Discrete Mathematics and Theoretical Computer Science . . . . . **vol. 26:3 #5 (2024)**

Cost-sharing in Parking Games

Jennifer Elder[1*] Pamela E. Harris[2†] Jan Kretschmann[2]
J. Carlos Martínez Mori[3‡]

[1] *Department of Computer Science, Mathematics and Physics, Missouri Western State University, St. Joseph, MO, USA*
[2] *Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI, USA*
[3] *H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA*

revisions 25th Feb. 2024, 4th Sep. 2024; accepted 7th Sep. 2024.

In this paper, we study the total displacement statistic of parking functions from the perspective of cooperative game theory. We introduce *parking games*, which are coalitional cost-sharing games in characteristic function form derived from the total displacement statistic. We show that parking games are supermodular cost-sharing games, indicating that cooperation is difficult (i.e., their *core* is empty). Next, we study their *Shapley value*, which formalizes a notion of "fair" cost-sharing and amounts to charging each car for its expected marginal displacement under a random arrival order. Our main contribution is a polynomial-time algorithm to compute the Shapley value of parking games, in contrast with known hardness results on computing the Shapley value of arbitrary games. The algorithm leverages the permutation-invariance of total displacement, combinatorial enumeration, and dynamic programming. We conclude with open questions around an alternative solution concept for supermodular cost-sharing games and connections to other areas in combinatorics.

Keywords: parking functions, cooperative games, Shapley value

1 Introduction

Consider a one-way street with $n \in \mathbb{N} \coloneqq \{1, 2, \ldots\}$ numbered parking spots. A sequence of n cars arrive one at a time, each with a preferred spot. Upon the arrival of car $i \in [n] \coloneqq \{1, 2, \ldots, n\}$, it drives to its preferred spot $a_i \in [n]$. If spot a_i is unoccupied, it parks there. Otherwise, it is *displaced* down the one-way street until it finds the first unoccupied spot in which to park, if one exists. If no such spot exists, the car is unable to park and the parking process fails. Let $\alpha = (a_1, a_2, \ldots, a_n) \in [n]^n$ be the n-tuple encoding the parking preference of every car. If all cars are able to park, then α is a *parking function* of length n.

*J. Elder was partially supported through an AWM Mentoring Travel Grant.
†P. E. Harris was supported through a Karen Uhlenbeck EDGE Fellowship.
‡J. C. Martínez Mori is supported by Schmidt Science Fellows, in partnership with the Rhodes Trust. Part of this research was performed while J. C. Martínez Mori was visiting the Mathematical Sciences Research Institute (MSRI), now becoming the Simons Laufer Mathematical Sciences Institute (SLMath), which is supported by NSF Grant No. DMS-1928930.

ISSN 1365–8050

Parking functions were independently introduced by Konheim and Weiss (1966) in their study of hashing functions and by Pyke (1959) in his study of Poisson processes. Since then, parking functions have become classical objects in combinatorics, displaying rich mathematical structure of their own, as well as diverse connections to other research areas; including hashing Knuth (1998), hyperplane arrangements Stanley (1996), noncrossing partitions Stanley (1997), spanning trees Kreweras (1980), Dyck paths Armstrong et al. (2016), polyhedral combinatorics Amanbayeva and Wang (2022); Stanley and Pitman (2002), sandpile groups Cori and Rossin (2000), the Bruhat order Elder et al. (2023), Brownian motion Diaconis and Hicks (2017), and sorting Harris et al. (2024), to name just a few. Refer to Martínez Mori (2024); Carlson et al. (2021) for expository introductions to parking functions and their many variants, and to Yan (2015) for a survey of results.

In this paper, we study the "fair" distribution of parking costs. Specifically, we take the *total displacement* collectively incurred by all cars as the basis for the cost of parking. As a motivating example, consider the all-ones parking function $(1, 1, \ldots, 1) \in \mathrm{PF}_n$. In this case, car 1 is lucky and parks in spot 1 without incurring any displacement. However, car 2 is not as lucky, and it is displaced one unit before parking in spot 2. Similarly, car i is displaced $i - 1$ units before parking in spot i. Therefore, the total displacement of $(1, 1, \ldots, 1)$ is

$$\sum_{i=1}^{n}(i-1) = \frac{n(n-1)}{2}.$$

Certainly, one can recover the total displacement by charging each car for the displacement it incurs: that is, charging $i - 1$ units to each car i. However, this seems rather unfair given that all cars have the same preference; it just so happens that certain cars arrive before others (and the arrival order might be beyond the cars' control). Therefore, in this case, it seems only fair to charge $(n - 1)/2$ units to each car i, which again recovers the total displacement. Now, suppose some car changed its preference from spot 1 to spot 2. This would ever so slightly alleviate the parking demand around spot 1, and in fact the total displacement would decrease by one unit. How should this car be "fairly" compensated for its more favorable preference?

1.1 Summary of Results

We answer this question through the lens of cooperative game theory (refer to Peleg and Sudhölter (2007) for a comprehensive treatment of this area, and to Myerson (1991) for its broader context in game theory). Our contributions are as follows.

We first note that the total displacement of a parking function is invariant under the action of the permutation group, even when there are more spots than cars (Theorem 3.2). In turn, this allows us to define *parking games* as a class of (transferable utility) cooperative games in characteristic function form (Definition 3.3).

We then show that parking games are supermodular cost-sharing games (Lemma 3.5). As such, their *core* Shapley (1955); Gillies (1959) is typically empty, indicating that cooperation is difficult. Supermodularity arises whenever costs are exacerbated by "congestion" effects, such as in scheduling Goemans et al. (2002); Queyranne (1993); Schulz and Uhan (2010, 2013). In much the same way, parking games are supermodular because a car that arrives at an already busy street tends to be displaced significantly before it finds an unoccupied spot.

Next, we adopt the *Shapley value* Shapley (1953) as a notion of "fair" cost-sharing. In parking games, the Shapley value amounts to charging each car for its *expected marginal displacement* assuming its

arrival order is determined uniformly at random. However, a simple computation of this quantity requires exponential time. Therefore, our main contribution is a polynomial-time algorithm to compute the Shapley value of parking games (Theorem 3.10). Our algorithm leverages the permutation-invariance of total displacement, combinatorial enumeration, and dynamic programming. We contrast this positive result for the special case of parking games with known hardness results on computing the Shapley value for arbitrary cooperative games Deng and Papadimitriou (1994); Faigle and Kern (1992).

