Parking Functions on Directed Graphs and Some Directed Trees

Westin King
Department of Mathematics
Texas State University
San Marcos, TX, U.S.A.
westin_king@alumni.baylor.edu

Catherine H. Yan
Department of Mathematics
Texas A&M University
College Station, TX, U.S.A.
cyan@math.tamu.edu

Submitted: Oct 8, 2019; Accepted: May 19, 2020; Published: Jun 12, 2020

Abstract

Classical parking functions can be defined in terms of drivers with preferred parking spaces searching a linear parking lot for an open parking spot. We may consider this linear parking lot as a collection of n vertices (parking spots) arranged in a directed path. We generalize this notion to allow for more complicated "parking lots" and define parking functions on arbitrary directed graphs. We then consider a relationship proved by Lackner and Panholzer between parking functions on trees and "mapping digraphs" and we show that a similar relationship holds when edge orientations are reversed.

Mathematics Subject Classifications: 05A19

1 Introduction

Parking functions were first defined by Konheim and Weiss [8] during their study of the linear probing solution to collisions on hash tables. The authors described a sequence of n drivers attempting to park randomly along a one-way street. If the spot a driver attempts to park in is occupied, she drives to the next available spot and parks. As we are interested in the number of ways such a procedure results in all n drivers parking, we may consider the initial checked spot as a preferred parking place, rather than a randomly chosen spot. Let $s \in [n]^n$ and consider a directed path with vertex set $[n]$ and edge orientations $i \to i+1$. One-by-one the drivers attempt to park according to the following process:

1) Driver i begins at vertex s_i.

2) If the current vertex is unoccupied, the driver parks there. If it is occupied, the driver drives to the next vertex, following edge orientation, and repeats Step 2.

 https://doi.org/10.37236/9051

3) If she parks, the process continues with driver $i+1$ attempting to park at vertex s_{i+1}. Otherwise, the process terminates.

A sequence s such that all n drivers successfully park is called a *classical parking function* of length n. With some consideration, one can see that as long as there is no i such that too many drivers prefer a vertex from $\{i, i+1, i+2, \ldots, n\}$, then all drivers will park. This is formulated in an alternative definition of parking functions:

Definition 1 (Parking Functions)**.** A *classical parking function* of length n is a sequence $s \in [n]^n$ such that for all $i \in [n]$,

$$|\{j : s_j \geqslant i\}| \leqslant n - i + 1.$$

Additionally, we may consider the case when $m \leqslant n$ drivers attempt to park. We call these (n, m)-parking functions and the definition is the same as in the classical case for $s \in [n]^m$.

It is well-known that there are $(n+1)^{n-1}$ parking functions of length n, while there are $(n-m+1)(n+1)^{m-1}$ classical (n, m)-parking functions [3]. Classical parking functions have appeared throughout combinatorics as chains in the noncrossing lattice, in enumeration of hyperplane arrangements, in noncrossing partitions, and in tree enumeration (see [5, 12, 14] as well as references herein).

Parking functions have seen many generalizations: G-parking functions [11], $\mathbf{u}$-parking functions [9], parking sequences [4], rational parking functions [1], and those defined on tree-shaped parking lots [2, 10]. In this paper, we extend the "drivers searching for a parking spot" analogy from the trees in [10] to general digraphs and give a description that generalizes the set definition of the classical parking functions in Section 2. In Sections 3 and 4, we closely follow the results of Lackner and Panholzer [10] and show that many of their theorems have analogues when the edge orientations of the tree and mapping digraphs are reversed. Finally, we conclude with some new research directions and propose extensions to the concepts of increasing and prime parking functions.

This paper is, in part, an expansion of Sections 2 and 4 of [6].

2 Parking Functions on Digraphs

One interpretation of classical parking functions is of drivers attempting to park in a preferred spot along a street and parking in the first available spot they find afterwards. We may extend this notion to general digraphs by allowing driver to choose which out-edge to travel along. More formally,

Definition 2 (Parking Process)**.** Pick n, m such that $0 \leqslant m \leqslant n$. Let $s \in [n]^m$ and D be a digraph with vertex set $[n]$. One-by-one m drivers attempt to park according to the following process:

1) Driver i begins at vertex s_i.

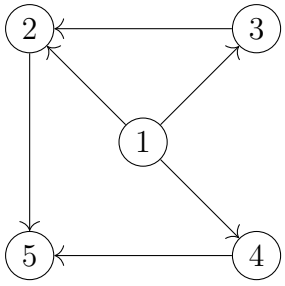

Figure 1: A digraph with parking function $s = (1, 1, 3, 2, 1)$.

2) If the current vertex is unoccupied, the driver parks there. If it is occupied, the driver chooses a vertex in the neighborhood of the current one and drives there.

3) The driver repeats step 2) until she either parks, and the next driver enters, or is unable to find an available parking space, and the process terminates.

When the maximum outdegree of a vertex in D is 1, the parking process is deterministic. In the general case, however, drivers must "choose" which edge to take in search of a parking spot. Our interest lies in the possibility of all drivers parking, so we give the following definition as our parking function generalization:

Definition 3. For a sequence $s \in [n]^m$ and digraph D with vertex set $[n]$, we say that s is a *parking function on* D if it is possible for all of the m drivers to park following the parking process. If s is a parking function on D, we call the pair (D, s) an (n, m)-*parking function.*

Figure 1 gives an example of an (n, n)-parking function. Drivers 2 and 5 are the only drivers who may make a choice of which edge to travel along and all drivers can park as long as at least one of those drivers uses the edge $(1, 4)$ during parking.

