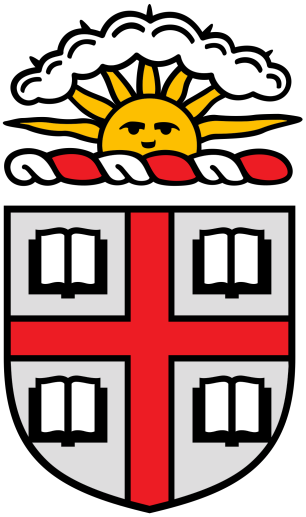
Abstract of "Honors Thesis:
Banzhaf Power in Hierarchical Games" by John Randolph, B.S., Brown University, May 2022.

The Banzhaf Power Index is a widely-used method for understanding the power of participants in voting games. Some voting games exhibit a hierarchical structure, such as the French Senate or ensemble learning methods; such games are called hierarchical voting games. In this thesis, we introduce an algorithm to calculate the Banzhaf Power Index in a hierarchical voting game with a runtime of $O(nS2^S)$, where n is the number of players in the game and S is the branching factor. This is an exponential improvement over the naive algorithm, which has runtime $O(n^2 2^n)$. We apply our algorithm to two examples: French Senate elections and the importance of words in determining positive or negative sentiment in language. We calculate the power of individual voters in France, which is an intractable problem running the naive algorithm, and show clear improvement in the complexity of computing the importance of words.

Honors Thesis:
Banzhaf Power in Hierarchical Games



by
John Randolph
B. S., Brown University., 2022
Applied Math-Computer Science, Behavioral Decision Sciences

A thesis submitted in partial fulfillment of the
requirements for the Degree of Bachelor of Science
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2022

This thesis by John Randolph is accepted in its present form by
the Department of Computer Science as satisfying the thesis requirement
for the degree of Bachelor of Science.

Date _____ _____

Amy Greenwald, Director

Recommended to the Council

Date _____ _____

Caroline Klivans, Reader

Approved by the Council

Date _____ _____

Dean of Computer Science

Acknolwedgements

I would like to acknowledge Professor Amy Greenwald and Denizalp Goktas for sheparding me through this process and supporting all of my scatter-brained research ideas over the past two years. Thank you for taking in someone with no research experience and teaching me not only how to read papers, think through complex problems, and communicate my ideas in an academic way, but how to have a good time doing it!

I would also like to thank Professor Caroline Klivans for being my second reader, Professor Bjorn Sandstede for being my advisor, and Professor Brian Knight for his help this fall. Finally, I would like to thank the whole Brown CS department and all of my great professors over my time here at Brown.

Contents

List of Figures

1 Introduction

The outcomes of elections across the world touch many aspects of our lives, from the taxes we pay to the quality of the air we breath. But these elections often involve complicated mechanisms, such as when voters have different weights in determining the outcome. One famous example is the United States electoral college. In the electoral college, which elects the president of the United States, each state has voting weight equal to its number of Senators plus its Representatives, a number that varies between 3 and 55 [5]. France's Senate elections are even more complicated: voters elect delegates from their communes, who in turn vote in one of many electoral colleges across the country to choose Senators [29]. There are also complex non-political elections: for example, stockholder votes in a corporation involve each stockholder having voting weight equal to the amount of stock they own [13] [20]. In these systems, where the mechanism for determining the outcome can be complicated, it is often unclear how much influence each voter has over the outcome. But it is important to understand the distribution of power among voters to see whether a system is fair or not - usually, we want to follow principles such as "one person, one vote" or voting power equal to some function of the population or tax base of a given representative. Objective measures of power help us determine whether the system meets that standard of fairness, and, when a system is unfair, detect it even when it the mechanism is complex.

One way to model elections is as voting games[5]. Voting games consist of a set of voters and a function, the characteristic function, which takes as input a coalition of these voters and returns a binary variable. If the characteristic function outputs a 1, we say that the coalition is a winning coalition, and if the characteristic function outputs a 0, we say that the coalition is a losing coalition. For example, if we model the United States electoral college as a voting game, then the set of voters is the set of states and the characteristic function returns 1 for a coalition whose electoral votes sum to at least 270. So any coalition of states whose total electoral votes is at least 270 is a winning coalition and every other coalition is a losing coalition.

The first studies on the distribution of power in voting games date back to as early as 1944. Following von Neumann and Morgenstern's seminal work on power in voting games[21], Banzhaf proposed a way to measure power in voting games [4]. The Banzhaf power index, or BPI, was first applied by Turnovec and has since then remained a popualar tool [34]. It involves looking at every subset of voters and, if that subset is a winning coalition, determining which voters are critical to it being a winning coalition. The full algorithm can be found in the preliminaries section. A few applications include the United State electoral college [5], the European Union [12] [9] [3] [28] [25], the International Monetary Fund [2], feature importance in machine learning [15], and shareholders and corporate boards [13] [20].

An alternate and popular power index is the Shapley-Shubik power index, or SSPI, which involves looking at every permutation of voters rather than every subset of voters (as the BPI does) [32]. The SSPI has been widely used in modelling power distribution in the electoral college [35], feature importance in machine learning [11] and shareholders on corporate boards [23]. This paper is not concerned with the SSPI, but the techniques we develop in this thesis provide directions for future research to extend our results to the SSPI. In this thesis, we'll use the BPI as our method to determine the power of votes in an election.

