Summer Research

Random Sampling from Defect-d Preference Lists

Jasper Bown

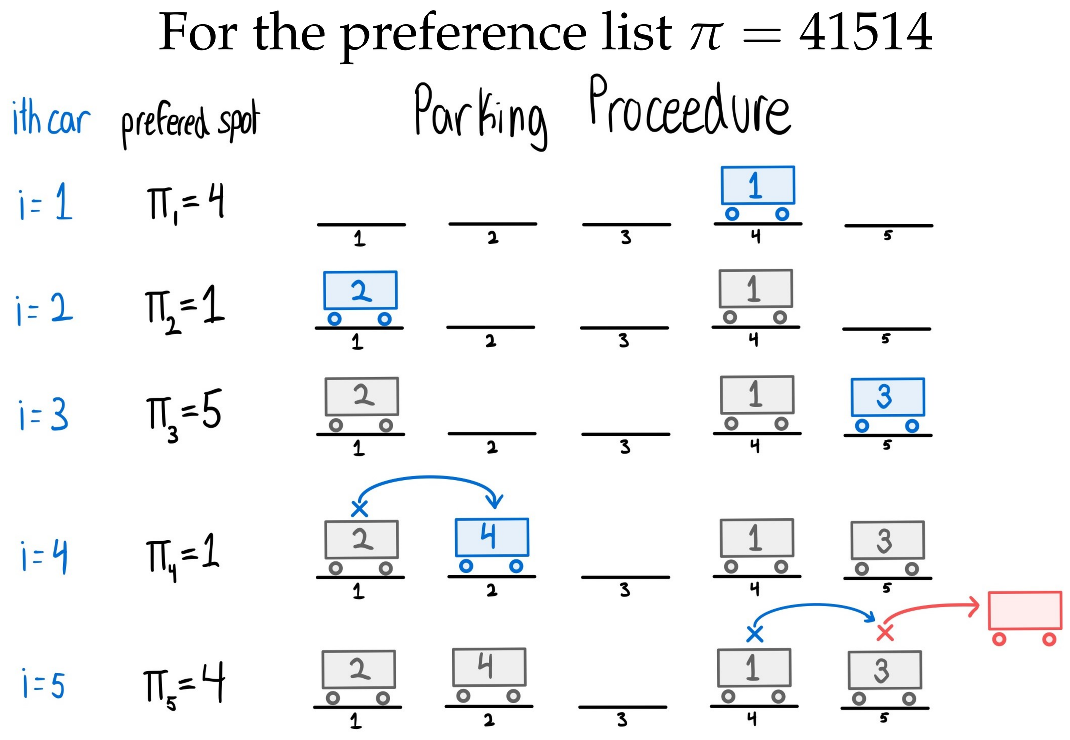
Introduction

Understanding how to sample uniformly often provides a useful starting point for understanding a space, enabling computation and highlighting structure. This poster describes the process of randomly sampling from defect-d preference lists.

Definitions

A **preference list**, $\pi \in [n]^m$, represents the parking preferences for m cars on a one way street with n spots.

For the preference list $\pi = 41514$



The **defect** of a preference list is the number of cars unable to park.

A **parking function**, $\pi \in PF_{n,m}$, is a preference list for n spots and m cars where all cars are able to park.

A **breakpoint** is an occupied spot where no other car attempts to park. For the above example, 2 is a break point.

A **prime parking function**, $\pi \in PPF_n$, is a parking function for n spots and cars with exactly 1 break point.

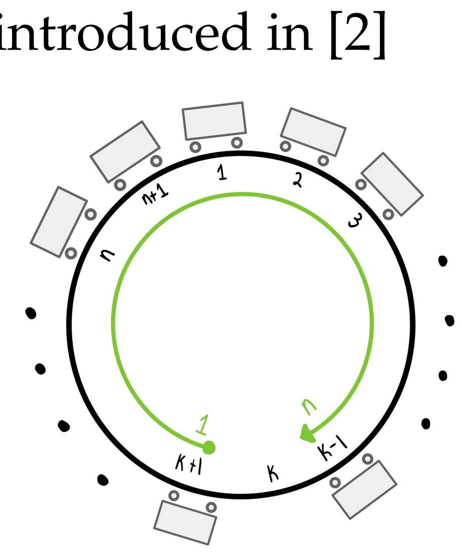
A **shuffle** interweaves two preference lists into a single preference list.
In the example above, $\pi = 41514$ is a shuffle of $\pi_1 = 11$ and $\pi_2 = 454$.

Existing Methods

Parking Functions

Key mathematical insight:
Every coset of the diagonal subgroup, $D = \langle (1, \cdots, 1) \rangle$, of C_{n+1}^m contains exactly 1 parking function.
This is Pollak's argument introduced in [2]
To sample:

- Pick $\pi \in C_{n+1}^m$.
- Pick $i \in [n-m+1]$.
- Let k be the ith empty spot from circular parking.
- Let $\pi_i' = \pi_i - k$.