Finally, we note that unlike the *scheduling games* studied by Schulz and Uhan (2010), which are similar to parking games in that they are supermodular and their Shapley value can be computed in polynomial time, the Shapley value of parking games is not a *least core* allocation (i.e., a cost-share distribution that minimizes the worst-case dissatisfaction from cooperation) (Lemma 4.1). We conclude with open questions around the least core and *least core value* of parking games.

1.2 Organization

The remainder of this paper is organized as follows. In Section 2, we present some necessary background on parking functions and cooperative game theory. In Section 3, we define parking games, establish supermodularity, and introduce our algorithm. We conclude in Section 4 with some additional properties and open questions.

2 Background

Notation

We briefly outline some notational conventions. All tuples considered have positive integer entries and are denoted in boldface, as in $\beta = (b_1, b_2, \ldots, b_n) \in \mathbb{N}^n$. Throughout, weakly increasing tuples are furthermore decorated with an apostrophe, as in $\beta' = (b'_1, b'_2, \ldots, b'_n)$. If $\beta = (b_1, b_2, \ldots, b_n)$ is an n-tuple and $i \in [n]$, then $|\beta| = n$ denotes its size and

$$\beta_{\hat{i}} = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$$

denotes the $(n-1)$-tuple obtained from β upon the removal of its ith entry. For any $n \in \mathbb{N}$, let $\mathfrak{S}_n$ denote the symmetric group over $[n]$. If $\pi \in \mathfrak{S}_n$, then π^{-1} denotes its inverse.

2.1 Parking Functions

Let $\mathrm{PF}_n \subseteq [n]^n$ denote the set of parking functions of length n. Similarly, let $\mathrm{PF}_n^{\uparrow} \subseteq \mathrm{PF}_n$ denote the set of weakly increasing parking functions of length n. Konheim and Weiss (1966) showed that $|\mathrm{PF}_n| = (n+1)^{n-1}$ (OEIS A000272). We refer the reader to Riordan (1969) for an elegant proof credited to Pollak. Stanley (1997) showed, through a connection to noncrossing partitions, that $|\mathrm{PF}_n^{\uparrow}| = C_n$, where $C_n = \frac{1}{n+1}\binom{2n}{n}$ is the nth Catalan number (OEIS A000108).

Let $\alpha = (a_1, a_2, \ldots, a_n) \in [n]^n$ and $\alpha' = (a'_1, a'_2, \ldots, a'_n)$ be its weakly increasing rearrangement. It is well-known (for instance, refer to Yan (2015)) that $\alpha \in \mathrm{PF}_n$ if and only if $a'_i \leq i$ for all $i \in [n]$. As a consequence, PF_n consists of the orbits of the elements of $\mathrm{PF}_n^{\uparrow}$ under the action of $\mathfrak{S}_n$ (which permutes the subscripts). In particular, PF_n is closed under the action of permutations. Later in this work, we use the following definition. For any fixed $\alpha \in \mathrm{PF}_n$, let $r : [n] \to [n]$ be a bijective rank function that maps each index i in α to its index $r(i)$ in α' with ties broken arbitrarily, so that $a_i = a'_{r(i)}$. Given any subset $S \subseteq [n]$, we denote $r(S) = \{r(i) : i \in S\}$.

Konheim and Weiss (1966) also considered parking functions where there are more parking spots than cars. Let $n, m \in \mathbb{N}$ with $m \geq n$. Suppose there are m numbered parking spots on the one-way street and n cars arrive one at a time, each with a preferred spot in $[m]$. Let $\alpha = (a_1, a_2, \ldots, a_n) \in [m]^n$ be the n-tuple encoding the parking preferences of every car. Each car i follows the same parking rule as before: it parks in the first unoccupied spot at or past its preferred spot $a_i \in [m]$, if one exists. Let $\mathrm{PF}_{n,m} \subseteq [m]^n$ denote the set of preference tuples with n cars and m spots under which all cars are able to park; we refer to these as (n, m)-parking functions. (Konheim and Weiss, 1966, Lemma 2) showed that $|\mathrm{PF}_{n,m}| = (m+1)^{n-1}(m+1-n)$. Similarly, let $\mathrm{PF}^{\uparrow}_{n,m} \subseteq \mathrm{PF}_{n,m}$ denote the set of weakly increasing (n, m)-parking functions.

2.2 Shapley Value and Core

Let $n \in \mathbb{N}$. A (transferable utility) *coalitional cost-sharing game* over a set of players $[n]$ is specified by a *characteristic function* $c : 2^{[n]} \to \mathbb{R}$ satisfying $c(\emptyset) = 0$. Here, $c(S)$ is the collective cost incurred by the members of the coalition $S \subseteq [n]$ should they act in unison. The set $[n]$ is referred to as the *grand coalition*. In simpler terms, the assumption of transferable utility is the existence of a tradable commodity (e.g., money), assumed to be subject to identical valuation from each player, so that the single number $c(S)$ suffices to capture the feasible cost-share allocations for the members of a coalition $S \subseteq [n]$. A *solution concept* is a function ϕ that associates with each game c a subset of player cost-share allocations. This theory assumes coalitions can reach binding agreements. Even in scenarios where cooperation may be undesirable for individual agents, as Schulz and Uhan (2010) put it, it might be in the interest of an external party such as a government authority to encourage/enforce cooperation (e.g., through contracts, monetary penalties) to alleviate the negative of externalities of failure to cooperate.

Shapley (1953) (a 2012 Nobel laureate for contributions to game theory Roth (2016)) considered solution concepts in cost-sharing games through an axiomatic approach. In particular, he posited the following as desirable properties of a solution concept.

Axiom 2.1 (Efficiency) $\sum_{i=1}^{n} \phi_i(c) = c([n])$.

Axiom 2.2 (Nullity) *If* $c(S + \{i\}) = c(S)$ *for every* $S \subseteq [n] \setminus \{i\}$*, then* $\phi_i(c) = 0$.

Axiom 2.3 (Symmetry) *If* $c(S \cup \{i\}) = c(S + \{j\})$ *for every* $S \subseteq [n] \setminus \{i, j\}$*, then* $\phi_i(c) = \phi_j(c)$.

Axiom 2.4 (Additivity) *If* c' *is another characteristic function over the set of players* $[n]$*, then*

$$\phi_i\left(c + c'\right) = \phi_i(c) + \phi_i(c')$$

for each $i \in [n]$.