Unfortunately, the computation of the BPI is in general intractable, since its runtime is exponential. When run directly, as in the naive algorithm laid out in the preliminaries section, each of the 2^n coalitions is checked to see if it is a winning coalition, an operation that takes $O(n)$ time, all of this for each of the n players. This leads to an overall time complexity of $O(n^2 2^n)$ [14], which is prohibitively expensive. In this thesis, we'll show how we can exploit a particular type of characteristic function to reduce that complexity.

A characteristic function can be any function that takes a coalition of voters as input and produces a binary output, so many characteristic functions are not as simple as the one that models the United States electoral college. Some voting games are multi-level rather than single level: research has introduced the notion of compound voting, which is modelled as a multi-tiered voting game [8] [10]. These games are structured as trees, where the voters are the leaf nodes and their votes are propagated up the tree, at each level the result being determined by a characteristic function. Votes propagating up the tree means that the votes cast at the bottom determine how

higher-level delegates vote - for example, if most people in Arizona vote for the Republican candidate for president, then Arizona itself will cast its votes for the Republican candidate in the U.S. Electoral College. The characteristic function of the entire game, then, consists of the output of each of these characteristic functions, one at each branch of the tree. We call this type of voting game a hierarchical voting game. This is the structure we will exploit to speed up computation time.

One example of a compound voting game is elections for the French Senate, which involve multiple levels of electoral colleges. Voters vote for delegates from their communes, who vote in their own electoral colleges to elect the Senators from their district, who vote in the Senate. Then, this is a three-tiered game (voters choosing delegates, delegates choosing Senators, Senators voting on legislation). At each level, votes propagate up the tree via an electoral college - so the characteristic function at each level is an electoral college and the overall characteristic function is the entire mechanism of electing the French Senate. We discuss this example in much greater detail in section 4.

Another application of a hierarchical voting game is tree structured ensemble learning models, i.e., an ensemble learning model that combines multiple learning algorithms in a tree structure in order to obtain better predictive performance than any of the individual algorithms alone. In order to predict for some data point, each individual model computes its prediction. Then these predictions are propagated up the tree, with each level of the tree the predictions combined according to a characteristic function.

For each of these two cases, the BPI is instructive in understanding how much power each voter has. In a machine learning ensemble, it is helpful for an end user to understand which learning model has the greatest influence over the ensemble's prediction. In the French Senate, it is helpful to know if voters in different regions have different amounts of power (and thus representation) in their government.

In this paper, we introduce an efficient algorithm for computing the BPI on hierarchical voting games. Running the BPI directly takes time complexity of $O(n^2 2^n)$, where n is the number of players in the game. The runtime of our algorithm is $O(nS2^S)$, where n is the number of nodes in the tree and S is the branching factor. This sharply cuts the runtime of the calculation, making much larger problems tractable, especially if they have a small branching factor. Our algorithm can also be run in a depth-first manner, which makes the computation of the power for specific voters much quicker.

To demonstrate the power of this more efficient algorithm, we calculate the power of individual voters in Toulouse and Saint Martin in French Senate elections, which would be intractable to calculate directly with the naive algorithm. We find that a voter in Saint Martin has about six times as much power in the French Senate than a voter in Toulouse. We then run large-scale experiments for the problem of vocabulary selection, a case where our algorithm is only approximately equal to the BPI. In this case, we show that our algorithm significantly outperforms the naive algorithm: for example, running our algorithm on a paragraph of 10 words takes under thirty seconds, whereas it takes over three hours running the naive algorithm. There is also a minimal amount of accuracy lost, as fully displayed in the results section.

2 Preliminaries

2.1 Voting Games

A **voting game** $G = (N, v)$ is a tuple where N is a set of players and $v : 2^N \to \{0, 1\}$ is a characteristic function which takes as input a coalition and outputs a binary variable with $v(\emptyset) = 0, v(N) = 1$. N' is a **winning coalition** if $v(N') = 1$; otherwise it is a **losing coalition**. The **complement** of a coalition $c \subseteq N$ for a game $G = (N, v)$ is the coalition $N \setminus c$. A game is said to be **balanced** if the complement of a winning coalition is a losing coalition and the complement of a losing coalition is a winning coalition. That is, given $c \subseteq N$ for a balanced game $G = (N, v)$, if $v(c) = 1$, then $v(N - c) = 0$, and if $v(c) = 0$, then $v(N - c) = 1$.

A **minimal winning coalition** is a coalition $c \subseteq N$ s.t. $v(c) = 1$ and there does not exist a coalition $c' \subset c$ s.t. $v(c') = 1$ exists. A player $i \in N$ in game $G = (N, v)$ is called a **dictator** if $\{i\}$ is the only minimal winning

coalition in the game.

There are also a couple ways to create new games from other games.

Given $G_1 = (N_1, v_1), G_2 = (N_2, v_2)$, we define $G_1 \vee G_2 = (N_1 \cup N_2, v')$, where $v'(S) = \max(v_1(S \cap N_1), v_2(S \cap N_2))$, where $S \subseteq N_1 \cup N_2$.

Given $G_1 = (N_1, v_1), G_2 = (N_2, v_2)$, we define $G_1 \wedge G_2 = (N_1 \cup N_2, v')$, where $v'(S) = v_1(S \cap N_1)v_2(S \cap N_2)$, where $S \subseteq N_1 \cup N_2$.

Given $G_1 = (N_1, v_1), G_2 = (N_2, v_2)$, we allow players $i, j \in N$, to form a **bloc** ij. This results in a game $G' = (N \setminus \{i, j\} \cup \{ij\}, v')$, where $v'(S) = v(S)$ if $ij \notin S \subset N'$ and $v'(S) = v((S \setminus \{ij\}) \cup \{i, j\})$ if $ij \in S \subset N'$.