Definition 3 is clearly a generalization of the classical parking function case, but it is sometimes difficult to apply practically, so we now consider an equivalent definition that is more useful. For $i, j \in [n]$, we say $i \preceq_D j$ if and only if there exists a directed path from i to j in D. We have $i \preceq_D i$ for any i as the directed path between i and itself contains zero edges, making $\preceq_D$ a quasiorder on the vertices of D. If we wish to consider when $i \neq j$, we say $i \prec_D j$. For vertex i, define the set of vertices *reachable from* i as

$$R_D(i) = \{j \in [n] : i \preceq_D j\}.$$

Then for any $A \subseteq [n]$ define the *reachable set of* A as

$$R_D(A) = \bigcup_{i \in A} R_D(i).$$

Theorem 4. *Let D be a digraph with vertex set $[n]$ and $s \in [n]^m$. Let $C = \{C_1, \dots C_m\}$ be the set of cars, indexed such that the driver of C_i prefers spot s_i. Then s is a parking function on D if and only if for all $A \subseteq C$ we have*

$$|A| \leqslant |\bigcup_{s_i : C_i \in A} R_D(s_i)|.$$

Proof. If we let B be the bipartite graph with vertex set $C \cup [n]$ where $\{C_i, j\} \in E(B)$ if and only if $s_i \preceq_D j$, then by Hall's Theorem, we are claiming that s is a parking function if and only if there exists a matching on B saturating C. If s is a parking function, then park the drivers in some manner. Suppose C_i is parked on v_i for each i. Then, since we must have $s_i \preceq_D v_i$, the edge $\{C_i, v_i\}$ is in B. These edges define a matching that saturates C.

The other direction is not as clear because the parking process requires drivers to park in the first empty spot they find. We use a matching $M = \{\{C_i, v_i\}\}_{i=1}^m$ to determine a (not necessarily unique) way of parking on D.

The iterative process is the same for each C_i, starting at $i = 1$. At step i, pick any x with $s_i \preceq_D x \preceq_D v_i$ such that there exists a walk $s_i = y_0 \to y_1 \to \cdots \to y_k \to x$ such that the y_j's are spots occupied by cars in previous iterations. If no such y_0 exists, then $x = s_i$. Park C_i in x and delete $\{C_i, v_i\}$ from M. If $\{C_j, x\} \in M$ for some $j > i$, then replace this edge with $\{C_j, v_i\}$. Now repeat with C_{i+1}.

In each step, the vertex x is the first unoccupied vertex along some walk between s_i and the vertex with which C_i is matched. At least one such x exists because, at the start of step i, C_i is matched with a vertex which is not occupied by any car. At the end of step i, we know the updated M is a matching saturating $\{C_\ell\}_{\ell > i}$ because we know $s_j \preceq_D v_j = x$ from the edge $\{C_j, v_j\}$ and $x \preceq_D v_i$ by our choice of x. Thus, $s_j \preceq_D v_i$, so the edge $\{C_j, v_i\}$ is in B. □

Rather than considering subsets of the drivers, we may reformulate the statement in terms of the number of drivers wanting to park in various "regions" of the graph (c.f. Definition 1).

Corollary 5. *Let D be a digraph with vertex set $[n]$ and $s \in [n]^m$. Then s is a parking function on D if and only if for all $B \subseteq [n]$ we have*

$$|\{C_i : s_i \in R_D(B)\}| \leqslant |R_D(B)|.$$

Proof. Let s be a parking function on D and $B \subseteq [n]$. Let $A = \{C_i : s_i \in R_D(B)\}$. Because s is a parking function, we know $|A| \leqslant |\bigcup_{s_i : C_i \in A} R_D(s_i)|$, but also by the definition of $R_D(B)$, we have $\bigcup_{s_i : C_i \in A} R_D(s_i) \subseteq R_D(B)$.

On the other hand, suppose for all $B \subseteq [n]$ we have $|\{C_i : s_i \in R_D(B)\}| \leqslant |R_D(B)|$, let $A \subseteq C$, and $B = \bigcup_{C_i \in A} \{s_i\}$. By definition, $R_D(B) = \bigcup_{C_i \in A} R_D(s_i)$, and so

$$|A| \leqslant |\{C_i : s_i \in R_D(B)\}| \leqslant |R_D(B)| = |\bigcup_{C_i \in A} R_D(s_i)|,$$

thus s is a parking function. □

From this set definition, it is immediately obvious that the ordering of s does not matter.

Corollary 6. *Let (D, s) be a parking function and $\sigma \in \mathfrak{S}_m$. Then the permuted sequence $s_\sigma = (s_{\sigma(1)}, s_{\sigma(2)}, \ldots, s_{\sigma(m)})$ is also a parking function on D.*

Remark 7. The number of distinct $R_D(B)$ is the same as the number of filters of the quasiorder $\preceq_D$, which is, in general, much less than 2^n.

In the case of a classical parking function s, for any $B \subseteq [n]$ with smallest element b, we have that $|R_{\mathcal{P}_n}(B)| = |R_{\mathcal{P}_n}(b)| = n+1-b$. Thus, for $i \in [n]$, $|\{j : s_j \geqslant i\}| \leqslant n+1-i$, as in Definition 1. In addition, Corollary 5 generalizes the description of parking functions given in [10], which we now consider as we turn our attention to rooted trees with edges oriented away from the root.

3 Parking Functions on Source Trees

Lackner and Panholzer [10] and Butler, Graham, and Yan [2] independently generalized the "drivers searching for a parking space" interpretations of parking functions to rooted trees with edges oriented towards a root. In this section, we follow the enumerative discussion from [10] and show that several of their theorems have analogous results when edge orientations are reversed.