Axiom 2.1 states that the sum of cost-shares recover the total cost incurred by the grand coalition. Axiom 2.2 states that if player i never increases the cost of cooperation, in the sense that the marginal cost $c(S + \{i\}) - c(S)$ is zero for all coalitions $S \subseteq [n] \setminus \{i\}$ that player i could join, then player i is correspondingly charged nothing. Axiom 2.3 states that if players i and j are equivalent with respect to c, in the sense that $c(S + \{i\}) = c(S + \{j\})$ for all coalitions $S \subseteq [n] \setminus \{i, j\}$ that players i and j could join (so that all of such coalitions are ambivalent between which of the two players joins them), then players i and j are correspondingly charged the same. Lastly, Axiom 2.4 captures the idea that, if the players participate in a game that combines two independent (possibly completely different) games, then the outcome of one the games does not affect the outcome of the other.

Shapley showed that there is a unique solution concept satisfying Axioms 2.1-2.4. This solution concept is now known as the *Shapley value*.

Theorem 2.5 (Shapley (1953)) *Let $\mathcal{C}$ be the set of characteristic functions over the set of players $[n]$. The function $\phi : \mathcal{C} \to \mathbb{R}^n$ given by*

$$\phi_i(c) = \frac{1}{n!} \sum_{\pi \in \mathfrak{S}_n} c\left(\{j \in [n] : \pi(j) \leq \pi(i)\}\right) - c\left(\{j \in [n] : \pi(j) < \pi(i)\}\right) \tag{1}$$

for each $i \in [n]$ is the unique function satisfying Axioms 2.1-2.4.

In other words, in the Shapley value, each player is charged their *expected marginal cost* assuming the order in which they join the grand coalition is determined uniformly at random. Refer to (Myerson, 1991, Chapter 9.4) for an approachable presentation of Axioms 2.1-2.4 and of Theorem 2.5.

The Shapley value can be readily computed with exponentially-many oracle calls to the characteristic function (i.e., querying the value $c(S)$ for a given $S \subseteq [n]$), so it is natural to ask whether this many evaluation is generally necessary. (Faigle and Kern, 1992, Theorem 3) showed that any algorithm that computes the Shapley value for *arbitrary* characteristic functions requires exponentially-many calls. Moreover, (Deng and Papadimitriou, 1994, Theorem 9) showed that, in general, computing the Shapley value is $\#P$-complete. However, there are nontrivial special cases for which polynomial-time algorithms are known (e.g., (Deng and Papadimitriou, 1994, Theorem 1)). In this work, we show that parking games are another such special case.

Gillies (1959); Shapley (1955) introduced another solution concept, known as the *core*, with desirable "stability" properties. The core of a cooperative game c is the set

$$\left\{ \phi \in \mathbb{R}^n : \sum_{i=1}^{n} \phi_i = c([n]),\ \sum_{i \in S} \phi_i \leq c(S) \text{ for all } S \subseteq [n] \right\}.$$

Intuitively, it is the set of cost-share allocations that are simultaneously efficient (Axiom 2.1) and robust against coalitional defections. In other words, no coalition is charged more than the cost that they would incur by themselves.

The Bondareva-Shapley theorem Bondareva (1963); Shapley (1967) (refer also to (Peleg and Sudhölter, 2007, Chapter 3.1)) provides necessary and sufficient conditions for the non-emptiness of the core. In particular, it implies that if c is *submodular*, then the core is non-empty. Note that c is submodular if

$$c(S \cup \{i\}) - c(S) \geq c(T \cup \{i\}) - c(T)$$

for all $i \in [n]$ and all $S \subseteq T \subseteq [n] \setminus \{i\}$. Conversely, it is *supermodular* if

$$c(S \cup \{i\}) - c(S) \leq c(T \cup \{i\}) - c(T)$$

for all $i \in [n]$ and all $S \subseteq T \subseteq [n] \setminus \{i\}$. It is *modular* if it is simultaneously submodular and supermodular. Intuitively, submodularity and supermodularity capture the notions of decreasing and increasing marginal costs, respectively. It can be verified that the core of a supermodular game is empty unless it is modular, indicating that cooperation is difficult.

3 Parking Games

3.1 Displacement, Total Displacement, and Arrival Order

Fix any $\alpha = (a_1, a_2, \ldots, a_n) \in \mathrm{PF}_n$. Then, by definition, car i with preference $a_i \in [n]$ parks in some spot $p_i \in [n]$ with $p_i \geq a_i$. Therefore, the displacement incurred by car i upon its arrival under α is given by

$$p_i - a_i \geq 0. \tag{2}$$

Now, let $d : \mathrm{PF}_n \to \mathbb{N}$ be the *total displacement* function, where $d(\alpha)$ is the total displacement incurred by all cars upon their arrival with preferences $\alpha \in \mathrm{PF}_n$. Using Equation (2), for any fixed $\alpha \in \mathrm{PF}_n$ we have

$$d(\alpha) \coloneqq \sum_{i=1}^{n} p_i - a_i. \tag{3}$$

For example, given a weakly increasing parking function $\alpha' = (a'_1, a'_2, \ldots, a'_n) \in \mathrm{PF}_n^{\uparrow}$, the cars $1, 2, \ldots, n$ park in the order $1, 2, \ldots, n$. Therefore, in this case we have $d(\alpha') = \sum_{i=1}^{n}(i - a'_i)$.

We show that the total displacement statistic is invariant under rearrangement of the entries of a parking function. This result is standard (e.g., it is stated in (Yan, 2015, Chapter 13.2.2)), but we formally state and prove it for completeness and later use.

Lemma 3.1 *If $\alpha = (a_1, a_2, \ldots, a_n) \in PF_n$ and $\pi \in \mathfrak{S}_n$ acts on α by permuting its subscripts, i.e. $\pi(\alpha) = (a_{\pi^{-1}(1)}, a_{\pi^{-1}(2)}, \ldots, a_{\pi^{-1}(n)})$, then $d(\alpha) = d(\pi(\alpha))$.*

Proof: Note that under α, car i with preference $a_i \in [n]$ parks in some spot $p_i \in [n]$. Since $\alpha \in \mathrm{PF}_n$, this implies $\{p_1, p_2, \ldots, p_n\} = [n]$ and we have

$$d(\alpha) = \sum_{i=1}^{n} (p_i - a_i) = \sum_{i=1}^{n} i - \sum_{i=1}^{n} a_i = \frac{n(n+1)}{2} - \sum_{i=1}^{n} a_i.$$

Similarly, note that under $\pi(\alpha)$, car i with preference $a_{\pi^{-1}(i)}$ parks in some spot $q_i \in [n]$. Since $\pi(\alpha) \in \mathrm{PF}_n$, this implies $\{q_1, q_2, \ldots, q_n\} = [n]$ and we have

$$d(\pi(\alpha)) = \sum_{i=1}^{n} \left(q_i - a_{\pi^{-1}(i)}\right) = \sum_{i=1}^{n} i - \sum_{i=1}^{n} a_{\pi^{-1}(i)} = \frac{n(n+1)}{2} - \sum_{i=1}^{n} a_{\pi^{-1}(i)}.$$

Lastly, note that

$$\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} a_{\pi^{-1}(i)}$$

since π is a bijection from $[n]$ to $[n]$. □

We now generalize this result to the case of (n, m)-parking functions. For $n, m \in \mathbb{N}$ with $m \geq n$, let $d : \mathrm{PF}_{n,m} \to \mathbb{N}$ be the total displacement function, where $d(\alpha)$ is the total displacement incurred by all cars upon their arrival with preferences $\alpha \in \mathrm{PF}_{n,m}$.