For a **hierarchical voting game**, we refer to a tree $T = (M, N, \{E_k\}_{k \in M}, \{v_k\}_{k \in M})$ with nodes M and leaf nodes $N \subseteq M$. Each $E_k \subseteq M$ is the set of children of k. The set of leaf nodes $N \subseteq M$ is referred to as the **players**. Note that $E_i = \emptyset$, for all leaf nodes $i \in N$. Each local characteristic function v_k, where $k \in M$, is defined $v_k : 2^{E_k} \to \{0, 1\}$.

Edges point away from the root. A **parent** $t(k) \in M$ is the unique node such that $k \in E_{t(k)}$. As a result of the tree structure, all nodes have exactly one parent, except for the root, which has no parent. The **ascendants** of a node k, denoted $A_k \subseteq M$, is the set of nodes that are either the parent of k or the parent of an ascendant of k. The **descendants** of a node k, denoted $D_k \subseteq M$, is the set of nodes that are either the child of k or the child of an descendant of k. The node $d_{j \to i} \in M$, given $i \in D_j$, is the unique node s.t. $d_{j \to i} \in E_j$ and $d_{j \to i} \in A_i$ We denote the root of the tree $r \in M$. See Figure 1 (below) for a graphical example of these familial relations.

Then, a **hierarchical voting game** is a game with the characteristic function:

$$v(S) = v'(S; r) \tag{1}$$

$$v'(S; k) = \begin{cases} v_k(S) & \text{if } E_k = \emptyset \\ v_k(\cup_{i \in E_k} v'(S \cap (D_i \cup i); i)) & \text{if } E_k \neq \emptyset \end{cases} \tag{2}$$

where $S \subseteq N$ and $k \in M$.

We denote a hierarchical voting game as a tree $T = (M, N, \{E_k\}_{k \in M}, \{v_k\}_{k \in M})$. See figure 2 (below) for a graphical example.

For a node $k \in M$ in a hierarchical voting game $G = (M, N, \{E_k\}_{k \in M}, \{v_k\}_{k \in M})$, define $k_{\text{leaf}} = N \cap D_k$. Then, the **subgame** $G_k = (D_k, k_{\text{leaf}}, \{E_k\}_{k \in D_k}, \{v_k\}_{k \in D_k})$ is the hierarchical voting game with k as the root.

A **one-level subgame** $G_k^{\min} = (E_k, v_k)$ of a hierarchical voting game $G = (M, N, \{E_k\}_{k \in M}, \{v_k\}_{k \in M})$ is a voting game where $k \in M$.

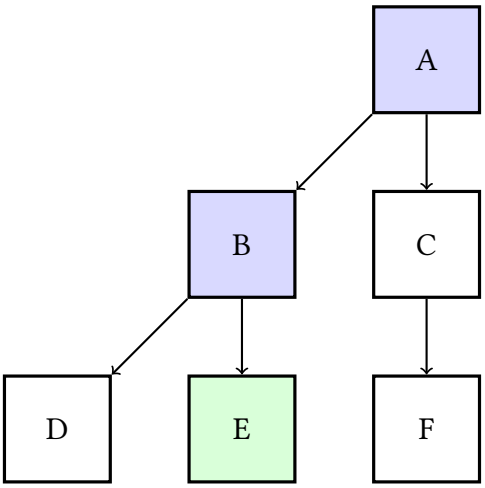
2.2 Weighted Games

A **weighted game** $(N, q, \boldsymbol{w})$ is a voting game such that the characteristic function is of the form:

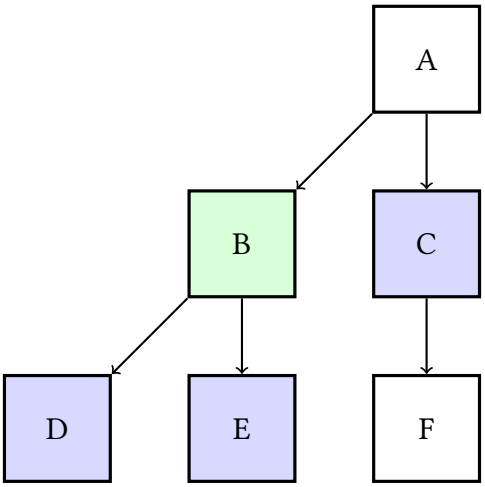
$$v(S) = \begin{cases} 1 & \text{if } \sum_{i \in S} w_i \geq q \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where $S \subseteq N$, $\boldsymbol{w}_i \in \mathbb{N}_+$ is a voting weight associated with each player $i \in N$, and $q \in \mathbb{N}_+$. A **weighted majority game** is a weighted game where $q = \frac{1}{2} \sum_{i \in N} w_i$. By default, $\boldsymbol{q} = \frac{1}{2}$, i.e., when it is unspecified. All weighted majority games are balanced. We denote a weighted majority game as a tuple $(N, \boldsymbol{w})$. One example of a weighted game is the U.S. Electoral College, where the players are the states, the weights are their electoral votes, and the quota is 270. For a small, graphical example, see Figure 3 (below).

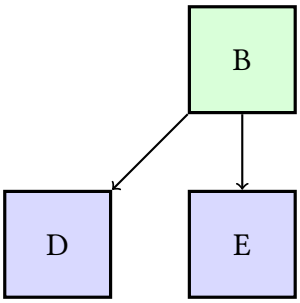
Figure 1: Examples of tree familial relations



Example of ascendants (B and A) and parent (B) of player E.



