 1</u> spots behind $q_{\_}$, one by one.
- If all _ 1 spots behind $q_{\_}$ are taken, car continues down the street until it finds an available spot to park in.

Consider the following parking preference vector $q = (6; 6; 4; 3; 3; 3)$:

i	p_i	$i-1$	Configuration					
1	6	0	___	___	___	___	___	c_1
2	6	1	___	___	___	___	c_2	c_1
3	4	2	___	___	___	c_3	c_2	c_1
4	3	3	___	___	c_4	c_3	c_2	c_1
5	3	4	___	c_5	c_4	c_3	c_2	c_1
6	3	5	c_6	c_5	c_4	c_3	c_2	c_1

Preferential Parking Function Findings

d0h h9

^T j 2J §\0j $(j; \ldots ; j) 2 [j]^j$ _ jn§ 9 q}0T0}0j§_9e q9}c_jX T«jH§_nju

p}nnT

By contradiction assuming that all cars can park.

Other Results

d0h h9

8ee q}0T0}0 j H0 ®0H§n} 9}0 _j ®0} 0 q}0T0}0 j §_9e q9}c_j X T« j H§_n j u

p}nnT

Direct proof.