Summer@ICERM 2022 Projects

Susanna Fishel, Pamela E. Harris (lead), and Gordon Rojas Kirby

Focus of the program: The proposed Summer@ICERM 2022 research topics include projects in a wide variety of subtopics in combinatorics and are motivated by computational experimentation stemming from the study of families of combinatorial objects known as parking functions.

1 Introduction to Parking Functions

Consider a parking lot consisting of n consecutive parking spots along a one-way street, where n is a positive integer. Suppose n cars want to park one at a time in the parking lot and each car has a preferred parking spot. Each car coming into the lot initially tries to park in its preferred spot. However, if a car's preferred spot is already occupied, then it will park in the next available spot. Since the parking lot is along a one-way street, it is not guaranteed that every car will be able to park before driving past the parking lot. This dilemma leads to the idea of a *parking function*.

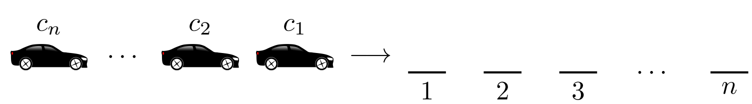


Figure 1: Parking function illustration.

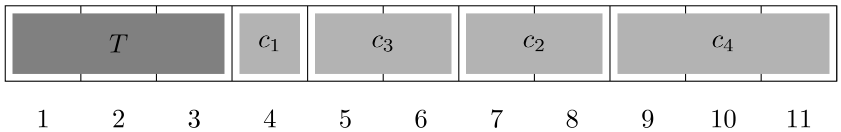
Let us make this definition precise. For a positive integer n, let $[n] := \{1, \ldots, n\}$. Formally, suppose the parking spots are labeled $1, 2, \ldots, n$, in order, along the one-way street and the cars are labeled according to the order in which they try to park. In other words, for each $i \in [n]$, car c_i is the i^{th} car to try to park and prefers spot $a_i \in [n]$. Note that more than one car can have the same preference. This is illustrated[1] in Figure 1. To park, cars first drive to their preferred spot and park in it if it is available. If their preferred spot is occupied then they drive forward and park in the next available spot. If all n cars can park in the parking lot under these conditions, then the preference list $(a_1, a_2, \ldots, a_n)$ is called a **parking function** (of length n). For example, $(1, 2, 4, 2, 2)$ is a parking function, but $(1, 2, 2, 5, 5)$ is not. Naturally, the first question that arises is: *"For any $n \in \mathbb{N}$, how many parking functions are there?"* Konheim and Weiss [11] showed that the number of parking functions of length n is $(n + 1)^{n-1}$.

2 Invariant and Prime Parking Sequences

Ehrenborg and coauthors [4, 5] generalized parking functions to parking sequences. In this new model, the car c_i has length $y_i \in \mathbb{N}$ for each i. Call $\vec{y} = (y_1, y_2, \ldots, y_n)$ the *length vector*. There is also a trailer T of length $z - 1$ parked at the beginning of the street. Given a sequence $\vec{p} = (p_1, \ldots, p_n) \in \mathbb{N}^n$ for $i = 1, \ldots, n$ the cars enter the street in order, and car c_i looks for the first empty spot $j \geq p_i$. If the spaces j through $j + y_i - 1$ are all empty, then c_i parks in these spots. If j does not exists or any of the spots $j + 1$ through $j + y_i - 1$ is already occupied, then there will be a collision and we say the parking fails.

[1] Black car vector. Digital image. The London Telegraph. 13 August 2019, `https://www.goodfreephotos.com/vector-images/black-car-vector.png.php`.

Assume there are $z - 1 + \sum_{i=1}^{n} y_i$ parking spots along the street, with the first $z - 1$ being occupied by a trailer. The sequence $\vec{p}$ is called a **parking sequence for** $(\vec{y}, z)$, where $\vec{y} = (y_1, \ldots, y_n)$ if all n cars can park without any collisions. For example, $\vec{p} = (3, 7, 5, 3)$ is a parking sequence for $\vec{y} = (1, 2, 2, 3)$ and $z = 4$ in which cars $c_1, \ldots, c_n$ and trailer T park in the following configuration.