If T is a rooted tree with vertex set $[n]$ and edges oriented towards the root, we let $\widetilde{T}$ be the tree obtained by reversing the orientation of all the edges. In our pictures, the root will be the top-most vertex. We call these *sink* and *source* trees, matching whether the root is a sink or source vertex. Similarly, if M_f is the digraph obtained from $f : [n] \to [n]$ by letting $V(M_f) = [n]$ with edge set $E = \{(i, f(i))\}_{i=1}^n$, we let $\widetilde{M_f}$ be the digraph obtained from M_f by reversing edge orientations. That is, $E(\widetilde{M_f}) = \{(f(i), i)\}_{i=1}^n$. We will call these *mapping* and *inverse mapping* digraphs, respectively. We note that digraphs in each of these families have a single cycle on each connected component. Define the sets

$$\mathcal{T}_n = \{T : V(T) = [n] \text{ and } T \text{ is a rooted sink tree}\},$$
$$\mathcal{M}_n = \{M : M \text{ is the mapping digraph of some } f : [n] \to [n]\}.$$

We similarly define $\widetilde{\mathcal{T}_n}$ and $\widetilde{\mathcal{M}_n}$ for the source trees and inverse mapping digraphs.

Figure 2 gives an example of a source tree and an inverse mapping digraph along with an $s \in [7]^7$ that is a parking function on both.

For source trees, because the indegree of a node is at most 1, for two distinct vertices, u and v, the sets $R_{\widetilde{T}}(u)$ and $R_{\widetilde{T}}(v)$ are either disjoint or one contains the other. Thus, the 2^n inequalities in Corollary 5 reduce to only n independent inequalities.

We briefly introduce some notation out of convenience. We call u the *parent* of v if the two are adjacent and u lies on the unique path between the root and v. Here, we do

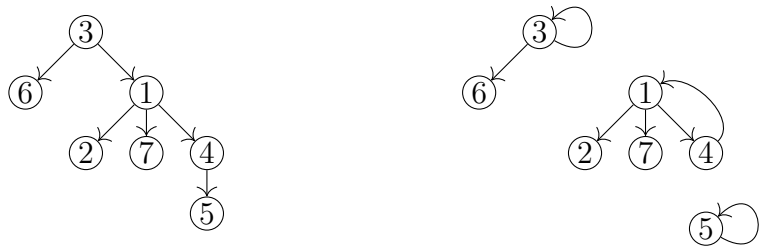

Figure 2: A source tree and inverse mapping digraph with a common parking function.

not consider edge orientation. Further, for a vertex u of $\widetilde{T}$, let $\widetilde{T}_u$ be the subtree induced by the set $R_{\widetilde{T}}(u)$. Thus, Corollary 5 can be restated as:

Corollary 8. *Let $\widetilde{T}$ be a source tree and $s \in [n]^m$. Then s is an (n, m)-parking function on $\widetilde{T}$ if and only if for all $u \in [n]$ we have*

$$|\{i : s_i \in \widetilde{T}_u\}| \leqslant |\widetilde{T}_u|.$$

Additionally, for digraph D with vertex set $[n]$, let

$$P(D, m) = |\{s \in [n]^m : s \text{ is a parking function on } D\}|.$$

We first study the extremal values of $P(\widetilde{T}, m)$.

Proposition 9. *Let $\widetilde{T}$ be a source tree, u a non-root vertex, v the parent of u, and w such that $v \preceq_{\widetilde{T}} w$ and $w \notin \widetilde{T}_u$. Let $\widetilde{T}'$ be the tree obtained by removing the edge (v, u) and adding the edge (w, u). Then $P(\widetilde{T}, m) \leqslant P(\widetilde{T}', m)$.*

Proof. To clarify which tree we are considering, we denote by x' the vertex in $\widetilde{T}'$ with label x. Let $(\widetilde{T}, s)$ be an (n, m)-parking function. By Corollary 5, we must check $|\{i : s_i \in \widetilde{T}'_{x'}\}| \leqslant |\widetilde{T}'_{x'}|$ for all $x \in V(\widetilde{T}')$. By the construction of $\widetilde{T}'$, $|\{i : s_i \in \widetilde{T}'_{y'}\}| = |\{i : s_i \in \widetilde{T}_y\}|$ and $|\widetilde{T}'_{y'}| = |\widetilde{T}_y|$ for all y' not satisfying $v' \prec_{\widetilde{T}'} y' \preceq_{\widetilde{T}'} w'$.

Therefore, let y' be a vertex satisfying $v' \prec_{\widetilde{T}'} y' \preceq_{\widetilde{T}'} w'$. We thus have:

$$\begin{aligned} |\{i : s_i \in \widetilde{T}'_{y'}\}| &= |\{i : s_i \in \widetilde{T}_y\}| + |\{i : s_i \in \widetilde{T}_u\}| \\ &\leqslant |\widetilde{T}_y| + |\widetilde{T}_u| \\ &= |\widetilde{T}'_{y'}|. \end{aligned}$$

□

As a result, we obtain an upper and lower bound on $P(\widetilde{T}, m)$.

Corollary 10. *The number of (n, m)-parking functions is maximized when $\widetilde{T}$ is a path and minimized when $\widetilde{T}$ is a star, meaning*

$$\sum_{i=0}^{m} \binom{m}{i} (n-1)^{\underline{m-i}} \leqslant P(\widetilde{T}, m) \leqslant (n-m+1)(n+1)^{m-1},$$

where $n^{\underline{k}} = n(n-1)(n-2)\cdots(n-k+1)$ is the falling factorial.