Theorem 3.2 *If $\alpha = (a_1, a_2, \ldots, a_n) \in PF_{n,m}$ and $\pi \in \mathfrak{S}_n$ acts on α by permuting its subscripts, i.e. $\pi(\alpha) = (a_{\pi^{-1}(1)}, a_{\pi^{-1}(2)}, \ldots, a_{\pi^{-1}(n)})$, then $d(\alpha) = d(\pi(\alpha))$.*

Proof: Upon replicating the proof technique from Lemma 3.1, it suffices to show that the subset of occupied spots under α is the same as the subset of occupied spots under $\pi(\alpha)$ (this is immediate in Lemma 3.1 since it assumes $\alpha, \pi(\alpha) \in \text{PF}_n$).

Suppose that under α, the cars park in spots in $S \subseteq [m]$ whereas under $\pi(\alpha)$, the cars park in spots in $T \subseteq [m]$. Note that $|S| = |T| = n$. If $S = T$, then the result follows. Suppose by way of contradiction that $S \neq T$, and let p be the smallest value in $(S \setminus T) \cup (T \setminus S)$. Without loss of generality, suppose $p \in S \setminus T$ (otherwise we swap the roles in the following argument). Since $p \in S$, there are k cars with preference less than or equal to p in α for some $1 \leq k \leq n$. Conversely, since $p \notin T$ and p is the smallest value in $S \setminus T$, there are $k-1$ cars with preference less than or equal to p in $\pi(\alpha)$. This is a contradiction, for α and $\pi(\alpha)$ are equal as multisets. □

Given Theorem 3.2, we now define parking games as coalitional games in characteristic function form, where each car is treated as a player.

Definition 3.3 (Parking Game) *Let $\alpha = (a_1, a_2, \ldots, a_n) \in PF_n$ be a parking function. The parking game of α is given by the characteristic function $c_\alpha : 2^{[n]} \to \mathbb{N}$, where*

$$c_\alpha(S) = d((a_i : i \in S)) \tag{4}$$

for each $S \subseteq [n]$.

In other words, $c_\alpha(S)$ is the total displacement of the $(|S|, n)$-parking function $(a_i : i \in S) \in \text{PF}_{|S|,n}$. Note that this definition relies on Theorem 3.2 to treat c_α as a set function that is independent of the arrival order of the cars in S. As noted in the following remark, the parking game of a parking function can be obtained through the parking game of its weakly increasing rearrangement.

Remark 3.4 *Let $\alpha = (a_1, a_2, \ldots, a_n) \in PF_n$ be a parking function and $\alpha' = (a'_1, a'_2, \ldots, a'_n) \in PF_n^{\uparrow}$ be its weakly increasing rearrangement. Then, by Theorem 3.2, it follows that*

$$c_\alpha(S) = d((a_i : i \in S)) = d((a'_i : i \in r(S))) = c_{\alpha'}(r(S))$$

for each $S \subseteq [n]$.

3.2 Supermodularity

We now show that parking games are supermodular cost-sharing games.

Lemma 3.5 *Let $\alpha = (a_1, a_2, \ldots, a_n) \in PF_n$ be a parking function. Then, c_α is supermodular, i.e.,*

$$c_\alpha(S \cup \{i\}) - c_\alpha(S) \leq c_\alpha(T \cup \{i\}) - c_\alpha(T)$$

for all $i \in [n]$ and $S \subseteq T \subseteq [n] \setminus \{i\}$.

Proof: Fix any $i \in [n]$ and $S \subseteq T \subseteq [n] \setminus \{i\}$. Consider the arrival of car i immediately after the arrival of the cars in S. If spot a_i is empty, then $c_\alpha(S \cup \{i\}) = c_\alpha(S)$ and the inequality is verified. Conversely, suppose there is a sequence of contiguously occupied spots starting at some spot s satisfying $1 \leq s \leq a_i$ and ending at some spot t satisfying $a_i \leq t \leq n-1$, implying the displacement of car i is $t - a_i + 1$. Now, similarly consider the arrival of car i immediately after the arrival of the cars in $T \supseteq S$. Then, there is again a sequence of contiguously occupied spots, except this time starting at some spot s' satisfying

$1 \leq s' \leq s$ and ending at some spot t' satisfying $t \leq t' \leq n-1$, implying the displacement of car i is $t' - a_i + 1 \geq t - a_i + 1$. □

Supermodularity captures the notion of increasing marginal costs, and it arises whenever costs are exacerbated by "congestion" effects, such as in scheduling Goemans et al. (2002); Queyranne (1993); Schulz and Uhan (2010, 2013). Intuitively, parking games are supermodular because a car that arrives at an already busy street tends to displace significantly before it finds an unoccupied spot. (Schulz and Uhan, 2010, Theorem 1) showed that supermodular games are in some sense common: the problem of minimizing a non-negative linear function over a supermodular polyhedron, which arises often in combinatorial optimization, has a supermodular objective value. In effect, parking games are a special family within the broader class of supermodular games.

It follows from Lemma 3.5 that the core of parking games is typically empty (with the exception of modular cases in which $\alpha \in \mathfrak{S}_n$). For example, this can be verified from the fact that each car in isolation incurs no displacement.

3.3 Expected Marginal Displacement

Let $\alpha \in \mathrm{PF}_n$ be a parking function and $c_\alpha : 2^{[n]} \to \mathbb{N}$ be its parking game. In this section, we present a polynomial-time algorithm to compute the Shapley value of c_α.