We can ask the question of which properties of parking functions can be generalized to the setting of parking sequences. For example, all parking functions are invariant under permutation, but not all parking sequences are. Parking sequences which may be arbitrarily rearranged and stay a parking sequence are called **invariant**. Characterizing and counting how many invariant parking sequences there are for a general $(\vec{y}, z)$ remains an open problem. Additionally, a parking function of length n is called **prime** if any instance of a 1 may be removed with the result that the remaining entries form a parking sequence of length $n-1$ and their count is given by $(n-1)^{n-1}$. One can generalize this notion to parking sequences by removing an instance of the smallest preference p_j and its corresponding length.

Adeniran and Yan [25] studied parking sequences for special length vectors, specifically the case $\vec{y} = (k, k, \ldots, k)$, the case where $\vec{y}$ is strictly increasing, and lastly where $\vec{y}$ is of the form $(a, \ldots a, b, \ldots b)$. An interesting connection to another generalization of parking functions known as $\vec{u}$-parking functions (see [12]), in which the cars preferred parking spots must satisfy a condition related to a fixed vector $\vec{u}$, was also discovered. In AIM UP `https://sites.google.com/view/aimup/home` a bijection was found between invariant parking sequences for length vectors of the form $(a, \ldots, a, b)$ and $\vec{u}$ parking functions, where the vector $\vec{u}$ depends only on z and n. The bijection leads to a determinantal formula for the invariant parking sequences.

In light of this work we propose the following research projects:

1. Consider invariance for other special cases of length vectors and determine the number of such invariant parking sequences with or without a trailer placed at the start of the street.
2. Consider cars $c_1, \ldots, c_n$ with length vector $\vec{y} = (y_1, \ldots, y_n)$. How many prime parking sequences for $(\vec{y}; z)$ are there? An initial approach might be to consider the special case where $\vec{y} = (k, k, ..., k)$ and then generalize to other cases.

3 k-Naples parking functions

Several generations of REU students (MSRI UP 2019 and AIM UP 2020) have explored k-Naples parking functions. In this generalizations cars are allowed backward movement in the following sense. The parking rule for the parking preference $\alpha = (a_1, a_2, ..., a_n)$ is as follows. Car c_i drives to its preferred spot a_i. If it is unoccupied it parks there. Otherwise, the car backs up one spot at a time up to k spaces and parks in the first available spot preceding a_i. However, if none of the spaces $a_i, a_i - 1, \ldots a_i - k$ are available then car c_i proceeds forward to the next available unoccupied spot. If all n cars are able to park in the parking lot under these conditions then we say that α is a k-Naples parking function.

For example, consider the parking preference $(3, 1, 3, 4)$. This is not an ordinary parking function because when car c_4 attempts to park it finds that its preferred spot 4 is occupied by c_3, and there are no more available parking spaces past spot 4. However, $(3, 1, 3, 4)$ is a 1-Naples parking function. Cars c_1, c_2 are able to parking in their desired spaces 3 and 1 respectively. Car c_3 finds its preferred spot 3 unavailable but is able to check spot 2 and park there. Then c_4 parks in its desired spot.

In past work REU students gave a description analogous to our first description of parking functions: if we write $(a_1, \ldots, a_n)$ in decreasing order, then $a'_i \leq i + k$ [3]. There is a recursive formula for their enumeration [3]. In Summer 2020's AIM UP, the students found closed formulas for several special cases. They defined a "recording permutation" function, which tracks where the cars park for a preference sequence, and were able to calculate the number of k-Naples parking functions which correspond to a fixed permutations. Much work remains to fully understand k-Naples parking functions. We propose the following research questions:

1. Develop generating functions for k-Naples parking functions.
2. Study various statistics such as the number of ascents, descents, ties, and displacement of k-Naples parking functions. Enumerate the number of length n k-Naples parking functions with a fixed number of ascents, descents, ties, displacement.
3. Answer a variety of problems stated in [3]. For example: Let $B_{n,k}$ be the set of all k-Naples parking functions of length n such that if cars $c_1, \ldots, c_{i-1}$ have already filled spaces $1, \ldots, a_i$, then there is no car c_i with a parking preference $1 \le a_i \le k$. Find a closed formula to count the number of elements in the compliment of $B_{n,k}$ in PP_n (parking preferences).