Proof. For a path, the parking functions, up to vertex labeling, are classical parking functions, and thus number $(n-m+1)(n+1)^{m-1}$.

On a star, for $0 \leqslant i \leqslant m$, when i drivers prefer the root, there are $\binom{n-1}{m-i}$ ways to choose the preferred non-root vertices, $\binom{m}{i}$ ways to place the drivers preferring the root in s, and $(m-i)!$ ways to order the drivers preferring non-roots in s. □

In the case of sink trees, the maximum and minimum number of parking functions also occur on paths and stars, respectively. From [10], we have

$$n^{\underline{m}} + \binom{m}{2} (n-1)^{\underline{m-1}} \leqslant P(T, m) \leqslant (n-m+1)(n+1)^{m-1}.$$

If T is a star, it is not immediately clear for an arbitrary choice of (n, m) which of $P(T, m)$ and $P(\widetilde{T}, m)$ is larger. However, we can say after inspecting the formulae that if m is 0 or 1 then the two are equal and if $n \geqslant 3$ then $P(T, 3)$ is larger. What happens when $3 < m < n$ remains open, but we can determine which is larger when $m = n$ for all T.

Theorem 11. *Let $T \in \mathcal{T}_n$. Then*

$$P(T, n) \leqslant P(\widetilde{T}, n)$$

with equality if and only if T is a path.

Proof. Let (T, s) be a parking function on sink tree T. We give a process to determine an involution $\tau \in \mathfrak{S}_n$ such that $\tau(s) = (\tau(s_1), \tau(s_2), \ldots, \tau(s_n))$ is a parking function on the source tree $\widetilde{T}$. Park cars on T following the parking procedure, highlighting an edge if it is used by a driver after failing to park at her preferred spot. Since T is a sink tree, each vertex has outdegree at most 1, so parking is deterministic. We define τ by individually considering the components connected by highlighted edges. So without loss of generality, we may assume that every edge in T is highlighted.

We define a collection of length $\geqslant 2$ "paths" in T, one for each leaf, whose vertices form a partition of the vertices of T. The purpose of these "paths" is to identify a section of the tree where we can "flip" the edge orientations and driver preferences and still guarantee each driver a spot to park.

Let $\{v_i\}$, for $1 \leqslant i \leqslant k$ be the leaves of T indexed such that $v_i < v_{i+1}$. We recursively define the sets P_i: the set P_i is the smallest set of vertices of the path between v_i and the root such that no vertices from $P_j, j < i$, are in P_i, there are $|P_i|$ drivers preferring P_i, and all vertices of P_i are connected to v_i through vertices in $\{P_j\}_{j=1}^{i}$. For example, on the left

tree of Figure 3, we have sets $P_1 = \{1, 2, 3, 5\}$ and $P_2 = \{4, 6\}$. Two drivers prefer the leaf $\{1\}$, three drivers prefer the vertices $\{1, 2\}$, four drivers prefer $\{1, 2, 3\}$ and $\{1, 2, 3, 5\}$, so the latter is P_1. When determining P_2, we skip over vertices in previously-chosen paths, in this case the vertex labeled 5.

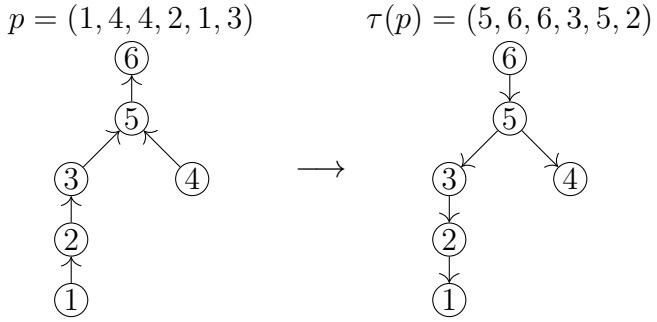

Figure 3: Constructing a parking function on $\widetilde{T}$ from one on T. $\tau = (15)(23)(46)$

The collection $\{P_i\}_{i=1}^k$ must partition the vertices of T. Suppose it does not and let $v \notin P_i$ for any i be such that v is the only vertex in T_v, the subtree rooted at v, with this property. The vertex v is not a leaf of T, as all leaves are in the sets $\{P_i\}_{i=1}^k$ by construction, and thus is the terminus of at least one edge, (u, v). Since none of the "paths" corresponding to leaves of T_v contain v and because all cars can park, all cars preferring spots $w \preceq_T u$ can park without occupying v. This means the edge (u, v) is not used by any driver after failing to park in her preferred spot, which contradicts our assumption that this was true of all edges. Therefore, such a v can not exist.

For each i, let $n_i = |P_i|$ and label the elements of P_i by $w_{i,j}$ such that $w_{i,1} = v_i \preceq_T w_{i,2} \preceq_T \ldots \preceq_T w_{i,n_i}$. Finally, we define $\tau(w_{i,j}) = w_{i,n_i+1-j}$. This reverses the driver preference along the "path" so that when the edge orientation is flipped for $\widetilde{T}$, the drivers may park as they did on T.

We can recover the P_i from $(\widetilde{T}, \tau(s))$ using the exact same method. Thus, we can invert the process. On the right tree of Figure 3, zero drivers prefer $\{1\}$, one driver prefers $\{1, 2\}$, two drivers prefer $\{1, 2, 3\}$, and four drivers $\{1, 2, 3, 5\}$, so this is P_1. For P_2, no drivers prefer $\{4\}$, we skip over 5 as it is already in P_1, and two drivers prefer $\{4, 6\}$.