Example of children (D and E), descendants (D and E), and sibling (C) of player B.



Subgame with player B as the root.

2.3 Hierarchical weighted games

For a **hierarchical weighted game** we refer to a tree $T = (M, N, \{E_k\}_{k \in M}, \boldsymbol{w})$ with nodes M and leaf nodes $N \subseteq M$. Just as in a hierarchical voting game, $E_k \subseteq M$ is the set of children of $k \in M$ and the set of leaf nodes $N \subseteq M$ is referred to as the **players**. Each node $k \in M$ in the tree is assigned a **weight** $w_k \in \mathbb{N}_+$. Each non-leaf

node $k \in M$ is assigned a **quota** $q_k \in \mathbb{N}_+$. The weight of the root is 1.

Then, a **hierarchical weighted game** is a hierarchical voting game with the local characteristic functions, for some $k \in M$ and $S \subseteq E_k$:

$$v_k(S) = \begin{cases} w_k & \text{if } \sum_{i \in S} v_i(S) \geq q_k \\ 0 & \text{otherwise} \end{cases}, \text{when } E_k \neq \emptyset \tag{4}$$

$$v_k(S) = \begin{cases} w_k & \text{if } k \in S \\ 0 & \text{otherwise} \end{cases}, \text{when } E_k = \emptyset \tag{5}$$

That means that the characteristic function of the game overall is:

$$v(S) = v'(S; r) \tag{6}$$

$$v'(S; k) = \begin{cases} w_k & \text{if } (E_k = \emptyset \text{ and } k \in S) \\ & \text{or } (E_k \neq \emptyset \text{ and } \sum_{j \in E_k} v'(S; j) \geq q_k) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $S \subseteq N$ and $k \in M$. This hierarchical weighted game is a **hierarchical weighted majority game** if $\forall k \in M \backslash N, q_k = \frac{1}{2} \sum_{i \in E_k} w_i$. We denote a hierarchical weighted game as a tuple $(T, \boldsymbol{q}, \boldsymbol{w})$. So, when it is unspecified, by default $q_k = \frac{1}{2} \sum_{i \in E_k} w_i$ for all $k \in M$. All hierarchical weighted majority games are balanced. We denote a hierarchical weighted majority game as a tuple $(T, \boldsymbol{w})$.

2.4 Power Indices

A **power index** is a function $\boldsymbol{p} : F_n \to \mathbb{R}^n$, where F_n is the set of games with n players, and $\mathbb{R}^n$ is a vector that contains a weight for each player $i \in N$, which we interpret as its power.

The **Banzhaf power index** is one such index. It is defined as follows:

$$p_i^{\text{BPI}}(N, v) = \frac{\sum_{S \subseteq N \backslash \{i\}} (v(S \cup \{i\}) - v(S))}{2^{|N|-1}}. \tag{8}$$

The Banzhaf power index is desirable because it is uniquely characterized by four axioms which are generally agreed upon to be desirable properties of a power index.

The sum principle: For $G_1 = (N_1, v_1), G_2 = (N_2, v_2), \boldsymbol{p}_i(G_1 \vee G_2) + \boldsymbol{p}_i(G_1 \wedge G_2) = \boldsymbol{p}_i(G_1) + \boldsymbol{p}_i(G_2)$

The dictator principle: For $G = (N, v)$, if i is a dictator in the game, then $\boldsymbol{p}_i(G) = 1$.

Equal treatment: Let i, j be two players in $G = (N, v)$, that satisfy $v(S \cup \{i\}) = v(S \cup \{j\}), \forall S \subseteq N \setminus \{i, j\}$. Then $\boldsymbol{p}_i(G) = \boldsymbol{p}_j(G)$.

The two-voter bloc principle: Let $G' = (N', v')$ be the game obtained from $G = (N, v)$ when the players $i, j \in N$ form a bloc ij. Then, $\boldsymbol{p}_{ij}(G') = \boldsymbol{p}_i(G) + \boldsymbol{p}_j(G)$.

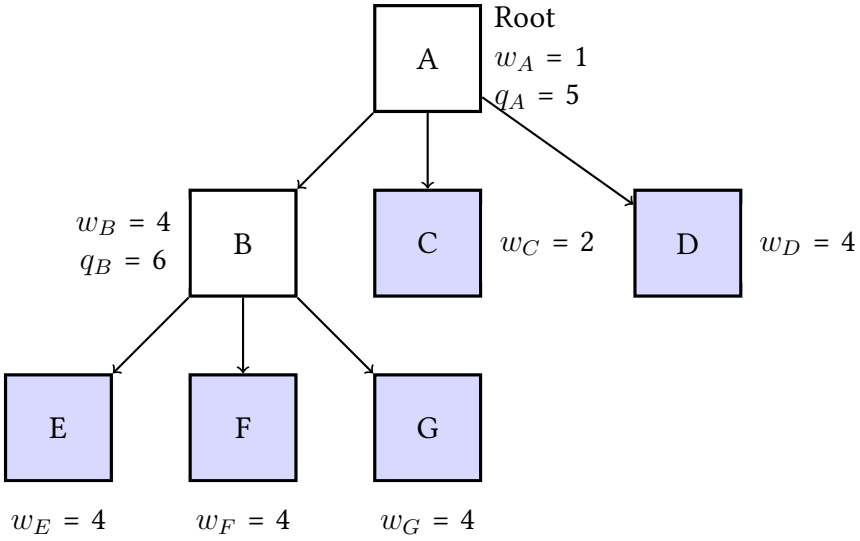
The ordinary way to compute the BPI on a voting game is as follows:

Algorithm 1: BH Flat Algorithm

Result: p_i^{BPI}, the power of each player $i \in N$ players

```
Input: players N
for i ∈ N do
    for S ⊆ N do
        p_i^BPI = p_i^BPI + |v(S ∪ {i}) − v(S)|
    end
    p_i^BPI = p_i^BPI / 2^{|N|−1}
end
```

Figure 2: Example of hierarchical weighted game



Example of a Hierarchical Weighted Majority Game $G = (M, N, \{E_k\}_{k \in M}, \boldsymbol{w})$, with

$$
\begin{aligned}
M &= \{A, B, C, D, E, F, G\} \\
N &= \{C, D, E, F, G\} \\
E_A &= \{B, C, D\} \\
E_B &= \{E, F, G\} \\
E_C &= E_D = E_E = E_F = E_G = \emptyset \\
\boldsymbol{q} &= \{5, 6\} \\
\boldsymbol{w} &= \{1, 4, 2, 4, 4, 4, 4\}
\end{aligned}
$$

Then, we know that the characteristic function is:

$$
\begin{aligned}
&v(S) = v_A(S) \\
&v_A(S) = \begin{cases} 1 & \text{if } (\sum_{j \in \{B,C,D\}} v_j(S) \geq 5) \\ 0 & \text{otherwise} \end{cases} \\
&v_B(S) = \begin{cases} 4 & \text{if } (\sum_{j \in \{E,F,G\}} v_j(S) \geq 6) \\ 0 & \text{otherwise} \end{cases} \\
&v_k(S) = \begin{cases} w_k & \text{if } k \in S \\ 0 & \text{otherwise} \end{cases} \quad , k \in \{C, D, E, F, G\}
\end{aligned}
$$

As this involves, for every player, calculating the characteristic function for every subset of the players, this has a complexity of $O(n^2 2^n)$, where n is the number of players in the game.

Some weighted voting games involve players that all have the same weight. It makes intuitive sense that all these players have the same BPI, and we will codify that in the following lemma. This lemma will be useful when we discuss the French Senate elections, as they involve some weighed games where each player has the same voting weight.

Lemma 2.1. *If $w_i = w_j \forall i, j, \in N$ for some weighted game $G = (N, \boldsymbol{w}, q)$, all players have equal BPI, and that is $\frac{1}{|N|}$.*

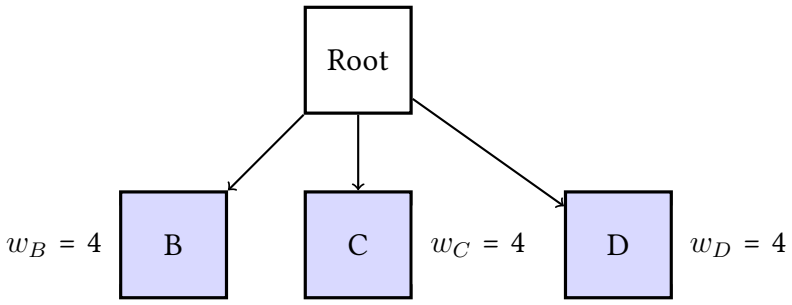
Proof. If two players $i, j \in N$ in a weighted game $G = (v, N, \boldsymbol{w})$ have the same weight, then we know that $v(S \cup \{i\}) = v(S \cup \{j\}), \forall S \subseteq N \setminus \{i, j\}$. Then, by the anonymity axiom, $\boldsymbol{p}_i(G) = \boldsymbol{p}_j(G)$. In a game where all players have the same weight, then they must all have the same power. Since all $|N|$ players have equal power and their power sums to one, they each have power $\frac{1}{|N|}$. □

Figure 3: Example of weighted majority game and BPI

Consider weighted majority game $G : (N, \boldsymbol{w})$ where $N = (B, C, D)$ and $\boldsymbol{w} = \{4, 4, 4\}$. Then, we know that the characteristic function is:

$$v(S) = \begin{cases} 1 & \text{if} \sum_{i \in S} w_i \geq 6 \\ 0 & \text{otherwise} \end{cases},$$

I.e., any coalition whose total weight is at least six is a winning coalition. The game is displayed graphically below, with players in blue.



We can apply the BPI on one player, for example B:

$$\begin{aligned} p_B^{\text{BPI}}(N, v) &= \frac{\sum_{S \subseteq N \setminus \{B\}} (v(S \cup \{B\}) - v(S))}{2^{|N|-1}} \\ &= \frac{(v(\{\} \cup \{B\}) - v(\{\})) + (v(\{C\} \cup \{B\}) - v(\{C\}))}{2^{|\{B,C,D\}|-1}} \\ &+ \frac{+(v(\{D\} \cup \{B\}) - v(\{D\})) + (v(\{C, D\} \cup \{B\}) - v(\{C, D\}))}{2^{|\{B,C,D\}|-1}} \\ &= \frac{(0-0) + (1-0) + (1-0) + (1-1)}{2^{3-1}} \\ &= \frac{2}{2^2} \\ &= \frac{1}{2} \end{aligned}$$

3 Improved Algorithm

A straightforward algorithm for calculating the BPI would involve enumerating every subset of the players, which is quite expensive. In this section we introduce an improved algorithm for computing the BPI on a hierarchical voting game. Our algorithm runs in $O(nS2^S)$ time, where n is the number of nodes in the tree and S is the branching factor of the tree. This is a major improvement over the direct method which has a runtime of $O(n^2 2^n)$. To do this, we will introduce some new notation and define multiplicative Banzhaf power.