In this setting, based on Equation (1), each car i is assigned a cost-share of

$$\phi_i(c_\alpha) = \frac{1}{n!} \sum_{\pi \in \mathfrak{S}_n} c_\alpha\left(\{j \in [n] : \pi(j) \leq \pi(i)\}\right) - c_\alpha\left(\{j \in [n] : \pi(j) < \pi(i)\}\right). \tag{5}$$

Note that given any arrival order $\pi \in \mathfrak{S}_n$, the difference

$$c_\alpha\left(\{j \in [n] : \pi(j) \leq \pi(i)\}\right) - c_\alpha\left(\{j \in [n] : \pi(j) < \pi(i)\}\right)$$

corresponds to the displacement of car i upon its arrival in the order given by π. In effect, Equation (5) is a formula for the *expected marginal displacement* of car i assuming its arrival order is determined uniformly at random. We leverage this interpretation in what follows.

Example 3.6 *Let $\alpha = (1,4,3,3,1,2,7) \in PF_7$. Based on* (3) *we have that $d(\alpha) = 7$. Based on* (5) *we have that:*

- $\phi_1(c_\alpha) = 7896/7! = 47/30$.
- $\phi_2(c_\alpha) = 2856/7! = 17/30$.
- $\phi_3(c_\alpha) = 5628/7! = 67/60$.
- $\phi_4(c_\alpha) = 5628/7! = 67/60$.
- $\phi_5(c_\alpha) = 7896/7! = 47/30$.
- $\phi_6(c_\alpha) = 5376/7! = 16/15$.
- $\phi_7(c_\alpha) = 0/7! = 0$.

Note that $\sum_{i=1}^{n} \phi_i(c_\alpha) = 7$ (this reflects Axiom 2.1). Moreover, note that cars with the same preference have the same cost-share allocation, such as cars 3 and 4 (this reflects Axiom 2.3). Lastly, note that car 7 is lucky (i.e., parks in its preferred spot) regardless of the arrival order. Therefore, $\phi_7 = 0$ (this reflects Axiom 2.2).

We first introduce some notation. Given a tuple β and an integer $t \in [n]$, let

$$\Lambda_t(\beta) = |\{b \in \beta : b \geq t+2\}|$$

be the number of entries of β that are greater than or equal to $t+2$. Similarly, given an integer $s \in [n]$, let

$$\Gamma_s(\beta) = |\{b \in \beta : b \leq s-2\}|$$

be the number of entries of z that are less than or equal to $s-2$. Moreover, given a weakly increasing tuple β', and three nonnegative integers $s, t, k \in \mathbb{N}$ with $s \leq t$, let $\mathcal{Q}(\beta', s, t, k)$ denote the number of size k sub-tuples of β' that, when treated as a preference k-tuple, cars park in spots $s, \ldots, t$. Formally,

$$\mathcal{Q}(\beta', s, t, k) = |\{(i_1, i_2, \ldots, i_k) : (b'_{i_1} - b'_{i_1} + 1, b'_{i_2} - b'_{i_1} + 1, \ldots, b'_{i_k} - b'_{i_1} + 1) \in \mathrm{PF}^{\uparrow}_{k,t-s+1}\}|.$$

We obtain the following enumeration.

Lemma 3.7 *Let $\alpha = (a_1, a_2, \ldots, a_n) \in PF_n$ be a parking function and $\alpha' = (a'_1, a'_2, \ldots, a'_n) \in PF^{\uparrow}_n$ be its weakly increasing rearrangement. Fix any car $j \in [n]$ and let $i \coloneqq r(j) \in [n]$ be its rank. Then, Equation* (5) *for car j is given by*

$$\phi_j(c_\alpha) = \phi_i(c_{\alpha'}) = \frac{1}{n!} \sum_{s=1}^{a'_i} \sum_{t=a'_i}^{n-1} (t - a'_i + 1)\mathcal{Q}(\alpha'_i, s, t, t-s+1)\mathcal{R}(\alpha'_i, s, t), \tag{6}$$

where

$$\mathcal{R}(\alpha'_i, s, t) = \sum_{\lambda=0}^{\Lambda_t(\alpha'_i)} \sum_{\gamma=0}^{\Gamma_s(\alpha'_i)} \binom{\Lambda_t(\alpha'_i)}{\lambda} \mathcal{Q}(\alpha'_i, 1, s-2, \gamma)(t-s+1+\lambda+\gamma)!(n-t+s-\lambda-\gamma-2)!.$$

Proof: Note that the cost-share $\phi_j(c_\alpha)$ of car j under α is equal to the cost-share $\phi_i(c_{\alpha'})$ of car i under α'. Therefore, consider the displacement of car i upon its arrival under α'.

Upon its arrival, car i drives to its preferred spot a'_i. If spot a'_i is empty, car i parks there incurring zero displacement. Conversely, suppose spot a'_i is occupied. Then, there is a sequence of contiguously occupied spots starting at some spot s satisfying $1 \leq s \leq a'_i$ and ending at some spot t satisfying $a'_i \leq t \leq n-1$; note that $t \neq n$ since, by assumption, car i is able to park. Now, fix any pair of possible values for s and t. This implies the following (for otherwise the block does not start at s and end at t):

- Spot $t+1$ is empty. As a consequence, car i parks in spot $t+1$ incurring $t - a'_i + 1$ displacement.
- If spot $s-1$ exists, it is empty as well.

In turn, the choice of s and t leads to three contiguous segments of parking spots in which cars may park prior to the arrival of car i:

1. Spots $1, \ldots, s-2$ (if these exist),
2. spots $s, \ldots, t$, and
3. spots $t+2, \ldots, n$ (if these exist).

Note that unlike spots $s, \ldots, t$, spots $1, \ldots, s-2$ need not be contiguously occupied. Similarly, spots $t+2, \ldots, n$ need not be contiguously occupied. We now count the number of different subsets of cars that can occupy each of these three segments.