If T is not a directed path, then this process is not surjective because the parking function in which all n drivers prefer the root of $\widetilde{T}$ is not obtainable in this manner as at least one driver prefers each leaf vertex. □

Define the following for $n \geqslant 1$:

$$F_{n,m} = \sum_{T \in \mathcal{T}_n} P(T, m), \text{ and } M_{n,m} = \sum_{M \in \mathcal{M}_n} P(M, m).$$

Similarly, define $\widetilde{F}_{n,m}$ and $\widetilde{M}_{n,m}$ for source trees and inverse mappings. Summing over all $T \in \mathcal{T}_n$ gives us

Corollary 12. *For $n \geqslant 1$,*

$$F_{n,n} \leqslant \widetilde{F}_{n,n},$$

with equality only when $n \in \{1, 2\}$.

As we previously mentioned, when $m < n$, the result of Theorem 11 does not necessarily hold. For the tree in Figure 4, $P(T, 2) = 15$, as any sequence except $(4, 4)$ parks. However, $P(\widetilde{T}, 2) = 14$ as neither $(1, 1)$ nor $(2, 2)$ are parking functions.

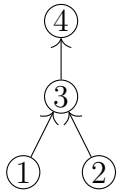

Figure 4: A tree T for which $P(T, 2) > P(\widetilde{T}, 2)$.

Remark 13. The extremal values for $P(\widetilde{M}_f, m)$ are less interesting than the corresponding values on trees. When $\widetilde{M}_f$ is a cycle, $P(\widetilde{M}_f, m)$ is maximal (as all n^m sequences can park) and is minimal when $f = id$. Thus

$$m! \leqslant P(\widetilde{M}_f, m) \leqslant n^m.$$

In fact, the same is true for $P(M_F, m)$.

4 Comparing $P(\widetilde{T}, m)$ and $P(\widetilde{M}_f, m)$.

Lackner and Panholzer [10] proved $n \cdot F_{n,m} = M_{n,m}$. In fact, this relationship still holds when the edge orientations are reversed. We prove the claim when $m = n$, then we will show the more general case. While the overall idea of the proofs are similar to their counterparts in [10], there are some technical differences in dealing with the source trees. Because, on sink trees, the spot in which each driver parked was well-defined, the authors of [10] were able to identify edges in the digraphs that were not used during parking and thus could be freely manipulated. In our case, as the drivers no longer necessarily have a unique walk along which to search for a parking spot, we must instead identify edges that are not necessary for *some* successful parking. This is simple enough on trees using the characterization of Corollary 5, but it is not immediately clear for inverse mapping digraphs. So, we first prove that at least one cycle edge on each component of an inverse mapping digraph is not needed for parking.

Lemma 14. *Let $(\widetilde{M}_f, s)$ be a parking function. Then there exists at least one edge in each cycle of $\widetilde{M}_f$ that can be deleted such that all drivers can still park.*

Proof. We induct on the number of vertices in a cycle. Let $(\widetilde{M_f}, s)$ be an (n,n)-parking function. Without loss of generality, we may assume there is only one component of $\widetilde{M_f}$ and thus a unique cycle in the graph. If only one vertex is in the cycle, then there is an edge of the form (u,u) which is useless for parking and may be deleted.

Now suppose the cycle has length $r > 1$. Furthermore, suppose without loss of generality that the vertices of the cycle are labeled by $[r]$, $f(r) = 1$, and $f(i) = i+1$ for $i \in [r-1]$. Let M be the graph obtained by deleting all cycle edges. Define for $1 \leqslant i \leqslant r$, $V_i := |R_M(i)|$ and $\alpha_i := |\{j : s_j \in R_M(i)\}|$. These are the numbers of vertices in and drivers preferring the subtree induced by the vertex i along with all $i \preceq_{\widetilde{M_f}} v$ for v non-cycle vertices in $\widetilde{M_f}$.

If $\alpha_i < V_i$ for all $i \in [r]$, then there are strictly fewer than n drivers attempting to park, contradicting the assumption that $(\widetilde{M_f}, s)$ is an (n,n)-parking function. Hence, we know $\alpha_i \geqslant V_i$ for some i. Since s is a parking function on $\widetilde{M_f}$, at least one driver prefers i (otherwise, too many drivers prefer non-cycle vertices). We construct an $(n-1,n-1)$-parking function by contracting the edge $(i,i-1)$ to identify the vertices $i-1$ and i as a single vertex (with label $i-1$) to form the digraph M', deleting the first instance of i from s, and changing all others to $i-1$ to form the sequence s'. For non-cycle vertex v, $|R_{\widetilde{M_f}}(v)| = |R_{M'}(v)|$, as are the number of drivers preferring each set. If v is instead a cycle vertex, then $n = |R_{\widetilde{M_f}}(v)| = |R_{M'}(v)| + 1$, while n drivers prefer $R_{\widetilde{M_f}}(v)$ and $n-1$ drivers prefer $R_{M'}(v)$. Thus, (M', s') is a parking function with $r-1$ vertices in the cycle. By the inductive hypothesis, there exists an edge e that can be deleted from M'. We claim this same edge in $\widetilde{M_f}$ is not necessary for parking via s.

Let T be the digraph obtained by deleting e from $\widetilde{M_f}$ and T' be obtained by deleting e from M'. Since (T', s') is a parking function, we know for any $v \in T'$, we have $|\{j : s'_j \in R_{T'}(v)\}| \leqslant |R_{T'}(v)|$. We now check the vertices of T to determine if (T, s) is a parking function.