4 Parking Functions and Posets

We propose to continue the study of Stanley's bijection between maximal chains in the noncrossing partition poset NC_n and parking functions. We will restrict the bijection to subposets of NC_n, given by bond lattices of certain graphs.

To concretely describe the problem we recall that a partition of a set S is a collection of disjoint, nonempty subsets of S whose union is S. The subsets in the set partition are called its *blocks*. Recall that a simple graph $G = (V, E)$ is a set of vertices V along with a set E of two element subsets of V, called edges. Suppose $W \subseteq V$. The graph induced by W, denoted $G\mid_W$, is the graph with vertex set W and edges contained in W. An graph is connected if for any pair of vertices x and y, there is a sequence of vertices $x_0 = x, x_1, \ldots, x_k = y$ such that $\{x_i, x_{i+1}\}$ is an edge for $0 \le i \le k-1$. The elements of the bond partially ordered set (poset) $\mathrm{BL}(G)$ [6] of the simple graph G are the set partitions of $\{B_1, \ldots, B_j\}$ of V such that $G\mid_{B_i}$ is connected for $1 \le i \le j$. The partial order is given by reverse refinement: a set partition π is less than the partition σ if every block of π is contained in a block of σ.

With this background at hand we can now describe Stanley's bijection: A partition of the set $[n] = \{1, \ldots, n\}$ is called *noncrossing* if there do not exist $1 \le a < b < c < d \le n$ such that a and c are in the same block and b and d are in the same block. The set of noncrossing partitions of $[n]$ ordered by reverse refinement is called the noncrossing partition or Krewaras poset and denoted NC_n. A maximal chain in a poset is a sequence of elements $x_0 < x_1 < \cdots < x_k$ which is maximal under inclusion. Stanley [7] showed that the number of maximal chains of NC_n is the same as the number of parking functions of length n with a bijection.

It is known that for certain graphs G, the bond poset $\mathrm{BL}(G)$ is an induced subposet of the noncrossing partition poset. In AIM UP `https://sites.google.com/view/aimup/home`, we characterized such graphs. We also found several families (paths, cycles) of graphs where the restriction of Stanley's bijection to the bond lattice of the graph produced interesting classes of parking functions. Additionally, in [2], a paper resulting from an REU, the authors defined a different subposet of the noncrossing partition lattice based on Stanley's bijection and certain parking functions. They decomposed it as a product of posets, characterized its intervals, and found its Möbius functions, among many results. They made a conjecture about its order complex, proved in [8].

With this in mind we propose the following research questions:

1. How similar is the poset in [2] to the bond posets found in AIM UP? Can we ask similar questions?
2. Graphs that "look like" triangulations of a convex polygon produce maximal bond subposets of the noncrossing lattices. Which parking functions do these produce?
3. We have enumerated the parking functions for certain graphs. Can we do it for others? The numbers grow quickly and we'll need good programs.