1. Since spot $s-1$ is empty, only those cars with preferences less than or equal to $s-2$ could possibly park in spots $1, \ldots, s-2$; there are $\Gamma_s(\alpha_{\hat{i}})$ such cars. Only some number $0 \leq \gamma \leq \Gamma_s(\alpha_{\hat{i}})$ of cars park in these spots prior to the arrival of car i. Therefore, for any fixed possible value of γ, the number of size-γ subset of cars that park in this segment is given by $Q(\alpha'_{\hat{i}}, 1, s-2, \gamma)$.
2. Since spot $s-1$ is empty, spots $s, \ldots, t$ are contiguously occupied, and spot $t+1$ is empty, we need the number of size-$(t-s+1)$ subsets of cars that park in this segment. This is given by $Q(\alpha'_{\hat{i}}, s, t, t-s+1)$.
3. Since spot $t+1$ is empty, only those cars with preferences greater than or equal to $t+2$ could possibly park in spots $t+2, \ldots, n$; there are $\Lambda_t(\alpha_{\hat{i}})$ such cars. Only some number $0 \leq \lambda \leq \Lambda_t(\alpha_{\hat{i}})$ of cars park in these spots prior to the arrival of car i. Therefore, for any fixed possible value of λ, the number of size-λ subset of cars that park in this segment is given by $Q(\alpha'_{\hat{i}}, t+2, n, \lambda)$. In fact, since $\alpha' \in \mathrm{PF}_n^{\uparrow}$, any size-$\lambda$ subset of cars with preference greater than or equal to $t+2$ is able to park in the segment $t+2, \ldots, n$, and it follows that

$$Q(\alpha'_{\hat{i}}, t+2, n, \lambda) = \binom{\Lambda_t(\alpha_{\hat{i}})}{\lambda}.$$

Note that in (1) above, the term $Q(\alpha'_{\hat{i}}, 1, s-2, \gamma)$ does not generally simplify into a binomial because, depending on the preferences, certain cars might displace into spot $s-1$, which we thus far assume to be empty.

Now, the choice of s, t, λ, and γ imply $(t-s+1)+\lambda+\gamma$ cars park prior to the arrival of car i, in one of $(t-s+1+\lambda+\gamma)!$ different arrival orders. Similarly, $n-1-((t-s+1)+\lambda+\gamma) = n-t+s-\lambda-\gamma-2$ cars arrive after the arrival of car i, in one of $(n-t+s-\lambda-\gamma-2)!$ different arrival orders.

Summing over the possible values for s, t, λ, and γ yields the sum portion of Equation 6:

$$\sum_{s=1}^{a'_i} \sum_{t=a'_i}^{n-1} (t-a'_i+1)\mathcal{Q}(\alpha'_{\hat{i}}, s, t, t-s+1)\mathcal{R}(\alpha'_{\hat{i}}, s, t)$$

where

$$\mathcal{R}(\alpha'_{\hat{i}}, s, t) = \sum_{\lambda=0}^{\Lambda_t(\alpha'_{\hat{i}})} \sum_{\gamma=0}^{\Gamma_s(\alpha'_{\hat{i}})} \binom{\Lambda_t(\alpha'_{\hat{i}})}{\lambda} \mathcal{Q}(\alpha'_{\hat{i}}, 1, s-2, \gamma)(t-s+1+\lambda+\gamma)!(n-t+s-\lambda-\gamma-2)!$$

Finally, there are $n!$ different orders in which all cars could arrive, and the arrival order is realized uniformly at random. This completes the proof. □

Example 3.8 *We demonstrate an execution of* (6) *using the instance in Example 3.6. Let*

$$\alpha = (1, 4, 3, 3, 1, 2, 7) \in PF_7$$

and note that its weakly increasing rearrangement is

$$\alpha' = (1, 1, 2, 3, 3, 4, 7) \in PF_7^{\uparrow}.$$

Consider $j = 2$, so that $\alpha_2 = 4$ and $r(j) = 6$. To compute $\phi_2(\alpha)$ we equivalently compute $\phi_6(\alpha')$. Here we only show one term of its computation.

Suppose $s = 2$ and $t = 4$ in (6)*. Then, upon its arrival, car 6 is displaced $t - a'_6 + 1 = 4 - 4 + 1 = 1$ unit. Now, by our choice of s and t, spots $2, 3, 4$ are contiguously occupied whereas spot 1 and 5 are unoccupied. Note the following:*

- *Neither of cars 1 or 2 could have arrived, for otherwise spot 1 would be occupied. In particular, $\Gamma_s(\alpha'_{\hat{6}}) = \emptyset$ and we only need to consider $\gamma = 0$.*
- *Car 7 could have arrived, for in either case spot 5 would remain unoccupied. In particular, $\Lambda_t(\alpha'_{\hat{6}}) = \{7\}$ and we only need to consider $\lambda = 0, 1$.*
- *Cars $3, 4, 5$ must have arrived, as this is the only way in which spots $2, 3, 4$ would be contiguously occupied. In particular, $\mathcal{Q}(\alpha'_{\hat{6}}, s, t, t - s + 1) = 1$.*

In the case in which $\gamma = 0, \lambda = 0$, a total of 3 cars arrive prior to car 6 whereas a total of 3 cars arrive after car 6. In the case in which $\gamma = 0, \lambda = 1$, a total of 4 cars arrive prior to car 6 whereas a total of 2 cars arrive after car 6. To summarize, this choice of s and t contributes

$$\underbrace{1}_{t-a'_6+1}\ \underbrace{1}_{\mathcal{Q}(\alpha'_{\hat{6}},s,t,t-s+1)}\left(\underbrace{1\cdot 1\cdot 3!\cdot 3!}_{\gamma=0,\gamma=0}+\underbrace{1\cdot 1\cdot 4!\cdot 2!}_{\lambda=1,\gamma=0}\right)=84$$

to the total sum.

Assuming oracle access to $\mathcal{Q}$, Equation (6) can be evaluated in polynomial time in α for any car $j \in [n]$ (note that each of the summations involves at most n terms). As a final step, we show that $\mathcal{Q}$ can be evaluated in polynomial time as well.

We first introduce a some additional notation. Given a tuple β, let $b^* = \min\{b \in \beta\}$ be the value of its smallest entry and

$$U(\beta) = (\max\{b, 1 + b^*\} : b \in \beta)$$

be the copy of β in which all entries with value equal to b^* are increased by one. For example, if $\beta = (3, 3, 4, 4, 5)$, then $b^* = 3$ and $U(\beta) = (4, 4, 4, 4, 5)$. We evaluate $\mathcal{Q}$ using the following recursive relation.

Lemma 3.9 *Let $\beta' = (b'_1, b'_2, \ldots, b'_{|\beta'|}) \in [w]^{|\beta'|}$ be a nonnegative, weakly increasing tuple where $w \in \mathbb{N}$, and let $s, t, k \in \mathbb{N}$ with $s \leq t$. Then, $\mathcal{Q}(\beta', s, t, k)$ satisfies the following recursive relation:*

$$\mathcal{Q}(\beta', s, t, k) = \begin{cases} 1, & \text{if } k = 0, \\ 0, & \text{if } k > t - s + 1 \vee k > |\beta'|, \\ \mathcal{Q}(\beta'_{\hat{1}}, s, t, k), & \text{if } b'_1 < s, \\ \mathcal{Q}(\beta', s+1, t, k), & \text{if } b'_1 > s, \\ \mathcal{Q}(U(\beta')_{\hat{1}}, s+1, t, k-1) + \mathcal{Q}(\beta'_{\hat{1}}, s, t, k), & \text{if } b'_1 = s. \end{cases} \tag{7}$$

Using dynamic programming, $\mathcal{Q}(\beta', s, t, k)$ can be evaluated in time polynomial in $|\beta'|$ and k.