Case 1: $i-1 \prec_T v$. We have

$$|\{j : s_j \in R_T(v)\}| = |\{j : s_j \in R_{T'}(v)| \leqslant |R_{T'}(v)| = |R_T(v)|.$$

Case 2: $v \prec_T i$. Then,

$$|\{j : s_j \in R_T(v)\}| = |\{j : s_j \in R_{T'}(v)| + 1 \leqslant |R_{T'}(v)| + 1 = |R_T(v)|.$$

Case 3: $v = i$ gives

$$|\{j : s_j \in R_T(i)\}| = |\{j : s_j \in R_{T'}(i-1)| + 1 \leqslant |R_{T'}(i-1)| + 1 = |R_T(i)|.$$

Case 4: $v = i-1$. Using the fact that $-\alpha_i \leqslant -V_i$, we know

$$\begin{aligned} |\{j : s_j \in R_T(i-1)\}| &= |\{j : s_j \in R_{T'}(i-1)| + 1 - \alpha_i \\ &\leqslant |R_{T'}(i-1)| + 1 - V_i \\ &= (|R_T(i-1)| + V_i - 1) + 1 - V_i \\ &= |R_T(i-1)|. \end{aligned}$$

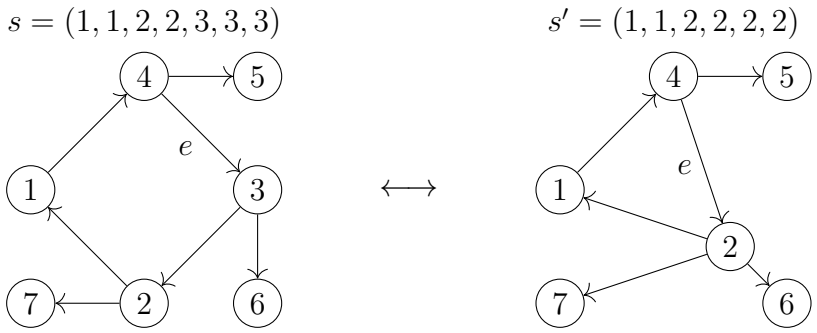

Figure 5: Identifying a deletable edge e by contracting $(3, 2)$.

Case 5: all other v. Since v is not a cycle vertex in $\widetilde{M}_f$ and s is a parking function on $\widetilde{M}_f$, the deleting of e does not affect the reachable set of v. Thus,

$$|\{j : s_j \in R_T(v)\}| = |\{j : s_j \in R_{\widetilde{M}_f}(v)\}| \leqslant |R_{\widetilde{M}_f}(v)| = |R_T(v)|.$$

So (T, s) is indeed a parking function and we know the edge e is not necessary for parking on $\widetilde{M}_f$. □

Figure 5 gives an example of the contraction to M' with $i = 3$. We identify a deletable e on M', which gives a deletable e on $\widetilde{M}_f$.

We now use Lemma 14 to prove

Theorem 15. *For $n \geqslant 1$, we have the relationship*

$$n \cdot \widetilde{F}_{n,n} = \widetilde{M}_{n,n}.$$

Construction of the bijection. Let $(\widetilde{T}, s)$ be a parking function for source tree $\widetilde{T}$, and pick $v \in V(\widetilde{T})$. We define a bijection ψ such that $\psi\left((\widetilde{T}, s, v)\right) = (\widetilde{M}_f, s)$ for some appropriate inverse mapping digraph $\widetilde{M}_f$, constructed by identifying edges in $\widetilde{T}$ that can be manipulated without affecting the ability of the cars to park. The sequence s will not change.

Let (u, w) be an edge in $\widetilde{T}$. If $|\{i : s_i \in \widetilde{T}_w\}| = |\widetilde{T}_w|$, then for any successful parking, no car may cross (u, w) as otherwise too many cars would attempt to park in the subtree $\widetilde{T}_w$. Additionally, at least one driver must prefer w, as one driver must park in w and no driver may use the edge (u, w). These two observations will allow us to select edges to manipulate in $\widetilde{T}$.

Consider the path $\text{root}(\widetilde{T}) = v_1 \to v_2 \to \ldots \to v_k = v$ for some $k \geqslant 1$. We first identify the edges that are freely manipulatable, then we use the order of s to choose a subset of those. For $1 \leqslant i \leqslant k$, let $v_i \in A$ if and only if $|\{j : s_j \in \widetilde{T}_{v_i}\}| = |\widetilde{T}_{v_i}|$. Since $\widetilde{T}_{v_1} = \widetilde{T}$, A is nonempty. By the second observation above, all $v_i \in A$ appear as preferences in s. The edge (v_{i-1}, v_i) is not used by any driver, so we may manipulate

it (even delete it) without affecting parking. Next, for the $v_i \in A$, we define the rank $d(v_i)$ to be the index of the first appearance of v_i in s. We now let $B \subseteq A$ be given by the elements $\{v_i : \forall j < i, d(v_j) < d(v_i)\}$. That is, the elements of B are those in A that appear in s after their ancestors from A. Note that $B \neq \emptyset$ as $\text{root}(\widetilde{T}) = v_1 \in B$.