5 G-parking functions

In 2004 Postnikov and Shapiro [23] introduced a new generalization of parking functions, G-parking functions associated to a connected directed graph. In this setting we let G be a graph with vertex set $[n]_0 = \{0, 1, 2, \ldots, n\}$, where multiple edges and loops are allowed. We think of the vertex 0 as the *root* and denote directed edges by an ordered pair of of vertices (i, j), where i is the tail and j is the head. For a vertex i the *indegree* $\mathrm{indeg}(i)$ is the number of edges with tail i and the *outdegree* $\mathrm{outdeg}(i)$ is the number of edges with head i. Addtionally, for any subset $U \subseteq [n]$ and $i \in U$ we define $\mathrm{outdeg}_U(i)$ to be $|\{(i, j) \in E(G) \mid j \notin U\}|$. Then a **$G$-parking function** is a function $f : [n] \to \mathbb{Z}_{\geq 0}$ such that for every subset of the vertices $U \subseteq [n]$ there exists a vertex $i \in U$ such that $f(i) < \mathrm{outdeg}_U(i)$. In particular, viewing the complete graph K_{n+1} as a directed graph on $[n]_0$ with one directed edge (i, j) for every $i \neq j$, we can see that a $f : [n] \to \mathbb{Z}_{\geq 0}$ is a G-parking function if and only if $(f(1), f(2), \ldots, f(n))$ is an ordinary parking function.

There is an interesting relation between G-parking functions and *chip-firing games* and hyperplane arrangements which we propose to study. We can describe a simple version of the chip-firing game on undirected graphs, where we allow multiple edges but no loops. Let G be a connected graph on the vertex set $[n]_0$, where 0 is the root. A *chip configuration* for G is any nonnegative integer vector $\mathbf{c} = (c_1, \ldots, c_n)$, where we interpret $\mathbf{c}$ as recording the number of "chips" located at each non-root vertex of G. A vertex v is said to be *unstable* if the number of chips at v is at least the number of neighbors of v, and a configuration is said to be *stable* if all non-root vertex is unstable. An unstable vertex may v *fire* by sending one chip to each of its neighbors. This results in a new configuration $\mathbf{c}'$ such that $c'_i = c_i - \deg(i)$ and $c'_j = c_j + e(i, j)$, where $e(i, j)$ is the number of entries between i and j.

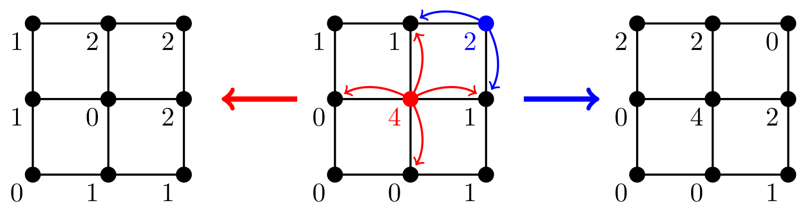


Figure 2: Firing at the red and blue unstable vertices

We may also consider a simultaneous firing at multiple vertices called a cluster-fire, and we call a configuration $\mathbf{c}$ *superstable* if there are no legal cluster-fires from $\mathbf{c}$. It is know that if G is a graph with sink vertex 0, then the G-parking functions of G are precisely the set of superstable configurations of G, see [26] for example.

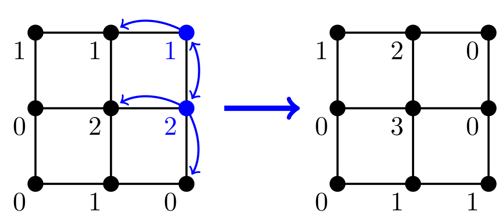


Figure 3: A cluster-fire at the blue vertices in the stable but not superstable configuration.

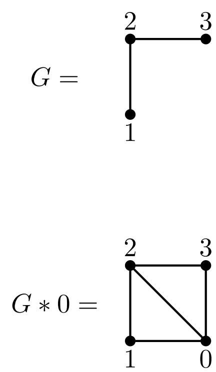
Geometrically, we can relate G-parking functions to a certain hyperplane arrangements known as *G-Shi arrangements*. Given a graph $G = (V, E)$ on n vertices, the G-Shi arrangement is the hyperplane arrangement in $\mathbb{R}^n$ consisting of hyperplanes

$$\{x_i = x_j, x_i = x_j + 1 \mid i < j, i, j \in E\}.$$

Pak and Stanley [27] showed for $G = K_{n+1}$ there is a labeling of the regions of the G-Shi arrangement in terms of ordinary parking functions. In fact, this construction can be extended to any graph G, and it turns out that the regions labels that appear are precisely $G*0$-parking functions or equivalently the superstable configurations of $G*0$ [26]. For example in Figure 4 we have that G is a path graph on 3 vertices with edges $\{1,2\}$ and $\{2,3\}$ accordingly the G-Shi arrangement consists of the hyperplanes

