

I ended up working on organizing my files (still need to finish) and then working on computing/proving some things about when order of the cars matters

Typed board work

Making distance displaced precise

Previously, we talked about distance that a car has been displaced as a potential "permutation"-statistic (parking-statistic). One thing that we didn't make precise was how cars which are not parked count towards this. There are two logical choices: inclusive distance and exclusive distance. For inclusive distance, the cars which cannot park count as having a distance displaced all the way to the end of the parking lot. For exclusive distance, we ignore these cars.

Exclusive distance is not invariant under the action of S_n . For the notation, subscript indexes when the car gets to attempt to park, superscripts indicate the desired spot. Counter example:

$$\begin{aligned} 2 \textcolor{red}{2} 3 4 &\rightarrow _ \underline{c_3^2} \underline{c_2^3} \underline{c_1^4} && \text{exclusive distance: 0} \\ 4 \textcolor{red}{3} 2 2 &\rightarrow _ \underline{c_1^2} \underline{c_2^2} \underline{c_3^3} && \text{exclusive distance: 2} \end{aligned}$$

Order invariance of parkability for 1-element subset preferences

Note that S_n is generated by adjacent transpositions, so to show that something is invariant under the action of S_n , we just need to show that it is invariant under the action of adjacent transpositions. Consider the action of the transposition $(i \ (i+1))$. The first $i-1$ cars will have parked in the same way, so we think of this as two different cars trying to park given a set of occupied spots.

Given the spots occupied, the i th car and $i+1$ th car each have a preferred spot p_i and p_{i+1} respectively. If $p_i \neq p_{i+1}$, it doesn't matter what order these two cars park since they want to park in different spots. However if there is a collision and $p_i = p_{i+1}$, then the collision will change the outcome of which car gets to park where. However, a collision will not change which spots are ultimately occupied. Additionally, it will not change the cumulative distance that the cars are displaced

2-element subsets depend on the order of the cars Example:

$$\begin{aligned} (2, 4)(1, 2)(1)(4) &\rightarrow \underline{(1)} \underline{(1, 2)} _ \underline{(4)} && \text{parkability: 3} \\ (1, 2)(2, 4)(1)(4) &\rightarrow \underline{(1)} \underline{(2, 4)} \underline{(1, 2)} \underline{(4)} && \text{parkability: 4} \end{aligned}$$

I think this might be the smallest example of order changing parkability

Future directions/things I didn't get done/questions

- Think about the submodule of $S_n \wr S_n$ where parkability for 1 element subset preferences is constant
- Read more on wreath product rep theory (along the way learn more about Gelfand pairs (related to two sided cosets, algebra connection with commutative algebra) and spherical functions (related to multiplicity free-ness and modules whose representations only need one column of an irreducible representation))
- Read Theses
- When can graph notion of local and rep theory notion of local match up? When can graph notion of local provide a finer categorization of local? Does that relate to when $G \subset Aut(\Gamma)$ (ex: \mathbb{Z}_2^n) versus $G = Aut(\Gamma)$
- Develop better more consistent notation and names for parking function adjacent things

Photos from board work

An outline for code structure:

Thoughts on how to code

obj: parking
n: # spots

choice: Should cars be referred to by index in cars or an intrinsic value?

by default $m=n$

cars: List of length m , each element in list is a car obj
lot: List of constraints of parking lot (len: n)
park: returns list of len n w/ either car obj or None - should always be n to update Lot.

displacement: total displacement from desired spot
parkability: total # cars which can park } can use lot

apply-wr (list: S_n^n, S_n): Apply wreath prod. elem, then repark

Outline for why order doesn't matter for one element subset preferences

Given 1 element subsets

1 2 ... n

1 . .

If the ordered list L leaves spot x open, so does σL for $\sigma \in S_n$

Case 1: x is in the list. Done.

Case 2: x is the first spot & L then let

$$N = |\{l \in L \mid l \leq x\}|$$

if $N < x$, x must be unoccupied
& parking spots: $[x+1, n]$ are the same sub problem, totally disregarding $[0, x]$

if $N \geq x$, x must be occupied & $[N+1, n]$ is its own subproblem, still considering $[0, N]$

In

Total distance from desired spot is order invariant (given the right way of counting distance).
And parkability is not invariant under action of subgroup $e \in S_n$ (aka depends on order of cars).

Is distance also order invariant? depends on how you count unparked

$$\begin{array}{r} \dots \dots \dots \\ \downarrow \\ \dots \dots \dots \\ \dots \dots \dots \end{array}$$

unparked = 0
 unparked = 1 (head) unparked = 0 (parking lot cars)
 0 6 7
 2 6 7
 not invariant
 always differ by 1
 $\sum_{i=0}^n (i-1)$

In case of ∞ parking lot, let us show distance is invariant under $e \in S_n$
 S_n generated by adj. trans. - just look at three.

between diff. groupings (from left), order doesn't change parking at all
 Consider $(i, i+1)$. @ time of parking, already $i-1$ cars parked

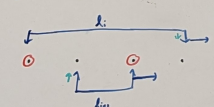
$[0, x]$
 idling $[0, N]$

Spot i is 1st car want
 want given occupied

if desired spots are diff, order doesn't matter at all
 if desired spots are same, order matters for which cars where
 but not for total distance.

2 elem. subgrp?

Same logic as dist. inv. arg.
 S_n gen. by adj. trans.
 look at parking lot after i cars park

ex. 

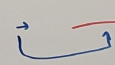
notation: $d_i(l)$
 is the spot that car i wants
 after i cars park

cases:

- $d_i(l_i)$ and $d_i(l_{i+1})$ don't collide. Yay!
- $d_i(l_i)$ & $d_i(l_{i+1})$ collide

and:

- both are second choice. same as 1-tuple
- one of each



$\{2, 4, 1, 2\} \{1, 3, 5, 4\} \Rightarrow \frac{3}{0} \frac{4}{0} - \frac{1}{0} \frac{2}{0} = \frac{5}{2}$ not
 $\{1, 2\} \{2, 4, 3, 1\} \{3, 5, 4\} \Rightarrow \frac{3}{0} \frac{4}{0} - \frac{5}{1} \frac{2}{0} = \frac{5}{2}$ not
 need at least 4 to make contradiction.