Consider the unique sequence $\{v_{i_j}\}_{j=1}^{|B|}$ such that $v_{i_j} \in B$ and $i_j < i_{j+1}$. For $j > 1$, remove the edge (v_{i_j-1}, v_{i_j}) and add the edge $(v_{i_j-1}, v_{i_{(j-1)}})$. Finally, add the edge $(v, v_{i_{|B|}})$ (if v is the root, this will be a loop as then $v = v_1$). The resulting graph is an inverse mapping digraph, $\widetilde{M}_f$, where $f(i)$ is the unique j such that (j, i) is an edge. In particular, the edges that were manipulated and the one that was added are $\{(f(v_{i_j}), v_{i_j})\}_{j=1}^{|B|}$. □

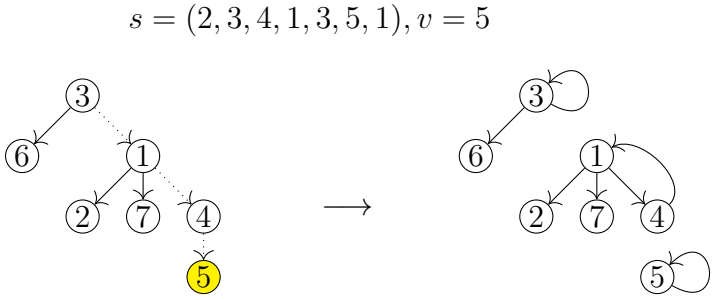

Figure 6: Turning a source tree into an inverse mapping digraph.

Figure 6 gives an example of ψ. In it, $A = \{1, 3, 4, 5\}$ and $B = \{1, 3, 5\}$, with $v_{i_1} = 3$, $v_{i_2} = 1$, and $v_{i_3} = 5$.

Remark 16. In each component, the cycle vertex appearing in B has the highest rank of all other cycle vertices in that component. Further, if an edge was necessary for parking on $\widetilde{T}$, it is still necessary for parking on $\widetilde{M}_f$.

For the inverse, we know by Lemma 14 that at least one edge in each cycle of the graph is not necessary for parking. We let the set $\hat{A}$ be the set of vertices in the cycles of $\widetilde{M}_f$ that are the terminal vertices of an edge that is not necessary for parking. Define $\hat{B} \subseteq \hat{A}$ as the set of vertices which have the highest rank in each cycle. By Remark 16, if $(\widetilde{M}_f, s) = \psi\left((\widetilde{T}, s, v)\right)$, we know $B = \hat{B}$. Label these elements of $\hat{B}$ by $\{b_i\}_{i=1}^{|\hat{B}|}$ such that $d(b_1) < d(b_2) < \ldots < d(b_{|\hat{B}|})$. For $1 \leqslant i \leqslant |\hat{B}| - 1$, remove the edge $(f(b_i), b_i)$ and add the edge $(f(b_i), b_{i+1})$. Finally, delete the edge $\left(f(b_{|\hat{B}|}), b_{|\hat{B}|}\right)$ and mark $f(b_{|\hat{B}|})$. The resulting tree is $\widetilde{T}$, so $\psi^{-1}\left((\widetilde{M}_f, s)\right) = (\widetilde{T}, s, f(b_{|\hat{B}|}))$.

As promised, we can extend this result to (n, m)-parking functions.

Theorem 17. *Let $n \in \mathbb{N}$ and $0 \leqslant m \leqslant n$. Then*

$$n \cdot \widetilde{F}_{n,m} = \widetilde{M}_{n,m}.$$

Proof. Let $s \in [n]^m$ be a parking function on $\widetilde{T} \in \widetilde{\mathcal{T}}_n$, and let $v \in V(\widetilde{T})$. Our goal is to extend s to $s' \in [n]^n$ in a reversible manner, then apply ψ. We must do so in a way that is not affected by the change in edges caused by ψ, which suggests we avoid using edges along the path $\text{root}(\widetilde{T}) \to v$.

To this end, drivers choose to park as follows. We recursively define $\{A_i\}_{i=1}^m$ so that $A_1 = \{s_1\}$ and in general A_i are the spots that driver i could park at so that the remaining drivers may successfully park, given the first $i-1$ drivers are parked. Let $B_i \subseteq A_i$ be the vertices of A_i that are reachable from s_i by utilizing a minimal number of edges in the path $\text{root}(\widetilde{T}) \to v$. From there, driver i parks at the vertex with label $\min(B_i)$. Once driver i is parked, we construct A_{i+1} and continue until all drivers have parked.

Let $\{x_i\}_{i=1}^{n-m}$ be the unoccupied spaces after the drivers have parked in this manner, ordered in an increasing manner. We then define s' as follows:

$$s'_i = \begin{cases} s_i & \text{if } i \leqslant m \\ x_j & \text{if } i = m + j \end{cases}$$

Then, we apply ψ to $(\widetilde{T}, s', v)$. Since s' does not change under ψ and is a parking function, s is also a parking function on the resulting mapping digraph $\widetilde{M}_f$. In order to reverse, we must be able to extend s to s' on $\widetilde{M}_f$. Because the edges on the path between $\text{root}(\widetilde{T})$ and v become the cycle edges, drivers park as defined in the first paragraph, but instead of utilizing a minimal number of path edges, they use a minimal number of cycle edges. □

5 Concluding Remarks and Future Research

In this paper, we gave several equivalent characterizations of an extension of the "drivers searching for a parking space" description of parking functions to digraphs. Additionally, we follow the work of Lackner and Panholzer to show many of their results on trees with edges oriented towards a root still hold when the edge orientation is reversed. Furthermore, we showed that source trees never had fewer (n, n)-parking functions than sink trees.

