Exploring the Naples parking functions

Christo Meriwether Keller

(with Laura Colmenarejo, Pamela Harris, Zakiya Jones, Andrés Meléndez, Andrés Rodríguez, and Eunice Sukarto)

UMass Math REU / Williams College AIM UP

Generalizing permutations

A string of numbers is a permutation iff its increasing rearrangement is **equal to** $1, 2, \ldots, n$.

Equivalently, the increasing rearrangment $a_i = i$.

What is a natural way to generalize these objects?

Change equality to inequality!

Generalizing permutations

A string of numbers is a parking function iff its increasing rearrangement is **less than or equal to** $1, 2, \ldots, n$.

Equivalently, the increasing rearrangment $a_i \leqslant i$.

Now how can we think about these objects?

Konheim & Weiss 1966 introduced a *parking rule* [1]

Preliminaries on parking functions

Imagine n cars drive down a one-way road with n parking spots.

Each car prefers a particular parking spot, which it drives to first.

If the spot is open, it takes it. Else it takes the next available spot.

Cars that can park like this determine a parking function!

Playing with parking functions

Examples of parking functions:

- Any permutation $\pi \in \mathfrak{S}_n$ is a parking function.
- The trivial preference $1, 1, \ldots, 1$ is a parking function.

Examples of not parking functions:

- Cars wanting spots $3, 2, 2$ fail to park under our rule.
- The preference $\underbrace{n, n, \ldots, n}_{n \text{ times}}$ isn't a parking function for $n > 1$.

Generalizing parking functions

Now let's generalize the parking rule!.

Allow a car to check one spot behind it if its spot is taken.

Cars that can park like this are called Naples. [3]

So for instance $(3, 2, 2)$ is Naples.

Generalizing Naples parking functions

Let's generalize the parking rule again!.

Allow a car to check k spots behind it if its spot is taken.

Cars that can park like this are called k-Naples. [4]

So for instance $(3, 3, 3)$ is 2-Naples.

Counting parking functions

We know there are $n!$ permutations of length n, and it's well-known that there are $(n+1)^{n-1}$ parking functions. [2]

Is there a nice formula for the k-Naples parking functions?

Christensen et al. 2019 gave a recursion on n for fixed k: [4]

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

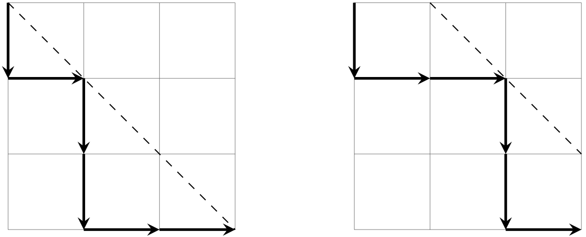
which has no known closed formula solution.

Counting parking functions

So what can we do? We can try to count **rearrangements**.

Because classical parking functions are characterized by their increasing rearrangement, all rearragements are parking functions.

The decreasing k-Naples parking functions are in bijection with SE-lattice paths below the line $y = n - x + k$ starting south.

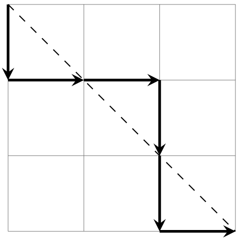


(a) $3, 1, 1 \in \mathrm{PF}_{3,0}$ (b) $3, 3, 1 \in \mathrm{PF}_{3,1}$

Rearranging parking functions

We prove that all rearrangements of 1-Naples parking functions with exactly one corner above the main diagonal are also 1-Naples.

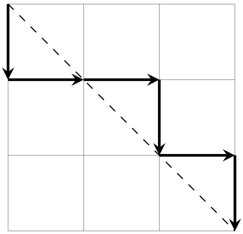
(This is Conjecture 5.6 from Christensen et al. 2019)



$3, 3, 1$ rearranges to $1, 3, 3$ and $3, 1, 3$

Rearranging parking functions

Of course, this doesn't work when there is more than one bump!

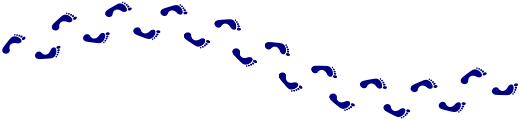


$3, 3, 2$ rearranges to $2, 3, 3$

Other directions of out work

- First results on reordering general k by considering bigger bump-sets over the main diagonal.
  - If π is k-Naples, then the decreasing arrangement of is also.
- Counting subsets of the decreasing parking functions.
- Counting the number of functions that park certain ways.

ᛏᛅᚴᚴᚠᚢᚱᛁᚱ



References and further reading

1. Konheim and Weiss, An occupancy discipline and applications, SIAM J. Appl. Math. 14 (1966), 1266-1274.
2. Stanley, Parking functions, available online as parking.pdf
3. Baumgardner, The naples parking function, Honors Contract-Graph Theory, Florida Gulf Coast University (2019)
4. Christensen et al, A generalization of parking functions allowing backward movement, The Electronic Journal of Combinatorics 27.1 (2019), 1.33