For any hierarchical voting game $G = (M, N, \{E_k\}_{k\in M}, \{v_k\}_{k\in M})$, we define the **multiplicative Banzhaf power index**, or MBPI, of some player $i \in N$ as $\boldsymbol{p}^{\text{MBPI}} : F_n \to \mathbb{R}^N$. We will prove that when the game is balanced, it is equal to the BPI.

$$p_i^{\text{MBPI}}(G) = \begin{cases} 1 & \text{if } E_r = \emptyset \\ p_i^{\text{MBPI}}(G_{d_{r\to i}})(p_{d_{r\to i}}^{\text{BPI}}(G_r^{\min})) & \text{otherwise} \end{cases} \tag{9}$$

3.1 Proof of correctness

In order to prove that the MBPI of any player in a hierarchical voting game is equal to the BPI of the same player, we will first prove two lemmas.

Lemma 3.1. *Any voting game (N, v) that is balanced has $2^{|N|-1}$ winning coalitions and $2^{|N|-1}$ losing coalitions.*

Proof. First notice that in any voting game with N players, there are $2^{|N|}$ coalitions. Every coalition $c \subseteq N$ is either winning or losing and its complement $N \setminus c$ is the opposite, so there is a one-to-one relationship between winning and losing coalitions. Thus there are 2^{N-1} winning coalitions and 2^{N-1} losing coalitions. □

Lemma 3.2. *Take any balanced hierarchical voting game $G = (M, N, \{E_k\}_{k\in M}, \{v_k\}_{k\in M})$ with some player $i \in N$. Then, if G' is the same game but with every node that is not a sibling of i, ascendant of i, sibling of an ascendant of i, or i itself removed, along with the corresponding edges and characteristic functions, $p_i^{\text{BPI}}(G) = p_i^{\text{BPI}}(G')$.*

Proof. Given a hierarchical voting game $G = (M, N, \{E_k\}_{k\in M}, \{v_k\}_{k\in M})$, pick some player $i \in N$. Now consider creating a new game $G' = (M', N', \{E_k\}_{k\in M'}, \{v_k\}_{k\in M'})$ by taking G and adding n nodes below some player j that is not an ascendant of i, as well as the corresponding edges and characteristic functions. Then these n new players will neither be siblings nor ascendants nor siblings of ascendants of i.

Then there is a hierarchical voting game ${G'}_j$ that is a subgame of G' whose root is j and whose players are the n new nodes. By Lemma 3.1, in this game, there are 2^{n-1} winning coalitions and 2^{n-1} losing coalitions in this game. Then, for each coalition $c \subseteq N \setminus \{i\}$ that i is critical to in G, there are corresponding 2^{n-1} coalitions in G'. This is because for each $c \subseteq N \setminus \{i\}$, if $j \in c$, then c corresponds to 2^{n-1} coalitions $c' \in N'$ where j is replaced by every winning coalition in ${G'}_j$, and if $j \notin c$ then c corresponds to 2^{n-1} coalitions $c' \in N'$ where every losing coalition in ${G'}_j$ is added to c.

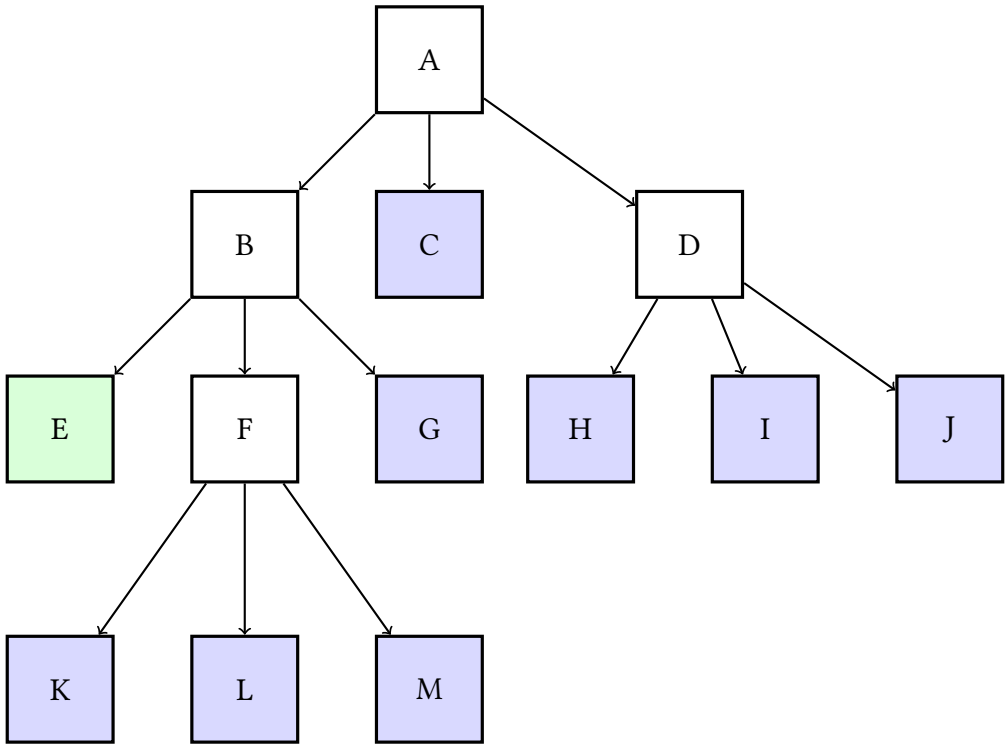
Now note that $|N'| = |N| + |n| - 1$, as there were n new nodes and j was removed as a player (though it is still a node).