$$x_1 = x_2, \ \ x_1 = x_2 + 1, \ \ x_2 = x_3, \ \text{ and } x_2 = x_3 + 1.$$

Figure 4 depicts a projection of this hyperplane arrangement onto $\mathbb{R}^2$. The labels in each of the regions then correspond to ordered triples $f(0)f(1)f(2)$, which define a $(G*0)$-parking function $f : [3] \to \mathbb{Z}_{\geq 0}$. Note however, that this is not a bijective correspondence, as the $(G*0)$-parking function 010 appears twice. More work remains to better understand this correspondence.



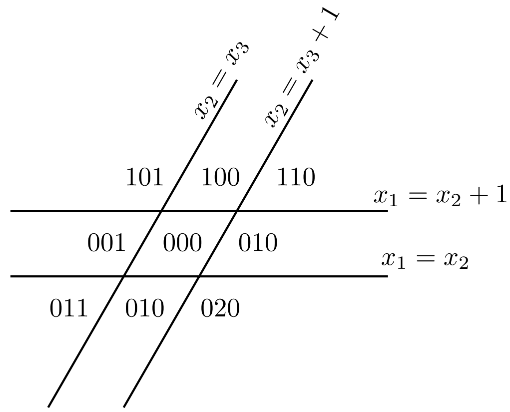


Figure 4: The labeling of the G-Shi arrangement in terms of $(G*0)$-parking functions.

With this in mind and the open problems mentioned in [26] we propose the following questions for future investigation:

1. Classify when the regions of a G-shi arrangement are in bijective correspondence with the $(G*0)$-parking functions.
2. Prove directly that the labels of the G-Shi labeling are componentwise downward closed.
3. Combinatorially connect these families of stable configurations to higher-dimensional trees.

References

[1] Alyson Baumgardner, *The Naples Parking Function* (2019). ↑

[2] Melody Bruce, Michael Dougherty, Max Hlavacek, Ryo Kudo, and Ian Nicolas, *A decomposition of parking functions by undesired spaces*, Electron. J. Combin. **23** (2016), no. 3, Paper 3.32, 22, DOI 10.37236/5940. MR3558069 ↑3

[3] Pamela E. Harris Alex Christensen Zakiya Jones, *A generalization of parking functions with backward movement*, The Electronic Journal of Combinatorics **27(1)** (2020), #P1.33, DOI https://www.combinatorics.org/ojs/index.php/eljc/article/view/v27i1p33/8024. ↑2, 3

[4] R. Ehrenborg and A. Happ, *Parking cars of different sizes*, Amer. Math. Monthly **23** (2016), 1045–1048. ↑1

[5] ——, *Parking cars after a trailer*, Australasian Journal of Combinatorics **70** (2018), no. 3, 402–406. ↑1

[6] Gian-Carlo Rota, *On the foundations of combinatorial theory. I. Theory of Möbius functions*, Z. Wahrscheinlichkeitstheorie und Verw. Gebiete **2** (1964), 340–368 (1964), DOI 10.1007/BF00531932. MR174487 ↑3

[7] Richard P. Stanley, *Parking functions and noncrossing partitions*, 1997, pp. Research Paper 20, approx. 14. The Wilf Festschrift (Philadelphia, PA, 1996). ↑3

[8] Michael Dougherty and Jon McCammond, *Undesired parking spaces and contractible pieces of the noncrossing partition link*, Electron. J. Combin. **25** (2018), no. 1, Paper No. 1.11, 13. MR3761925 ↑3

[9] Victor and Roichman Reiner Yuval, *Diameter of graphs of reduced words and galleries*, Trans. Amer. Math. Soc. **365** (2013), no. 5, 2779–2802, DOI 10.1090/S0002-9947-2012-05719-9. ↑