Proof: We first consider the base case. If we attempt to park more cars than there are spots (i.e., $k > t - s + 1$), or more cars than there are preferences (i.e., $k > |\beta'|$), the count is zero. Similarly, if we attempt to park no cars (i.e., $k = 0$), the count is one.

Therefore, suppose $1 \leq k \leq \min\{t-s+1, |\beta'|\}$. In this case, there are three possibilities, each leading to a distinct recursive call (recall b'_1 is the first entry of β'):

- If $b'_1 < s$, then the first car cannot park in the segment $s, \ldots, t$. Therefore, in this case, the count is the same as the count $\mathcal{Q}(\beta'_{\hat{1}}, s, t, k)$ upon the removal of the first car.
- If $b'_1 = s$, then there are two possibilities for the first car: it is selected as part of the size-k subset of cars, or it is not. In the first sub-case, no subsequent car can park in the spot s that starts the segment $s, \ldots, t$. Therefore, in this sub-case, the count is the same as the count $\mathcal{Q}(U(\beta')_{\hat{1}}, s + 1, t, k - 1)$ upon increasing the preference of any car that prefers spot s by one, removing the first car, increasing the segment start by one spot, and decreasing the number of cars to select by one. In the second sub-case, the count is the same as the count $\mathcal{Q}(\beta'_{\hat{1}}, s, t, k)$ upon the removal of the first car.
- If $b'_1 > s$, then the first car cannot park in the spot s that starts the segment $s, \ldots, t$. Therefore, in this case, the count is the same as the count $\mathcal{Q}(\beta', s + 1, t, k)$ upon increasing the segment start by one spot.

Finally, note that a dynamic programming table of polynomial size can be implemented since its indices require at most $|\beta'|$ suffixes of β', $s, t \leq |\beta'|$, and U need only be applied k-many times. □

While w does not play any explicit role in the proof of Lemma 3.9, the values it can possibly take are implicitly restricted by the problem definition which, for any given $n \in \mathbb{N}$, is restricted to $\mathrm{PF}_n^{\uparrow} \subseteq [n]^n$. We thus obtain our main result.

Theorem 3.10 *There exists a polynomial-time algorithm to compute the Shapley value of parking games. Namely, the one given by the formulas in Lemma 3.7 and Lemma 3.9.*

Proof: The correctness of the algorithm follows from the proofs of the lemmas. Its running time follows from Lemma 3.9 and the fact that, in Equation (6), the recursive relation Equation (7) is always evaluated passing weakly increasing β' and k satisfying $|\beta'|, k \leq n$. □

4 Conclusion

One can envision real-world systems in which displacement is a costly negative externality and, as a result, the parking rate at a given spot is a function of the displacement derived from its popularity. For example, in transportation settings, this might be because of increased environmental emissions. The results presented in this work form a methodological basis for operating such a system.

The results in this paper readily extend to (n, m)-parking functions by splitting them into small independent "parking functions," in the style underlying Theorem 3.2. Moreover, because of Axiom 2.4, Theorem 3.10 can be applied given any characteristic function that is an affine function of the total displacement.

We conclude with directions for future research. As noted in Lemma 3.5, parking games are supermodular, which indicates that their core is empty (except for modular cases in which $\alpha \in \mathfrak{S}_n$). Therefore, the *least core* Shapley and Shubik (1966); Maschler et al. (1979) is an appropriate alternative solution concept. The least core is the set of optimal solutions to

$$z^* = \min \left\{ z : \sum_{i=1}^{n} \phi_i = c([n]),\ \sum_{i \in S} \phi_i \leq c(S) + z,\ \text{for all } S \subseteq [n] \right\}, \tag{8}$$

and z^* is the *least core value*. Unlike the core, the least core is always non-empty. A least core allocation minimizes the worst-case dissatisfaction over all coalitions, and the least core value can be seen as the minimum amount that needs to be charged for defection (e.g., by a governing authority) in order to incentivize cooperation.

Schulz and Uhan (2010, 2013) study the least core and least core value of supermodular cost-sharing games. They show that computing the least core value of an arbitrary supermodular game is strongly NP-hard (Schulz and Uhan, 2010, Theorem 2). Interestingly, for *scheduling games*, not only can their Shapley value be computed in polynomial time, but it moreover happens to be a least core allocation (Schulz and Uhan, 2010, Theorem 3). (However, they note that computing the least core value of scheduling games remains weakly NP-hard). Parking games are similar to scheduling games in that, as we have shown, they are supermodular and their Shapley value can be computed in polynomial time. However, they are different in that their Shapley value is not a least core allocation.

Lemma 4.1 *The Shapley value of parking games is not a least core allocation.*

Proof: It suffices to show a counterexample. Let $\alpha = (1, 1, 2) \in \text{PF}_3$ and consider its parking game c_α. We have $c_\alpha(\emptyset) = c_\alpha(\{1\}) = c_\alpha(\{2\}) = c_\alpha(\{3\}) = c_\alpha(\{1, 3\}) = c_\alpha(\{2, 3\}) = 0$, $c_\alpha(\{1, 2\}) = 1$, and $c_\alpha(\{1, 2, 3\}) = 2$. Its Shapley value is $\phi_1(c_\alpha) = \phi_2(c_\alpha) = 5/6$ and $\phi_3(c_\alpha) = 2/6$. However, its least core value is $z^* = 1$ with the least core allocation $\phi_1 = \phi_2 = 1$ and $\phi_3 = 0$. In particular, for $S = \{1, 3\}$ we have

$$\phi_1(c_\alpha) + \phi_3(c_\alpha) = 5/6 + 2/6 = 7/6 \not\leq 1 = 0 + 1 = c_\alpha(\{1, 3\}) + z^*.$$

□

However, given the positive result on the complexity of computing the Shapley value of parking games, we ask whether a least core allocation or the least core value of parking games can be computed in polynomial time and/or interpreted combinatorially. First, we ask whether the least core allocation(s) relate to individual parking statistics, including but not limited to those concerning individual displacement. We

also ask about the coalitions $S \subseteq [n]$ for which the least core value z^* is attained in (8) for a particular least core allocation: do such coalitions fully determine z^* in terms of parking statistics that distinguish them from other coalitions? Similarly, we ask whether the Shapley value of parking games is an *approximate* least core allocation (refer to (Schulz and Uhan, 2013, Section 2)).