We propose here some generalizations of on other notions of classical parking functions. Prime parking functions were defined by Gessel [12] and can be understood as classical parking functions for which every edge in the path is necessary for parking. More formally,

Definition 18. A classical parking function $s \in [n]^n$ is *prime* if, for all $2 \leqslant i \leqslant n$, we have

$$|\{j : s_j \geqslant i\}| < n - i + 1.$$

So let us say

Definition 19. A parking function (D, s) is called *prime* if, for every $A \subseteq [n]$ such that $R_D(A) \neq [n]$, we have

$$|\{C_i : s_i \in R_D(A)\}| < |R_D(A)|.$$

In fact, the parking functions for which every edge is used by a driver after failing to park at her desired spot described in the proof of Theorem 11 are prime parking functions. See [7] for more about prime parking functions on sink trees.

Another type of interesting parking function are the *increasing* parking functions, those for which $s_i \leqslant s_{i+1}$ for all i. For the classical case, the set of increasing parking functions is counted by the famous Catalan numbers and is easily shown to be in bijection with Dyck paths of semilength n. We follow the terminology of [2] and call the generalization on digraphs *parking distributions*, to focus on the distribution of driver preference rather than the ordering of the sequence. Here, $f(i)$ may be understood as the number of drivers preferring vertex i.

Definition 20. Let $f : [n] \to [m]_0$ such that $\sum_{i=1}^{n} f(i) = m$. Then we say (D, f) is a *parking distribution* if for all $A \subseteq [n]$, we have

$$\sum_{i \in R_D(A)} f(i) \leqslant |R_D(A)|.$$

We present several avenues for future research:

1. We are interested in exact counts of $P(D, m)$ on specific digraphs or sums over families of digraphs. In particular, we do not know $\widetilde{F}_{n,m}$.

2. We are also interested in a formula more specific than that given in Theorem 11, describing the relationship between $P(T, n)$ and $P(\widetilde{T}, n)$.

3. As noted, it is possible for some $m < n$ and T to have $P(T, m) > P(\widetilde{T}, m)$. We ask for a characterization of when this occurs.

4. Both the path that classical parking functions are defined on and sink trees as a family support interesting numbers of parking functions. Are there other digraphs or families of digraphs which are associated with particularly nice numbers of (prime) parking functions or parking distributions?

5. We ask how one may extend several of the statistics defined on classical parking functions, such as the number of "lucky" drivers, those who park in their preferred spot, or the "total displacement", the distance driven by all drivers. In general, the locations in which drivers park is not well-defined, but perhaps we could get around this by considering a maximum or minimum over all successful parking outcomes.

The authors note that between submission for review and acceptance, Tian obtained some preliminary results towards 3. when the tree is either a star or the number of drivers is significantly less than the number of vertices [13].

Acknowledgments

The authors would like to thank Professor Chun-Hung Liu for suggesting the inductive proof of Lemma 14, which simplified our original argument. Additionally, the authors thank the anonymous reviewers for their close reading and comments.

References

[1] D. Armstrong, N. A. Loehr, and G. S. Warrington. Rational parking functions and catalan numbers. *Annals of Combinatorics*, 20(1):21–58, Mar 2016. https://doi.org/10.1007/s00026-015-0293-6.

[2] S. Butler, R. Graham, and C. H. Yan. Parking distributions on trees. *European Journal of Combinatorics*, 65:168 – 185, 2017. https://doi.org/10.1016/j.ejc.2017.06.003.

[3] P. J. Cameron, D. Johannsen, T. Prellberg, and P. Schweitzer. Counting defective parking functions. *Electronic Journal of Combinatorics*, 15 #R92, 2008. https://doi.org/10.37236/816.

[4] R. Ehrenborg and A. Happ. Parking cars of different sizes. *The American Mathematical Monthly*, 123(10):1045–1048, 2016. https://doi.org/10.4169/amer.math.monthly.123.10.1045.

[5] I. M. Gessel and S. Seo. A refinement of cayley's formula for trees. *Electronic Journal of Combinatorics*, 11(2), 2006, #R27. http://www.combinatorics.org/Volume_11/Abstracts/v11i2r27.htm.

[6] W. King and C. H. Yan. Parking functions on oriented trees. *Séminaire Lotharingien de Combinatoire*, (47). https://www.mat.univie.ac.at/~slc/wpapers/FPSAC2018/47-King-Yan.html.

[7] W. King and C. H. Yan. Prime parking functions on rooted trees. *Journal of Combinatorial Theory Series A*, 168:1–25, 2019. https://doi.org/10.1016/j.jcta.2019.05.015.

[8] A. G. Konheim and B. Weiss. An occupancy discipline and applications. *SIAM Journal on Applied Mathematics*, 14(6):1266–1274, 1966. http://www.jstor.org/stable/2946240.

[9] J. P. Kung and C. Yan. Gončarov polynomials and parking functions. *Journal of Combinatorial Theory, Series A*, 102(1):16 – 37, 2003. https://doi.org/10.1016/S0097-3165(03)00009-8.

[10] M.-L. Lackner and A. Panholzer. Parking functions for mappings. *Journal of Combinatorial Theory, Series A*, 142:1 – 28, 2016. https://doi.org/10.1016/j.jcta.2016.03.001.

[11] A. Postnikov and B. Shapiro. Trees, parking functions, syzygies, and deformations of monomial ideals. *Transactions of the American Math Society*, 356(8), 2004. https://www.ams.org/journals/tran/2004-356-08/home.html.

[12] R. Stanley. *Enumerative Combinatorics*, volume 2. Cambridge University Press, 1999.

[13] R. Tian. Parking functions on directed stars and orientation reversal with an extension to general directed trees, 2020. https://arxiv.org/abs/2001.06341v2.

[14] C. H. Yan. Parking functions. In M. Bóna, editor, *Handbook of Enumerative Combinatorics*, chapter 13, pages 835–894. CRC Press, 2015.