[10] Sami Assaf, *An inversion statistic for reduced words*, Adv. in Appl. Math. **107** (2019), 1–21, DOI 10.1016/j.aam.2019.02.005. MR3916545 ↑

[11] A. Konheim and B. Weiss, *An Occupancy Discipline and Applications*, SIAM Journal on Applied Mathematics **14** (1966), no. 6, 1266-1274, DOI 10.1137/0114101. ↑1

[12] Catherine H. Yan, *Parking functions*, Handbook of enumerative combinatorics, 2015, pp. 835–893. ↑2

[13] Ronald Pyke, *The supremum and infimum of the Poisson process*, Ann. Math. Statist. **30** (1959), 568–576, DOI 10.1214/aoms/1177706269. ↑

[14] Paul R. F. Schumacher, *Descents in parking functions*, J. Integer Seq. **21** (2018), no. 2, Art. 18.2.3, 8. ↑

[15] Alex Christensen and Pamela E. Harris and Zakiya Jones and Marissa Loving and Andrés Ramos Rodríguez and Joseph Rennie and Gordon Rojas Kirby, *A Generalization of Parking Functions Allowing Backward Movement*, Electron. J. Combin. **27** (2020), no. 1, P1.33. ↑

[16] Catherine H. Yan, *Generalized parking functions, tree inversions, and multicolored graphs*, Adv. in Appl. Math. **27** (2001), no. 2-3, 641–670, DOI 10.1006/aama.2001.0754. Special issue in honor of Dominique Foata's 65th birthday (Philadelphia, PA, 2000). ↑

[17] ———, *Generalized tree inversions and k-parking functions*, J. Combin. Theory Ser. A **79** (1997), no. 2, 268–280, DOI 10.1006/jcta.1997.2784. ↑

[18] Melody and Dougherty Bruce Michael and Hlavacek, *A decomposition of parking functions by undesired spaces*, Electron. J. Combin. **23** (2016), no. 3, Paper 3.32, 22, DOI 10.37236/5940. ↑

[19] Carrie Frizzell, *Generalized Bijective Maps between G-Parking Functions, Spanning Trees, and the Tutte Polynomial* (2020), available at `0807.3543`. preprint (arXiv:2005.06456). ↑

[20] Matjaž Konvalinka and Robin Sulzgruber and Vasu Tewari, *Trimming the permutahedron to extend the parking space* (2020), available at `2004.12093`. preprint (arXiv:2004.12093). ↑

[21] Jim Pitman and Richard P. Stanley, *A polytope related to empirical distributions, plane trees, parking functions, and the associahedron*, Discrete Comput. Geom. **27** (2002), no. 4, 603–634, DOI 10.1007/s00454-002-2776-6. ↑

[22] Paul R. F. Schumacher, *Descents in parking functions*, J. Integer Seq. **21** (2018), no. 2, Art. 18.2.3, 8. ↑

[23] Alexander Postnikov and Boris Shapiro, *Trees, parking functions, syzygies, and deformations of monomial ideals*, Trans. Amer. Math. Soc. **356** (2004), no. 8, 3109–3142, DOI 10.1090/S0002-9947-04-03547-0. ↑4

[24] Deepak Dhar, *Self-organized critical state of sandpile automaton models*, Phys. Rev. Lett. **64** (1990), 1613–1616, DOI 10.1103/PhysRevLett.64.1613. ↑

[25] Ayomikun and Yan Adeniran Catherine, *On increasing and invariant parking sequences*, Australas. J. Combin. **79** (2021), 167–182. ↑2

[26] Caroline J. Klivans, *The mathematics of chip-firing*, Discrete Mathematics and its Applications (Boca Raton), CRC Press, Boca Raton, FL, 2019. ↑4, 5

[27] Richard P. Stanley, *Hyperplane arrangements, interval orders, and trees*, Proc. Nat. Acad. Sci. U.S.A. **93** (1996), no. 6, 2620–2625, DOI 10.1073/pnas.93.6.2620. ↑5