Finally, given the abundance of connections between parking functions and other combinatorial objects (refer to Section 1 for some examples), future work might consider mappings that preserve the total displacement statistic, in this way defining equivalent cooperative games except with a different combinatorial interpretation.

Acknowledgements

J. C. Martínez Mori would like to thank A. Toriello for a helpful discussion.

References

A. Amanbayeva and D. Wang. The convex hull of parking functions of length n. *Enumer. Comb. Appl.*, 2 (2):Paper No. S2R10, 10, 2022. doi: 10.54550/eca2022v2s2r10.

D. Armstrong, N. A. Loehr, and G. S. Warrington. Rational parking functions and Catalan numbers. *Ann. Comb.*, 20(1):21–58, 2016. doi: 10.1007/s00026-015-0293-6.

O. N. Bondareva. Some applications of the methods of linear programming to the theory of cooperative games. *Problemy Kibernet.*, 10:119–139, 1963.

J. Carlson, A. Christensen, P. E. Harris, Z. Jones, and A. R. Rodriguez. Parking functions: Choose your own adventure. *The College Mathematics Journal*, 52(4):254–264, 2021.

R. Cori and D. Rossin. On the sandpile group of dual graphs. *European J. Combin.*, 21(4):447–459, 2000. doi: 10.1006/eujc.1999.0366.

X. T. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19(2):257–266, 1994. doi: 10.1287/moor.19.2.257.

P. Diaconis and A. Hicks. Probabilizing parking functions. *Adv. in Appl. Math.*, 89:125–155, 2017. doi: 10.1016/j.aam.2017.05.004.

J. Elder, P. E. Harris, J. Kretschmann, and J. C. Martínez Mori. Boolean intervals in the weak order of $\mathfrak{S}_n$. *arXiv preprint arXiv:2306.14734*, 2023.

U. Faigle and W. Kern. The Shapley value for cooperative games under precedence constraints. *Internat. J. Game Theory*, 21(3):249–266, 1992. doi: 10.1007/BF01258278.

D. B. Gillies. Solutions to general non-zero-sum games. In *Contributions to the theory of games, Vol. IV*, volume no. 40 of *Ann. of Math. Stud.*, pages 47–85. Princeton Univ. Press, Princeton, NJ, 1959.

M. X. Goemans, M. Queyranne, A. S. Schulz, M. Skutella, and Y. Wang. Single machine scheduling with release dates. *SIAM J. Discrete Math.*, 15(2):165–192, 2002. doi: 10.1137/S089548019936223X.

P. E. Harris, J. Kretschmann, and J. C. Martínez Mori. Lucky cars and the `quicksort` algorithm. *Amer. Math. Monthly*, 131(5):417–423, 2024. doi: 10.1080/00029890.2024.2309103.

D. E. Knuth. Linear probing and graphs. *Algorithmica*, 22(4):561–568, 1998. doi: 10.1007/PL00009240.

A. G. Konheim and B. Weiss. An occupancy discipline and applications. *SIAM Journal on Applied Mathematics*, 14(6):1266–1274, 1966.

G. Kreweras. Une famille de polynômes ayant plusieurs propriétés énumeratives. *Periodica Mathematica Hungarica*, 11(4):309–320, 1980.

J. C. Martínez Mori. What is...a parking function? *Notices Amer. Math. Soc.*, 71(8):1062–1065, 2024.

M. Maschler, B. Peleg, and L. S. Shapley. Geometric properties of the kernel, nucleolus, and related solution concepts. *Math. Oper. Res.*, 4(4):303–338, 1979. doi: 10.1287/moor.4.4.303.

R. B. Myerson. *Game theory: Analysis of conflict*. Harvard University Press, 1991.

B. Peleg and P. Sudhölter. *Introduction to the theory of cooperative games*, volume 34 of *Theory and Decision Library. Series C: Game Theory, Mathematical Programming and Operations Research*. Springer, Berlin, second edition, 2007. ISBN 978-3-540-72944-0.

R. Pyke. The supremum and infimum of the Poisson process. *Ann. Math. Statist.*, 30:568–576, 1959. doi: 10.1214/aoms/1177706269.

M. Queyranne. Structure of a simple scheduling polyhedron. *Math. Programming*, 58(2):263–285, 1993. doi: 10.1007/BF01581271.

J. Riordan. Ballots and trees. *J. Combinatorial Theory*, 6:408–411, 1969.

A. E. Roth. Lloyd shapley (1923–2016). *Nature*, 532(7598):178–178, 2016.

A. S. Schulz and N. A. Uhan. Sharing supermodular costs. *Oper. Res.*, 58(4):1051–1056, 2010. doi: 10.1287/opre.1100.0841.

A. S. Schulz and N. A. Uhan. Approximating the least core value and least core of cooperative games with supermodular costs. *Discrete Optim.*, 10(2):163–180, 2013. doi: 10.1016/j.disopt.2013.02.002.

L. S. Shapley. A value for n-person games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games, Volume II*, pages 307–318. Princeton University Press, Princeton, NJ, 1953.

L. S. Shapley. Markets as cooperative games. Technical report, RAND Corporation, Santa Monica, CA, 1955.

L. S. Shapley. On balanced sets and cores. *Naval Research Logistics Quarterly*, 14(4):453–460, 1967.

L. S. Shapley. Cores of convex games. *Internat. J. Game Theory*, 1:11–26; errata, ibid. 1 (1971/72), 199, 1971/72. doi: 10.1007/BF01753431.

L. S. Shapley and M. Shubik. Quasi-cores in a monetary economy with nonconvex preferences. *Econometrica: Journal of the Econometric Society*, pages 805–827, 1966.

R. P. Stanley. Hyperplane arrangements, interval orders, and trees. *Proc. Nat. Acad. Sci. U.S.A.*, 93(6): 2620–2625, 1996. doi: 10.1073/pnas.93.6.2620.

R. P. Stanley. Parking functions and noncrossing partitions. *Electron. J. Combin.*, 4 (The Wilf Festschrift volume)(2):Research Paper 20, 1997. doi: 10.37236/1335.

R. P. Stanley and J. Pitman. A polytope related to empirical distributions, plane trees, parking functions, and the associahedron. *Discrete Comput. Geom.*, 27(4):603–634, 2002. doi: 10.1007/s00454-002-2776-6.

C. H. Yan. Parking functions. In M. Bóna, editor, *Handbook of Enumerative Combinatorics*, pages 835–894. CRC Press, Boca Raton, FL, 2015.