List of Preference List Function ideas

Parking function statistics

Note a mark of S_m means invariant under reordering. A mark of D means invariant under the action of the diagonal subgroup for linear probing/circular parking

Based on displacement

- Total displacement (also called area [2]) $S_m D$
- Number of cars displaced by exactly d (for d = 0, this is the lucky function). Note that lucky appears in [5, 4] D
- Maximum displacement (of interest for linear probing) D
- Other questions about the distribution of displacements (like moments) D

Based on elements in preference list and their order

- Preference of first car [3]
- 1s. Number of cars which want to park in the first spot: $|\{i|\pi_i = 1\}|$. Appears in/is related to [3, 5]
- ks. An easy extension of 1s: Number of cars which want to park in the Kth spot: $|\{i|\pi_i = k\}|$. Related to [3]
- Repeats. Number of cars which want the same spot as the car immediately before them: $|\{i|\pi_i = \pi_{i+1}\}|$. Appears in [5] relating invariance under probabilistic parking. D
- Leading elements. Number of cars with the same preference as the first car. Appears in [5] - they show leading elements and ones have the same relation to unlucky by a generating function argument (pg 13). D

Based on resulting permutation - specifically for $parking(\pi)$ which lists which car ended up in which spot (ex: parking(221) = 312). Discussed in the context of $n = m$

- descents. Number of descents of $parking(\pi)$ (must be next to each other). Appears in [5], relates to number of leaves -1 for trees
- skyscrapers. For $parking(\pi)$ written in one line notation, if you were to put a sudoku skyscrapers clue at the end of the parking lot, what would it be? Appears in [5] as rlm, relates to degree of the root node for trees
- inversions of $parking(\pi)$

- longest increasing sub-sequence of $parking(\pi)$
- minimum number of adjacent transpositions to generate $parking(\pi)$
- number of cycles in $parking(\pi)$

Based on breakpoints (spots which are occupied and no other car has attempted to park there)

- Location of last break point S_m
- Number of break points $S_m D$
- For a given spot, over the parking process, how many cars attempted to park there? rotates with action of diagonal subgroup
- Questions about the distribution of the lengths of the prime parking function components D

Preference list statistics - linear parking lots

Ideas based on who can park

- **Defect**. The number of cars which are unable to park S_m
- First car unable to park
- Last car able to park
- Mean value of the cars unable to park
- Variance of the set of cars unable to park

Based on empty spots

- Which spots are unoccupied. First? Last?

Based on probabilistic parking (probability p to go forward, probability $1-p$ to go backwards)

- Probability that the preference list is a parking function. Appears in [5] does order matter?
- Probability that the preference list is a defect d preference lists
- Expected value of defect
- Anything else of the form: probability that *statistic* = fixed value, expected value of *statistic*, variance of *statistic*

Sections based off of parking function statistics, with notes on additinal considerations.

Based on displacement. Note that the definition of displacement can be extended in a couple of different ways:

- Inclusive displacement - cars that don't park count as displaced to the end
- Infinite displacement - cars that don't park count as if they had an infinite parking lot
- Exclusive displacement - cars that don't park are completely ignored

The one which preserves the most invarients under reordering properties is infinite displacement. Once you pick a definition, any questions about displacement for parking function statistics can be extended to a preference list.

Note that lucky doesn't depend on how you define displacement.

Based on elements in preference list. All of these parking function statistics naturally extend to preference lists

Based on resulting permutation. Would need to modify to use partial permutations (relate to rook monoid)

Based on breakpoints No modifications needed

Preference list statistics - linear probing/circular parking lot

Note: let D_{n-1} be the diagonal subgroup for just the first $n-1$ elements. Then cosets of $D_{n-1}D$ can be identified with parking functions.

Based on displacement
No modifications to parking function definitions required

Based on elements in preference list and their order
No modifications to parking function definitions required

Based on resulting permutation
Standard choice for obtaining a permutation = linear probing
Other choices are possible in terms of returning to the back of the queue (choices - where in the queue are cars sent, second preference?). For different choices, the resulting permutation is different.

Note that for linear probing, there is a module homomorphism which identifies the action of the diagonal group with the action of the cyclic subgroup of S_n associated with the shift operator

Other notation/choices

Notation/language

- A preference list refers to the list of preferences.
  Notation: $P_{n,m}$ (also sometimes P_m^n) for n cars and m spots. P_n just means n spots and m cars

- A preference list with one element subset preferences $\pi \in P_{n,m}$ is an element $\pi = (\pi_1, \pi_2, \cdots \pi_n)$ where each $\pi_i \in \mathbb{Z}$, $1 \leq \pi_i \leq m$

- A parking function is a preference list where all of the cars can park.
  Notation: PF_n (Diaconis, Hicks)

- A prime parking function is a parking function with no break points (See 6/20 write up)

- Given a preference list π, $filling(\pi)$ indicates where each car ended up parking. $parking(\pi)$ indicates what car ended up parking in each parking spot. These are permutations in the case of parking functions and partial permutations in the more general case. In the case of permutations, they are inverses of each other [1].

- The best underlying group to act on this module is $C_n \wr S_m$. Where each element of C_n^m acts on each individual coordinates and S_m acts to rearrange the cars:

  $$(c, \sigma) \cdot \pi(x) = (c, e) \cdot \pi(\sigma^{-1}x) = c_{\sigma^{-1}x} \cdot \pi(\sigma^{-1}x)$$

  Some important subgroups are S_m, C_m^n, and D the diagonal subgroup of C_m^n.

Choices/Variations that I saw in the past

- **Preference Type** For the above, the preference list statistics explored were for 1-element subsets, but there could also be preference lists for other preference types. (k-element subsets, 1-element subsets which will only go an additional k spots before giving up, posets, etc). I think that the next most interesting kind of preference to investigate might be 1-element subsets which will give up after k spots

- What group should we be working with? Past thoughts on options:
  - C_n
  - C_n^m
  - $C_n \wr S_m$
  - S_n^m
  - $S_n \wr S_m$

References

[1] D Armstrong, A Garsia, J Haglund, Brendon Rhoades, and Bruce Sagan. Combinatorics of tesler matrices combinatorics of tesler matrices in the theory of parking functions and diagonal harmonics. *Journal of Combinatorics*, 3, 05 2012.

[2] Andrew Berget and Brendon Rhoades. Extending the parking space. *Journal of Combinatorial Theory Series A*, 123, 03 2013.

[3] Persi Diaconis and Angela Hicks. Probabilizing parking functions. *Advances in Applied Mathematics*, 89:125–155, 2017.

[4] Ira M. Gessel and Seunghyun Seo. A refinement of cayley's formula for trees, 2005.

[5] Richard P. Stanley and Mei Yin. Some enumerative properties of parking functions, 2023.