Then it is the case that

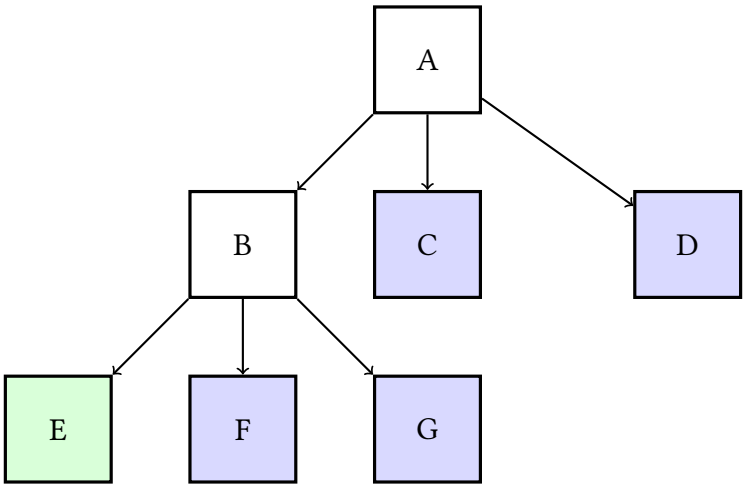
$$\boldsymbol{p}_i^{\text{bh}}(G') = \frac{\sum_{S\subseteq N'\setminus\{i\}} (v(S\cup\{i\}) - v(S))}{2^{|N'|-1}} = \frac{\sum_{S\subseteq N\setminus\{i\}} (v(S\cup\{i\}) - v(S)) * 2^{|n|-1}}{2^{|N|-1} * 2^{|n|-1}} = \frac{\sum_{S\subseteq N\setminus\{i\}} (v(S\cup\{i\}) - v(S))}{2^{|N|-1}} = \boldsymbol{p}_i^{\text{bh}}(G)$$

Thus the BPI of some player $i \in N$ is the same when a set of nodes that are not its siblings, ascendants, or siblings of ascendants are removed (with their corresponding edges and characteristic functions). So when all nodes that are not the siblings, ascendants, or siblings of ascendants of some player $i \in N$ are removed, the BPI of player i is the same. □

Figure 4: Example of removing nodes



G_4



G_5

If G_4 is a balanced hierarchical voting game, then $p_E^{\text{BPI}}(G_4) = p_E^{\text{BPI}}(G_5)$. This is because the none of the nodes that are removed (H, I, J, K, L, M) are siblings of E, ascendants of E, siblings of ascendants of E, or E itself.

Now, consider a hierarchical voting game $G = (M, N, \{E_k\}_{k \in M}, \{v_k\}_{k \in M})$ that is balanced and that has some player $i \in N$. Then consider the node $d_{r \to i} \in M$ (which is the unique node that is an ascendant of i and a child of r, the root). If, in this game, every node is a sibling of i, ascendant of i, a sibling of an ascendant of i, or i itself, then the number of coalitions that i is critical to in the subgame $G_{d_{r \to i}}$ multiplied by the number of coalitions that $d_{r \to i}$ is critical to in the one-level subgame $G_r^{\min}$ is equal to the number of coalitions that i is critical to in G. That is what is codified in the following lemma.

Lemma 3.3. *Given a hierarchical voting game $G = (M, N, \{E_k\}_{k\in M}, \{v_k\}_{k\in M})$ that is balanced, pick some $i \in N$. If $\forall j \in N, j \in E_k$ where $k \in A_i$, then*

$$\sum_{S\subseteq d_{r\to i}\text{leaf}} (v(S\cup\{i\}) - v(S)) \sum_{S\subseteq E_r} (v(S\cup\{d_{r\to i}\}) - v(S)) = \sum_{S\subseteq N} (v(S\cup\{i\}) - v(S))$$

Proof. Any coalition c that i is critical to in $G_{d_{r\to i}}$ can be combined with any coalition c' that $d_{r\to i}$ is critical to in $G_r^{\min}$ to create a coalition that i is critical to in G. Consider this coalition $c \cup c'$. Without i, c is a losing coalition in $G_{d_{r\to i}}$, so $v_{d_{r\to i}}(c \cup c') = 0$. Then, since c' is a losing coalition in $G_r^{\min}$, $v(c \cup c') = 0$. But since i is critical to c, $c \cup \{i\}$ is a winning coalition in $G_{d_{r\to i}}$, so $v_{d_{r\to i}}(c \cup c' \cup \{i\}) = 1$. Then, since $d_{r\to i}$ is critical to c', $c' \cup \{d_{r\to i}\}$ is a winning coalition in $G_r^{\min}$. Then $v(c \cup c' \cup \{i\}) = 1$. Thus, $c \cup c'$ is a losing coalition and $c \cup c' \cup \{i\}$ is a winning coalition, so i is critical to $c \cup c'$. Likewise, any coalition that i is critical to in G can be decomposed into a coalition that i is critical to in $G_{d_{r\to i}}$ and a coalition that $d_{r\to i}$ is critical to in $G_r^{\min}$.