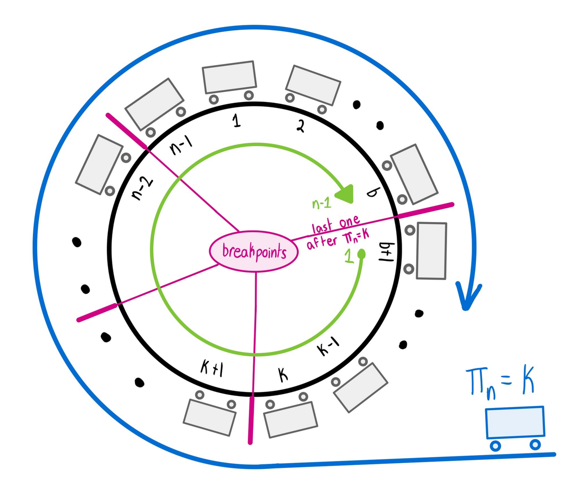
Enumeration:

$$|PF_{n,m}| = (n-m+1)(n+1)^{m-1}$$

Prime Parking Functions

Key mathematical insight:
Every coset of the diagonal subgroup of C_{n-1}^n contains exactly 1 prime parking function.
First seen in [3].
To sample:

- Pick $\pi \in C_{n-1}^n$.
- Let b be the last breakpoint after π_n.
- Let $\pi_i' = \pi_i - b$.



Enumeration:

$$|PPF_n| = (n-1)^{n-1}$$

Sampling from Defect d

To sample from defect d preference lists, focus on s, the number of spots occupied after the last breakpoint. To sample:

- Sample s from the appropriate distribution given n and d.
- Generate a parking function from $PF_{n-s,m-s-d}$.
- Generate a prime parking function from PPF_{s+d} with no preferences for the last d spots. Let us call this a **d-prime parking function**, denoted $PPF_{n,d}$.
- Shuffle the two preference lists.

Distribution for s

Let $s(\pi)$ be the number of occupied spots after the last point and $d(\pi)$ the defect of π. Then the following generating function is computed by counting possible shuffles:

$$F(x,y) = \sum_{\pi \in [n]^m} x^{s(\pi)} y^{d(\pi)}$$

$$F(x,y) = \sum_{d=0}^{m-1} \sum_{s=1}^{m-d} \binom{m}{s+d} |PF_{n-s,m-s-d}||PPF_{s+d,d}| x^s y^d + (n-m)n^{m-1}x^0 d^0 .$$

Note that this generating function provides all of the necessary information to sample from the distribution for s given n and p. Additionally, it provides a different method to enumerate defect d parking functions than [1].

p-prime Parking Function

Method 1

- Sample from $[s]^n$ where $s = n - p$ until you get a prime parking function.

Removing the preference lists which have an empty first spot and those with their first break point at i gives the following count:

$$|PPF_{n,p}| = s^n - (s-1)^n - \sum_{i=1}^{s-1} \binom{n}{i} (i-1)^{i-1}(s-i)^{n-i}.$$

Method 2

- Repeatedly sample from prime parking functions until you get a p-prime parking function.

Following an inclusion-exclusion based argument to remove prime parking functions which are not p-prime gives the following:

$$|PPF_{n,p}| = \sum_{i=0}^{p} \binom{n}{i} (n-i-1)^{n-i-1}(p-i)^i(-1)^i .$$

Bridging the Two

- Abel's identity gives a computational way to show the two counts are equal.
- When $p < n - (n-1)^{\frac{n-1}{n}}$, method 2 is more likely to succeed on the first attempted sample. Otherwise the reverse is true.

Conclusions

- Break points and prime segments are basic building blocks of preference lists.
- If you understand an attribute of interest for parking functions and p-prime parking functions, the distribution for s provides a method of extending this result to defect d preference lists.
- Defect 1 preference lists can be understood with only prime parking functions and parking functions.

References

[1] Peter J. Cameron, Daniel Johannsen, Thomas Prellberg, and Pascal Schweitzer. Counting defective parking functions, 2008.

[2] Dominique Foata and John Riordan. Mappings of acyclic and parking functions. *Aequationes mathematicae*, 10:10–22, 1974.

[3] Richard P. Stanley and Sergey Fomin. *Enumerative Combinatorics*, volume 2 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1999.

Acknowledgments

Thank you to Prof Orrison for numerous helpful and fun conversations in addition to the incredible support along the way. Thank you to the Harvey Mudd Math department for funding this research.

Advisor: Michael Orrison