Thus there is a one-to-one relationship between the set of coalitions that i is critical to in G and the set of pairs of coalitions where the first is one that i is critical to in $G_{d_{r\to i}}$ and the second is one that $d_{r\to i}$ is critical to in $G_r^{\min}$.

Thus the product of the number of coalitions that i is critical to in $G_{d_{r\to i}}$ and the number of coalitions that $d_{r\to i}$ is critical to in $G_r^{\min}$ is equal to the number of coalitions i is critical to in G. □

With these lemmas in hand, we now introduce the main result in this section.

Theorem 3.4. *In any hierarchical voting game that is balanced, the multiplicative Banzhaf power index is equal to the Banzhaf power index.*

Proof. Proof by induction.

Base case: The game tree itself is a leaf node. Likewise, when considered as a flat model, it is still just one leaf node. The power of this node is 1 computed by both the MBPI and the BPI.

Inductive step: Assume we have hierarchical voting games $G_1, G_2, \ldots G_n$ where $G_i = (M_i, N_i, \{E_k\}_{k\in M_i}, \{v_k\}_{k\in M_i})$ for $i \in n$. By the inductive hypothesis, the MBPI is equal to the BPI in each game. That is, for each G_i where $i \in 1 \ldots n$ and $j \in N_i$, $p_j^{\text{MBPI}}(G_i) = p_j^{\text{BPI}}(G_i)$.

Now, consider a hierarchical voting game $G_0 = (M_0, N_0, \{E_k\}_{k\in M_0}, \{v_k\}_{k\in M_0})$, constructed in the following way: $N_0 = \bigcup_{i=1}^{n} N_i$. $M_0 = \bigcup_{i=1}^{n} M_i \cup \{G_0\}$. $E = \bigcup_{i=1}^{n} E_i \cup \{\{G_0, G_1\}, \ldots, \{G_0, G_n\}\}$. $v = \bigcup_{i=1}^{n} v_i \cup \{v_{G_0}\}$. $\{v_{G_0}\}$ can be any characteristic function.

Pick some player $i \in G_0$. Then, i is a player in some game G_j, where $j \in (1, \ldots, n)$. By assumption, $p_i^{\text{MBPI}}(G_j) = p_i^{\text{BPI}}(G_j)$.

Now create a game $G_0' = (M_0', N_0', \{E_k\}_{k\in M_0}, \{v_k\}_{k\in M_0})$ that is the same as G_0 but with all nodes that are not siblings of i, ascendants of i, siblings of ascendants of i, or i itself removed. By Lemma 3.2, the power of i in this game is the same as the power of i in G_0. That is, $p_i^{\text{BPI}}(G_0') = p_i^{\text{BPI}}(G_0)$. Note that this also means that $p_i^{\text{BPI}}(G_j') = p_i^{\text{BPI}}(G_j)$.

Now we can use the following logic. First, by the definition of the MBPI, we know:

$$p_i^{\text{MBPI}}(G_0) = p_i^{\text{MBPI}}(G_j)\frac{\sum_{S\subseteq E_r\setminus\{G_j\}}(v(S\cup\{G_j\})-v(S))}{2^{|E_r|-1}} \quad \text{(Definition of MBPI)} \tag{10}$$

$$= p_i^{\text{BPI}}(G_j)\frac{\sum_{S\subseteq E_r\setminus\{G_j\}}(v(S\cup\{G_j\})-v(S))}{2^{|E_r|-1}} \quad \text{(Inductive hypothesis)} \tag{11}$$

$$= p_i^{\text{BPI}}(G'_j)\frac{\sum_{S\subseteq E_r\setminus\{G_j\}}(v(S\cup\{G_j\})-v(S))}{2^{|E_r|-1}} \quad (p_i^{\text{BPI}}(G'_j) = p_i^{\text{BPI}}(G_j)) \tag{12}$$

$$= p_i^{\text{BPI}}(G'_j)\frac{\sum_{S\subseteq E_r\setminus\{G'_j\}}(v(S\cup\{G'_j\})-v(S))}{2^{|E_r|-1}} \quad (G_k = G'_k \forall k \in \{1,\dots,n\}) \tag{13}$$

$$= \frac{\sum_{S\subseteq N'_j\setminus\{i\}}(v(S\cup\{i\})-v(S))}{2^{|N'_j|-1}}\frac{\sum_{S\subseteq E_r\setminus\{G'_j\}}(v(S\cup\{G'_j\})-v(S))}{2^{|E_r|-1}} \quad \text{(Definition of BPI)} \tag{14}$$

$$\tag{15}$$

Note that, since none of $G_k, k \in \{1\dots n\}\setminus j$ have children, the number of players in the game G'_0 is equal to the number of players in the game G'_j plus the number of children of r, minus one for G_j, which is the only child of r that has children (and so is not a player). That is,

$$|N'_j|+|E_r|-1 = |N'_0|$$
$$2^{|N'_j|}2^{|E_r|-1} = 2^{|N'_0|}$$

So, substituting in the denominator, our whole expression becomes:

$$= \frac{\left(\sum_{S\subseteq N'_j\setminus\{i\}}(v(S\cup\{i\})-v(S))\right)\left(\sum_{S\subseteq E_r\setminus\{G'_j\}}(v(S\cup\{G'_j\})-v(S))\right)}{2^{|N'_0|-1}}$$

Now consider the numerator of the equation, which is:

$$\left(\sum_{S\subseteq N'_j\setminus\{i\}}(v(S\cup\{i\})-v(S))\right)\left(\sum_{S\subseteq E_r\setminus\{G'_j\}}(v(S\cup\{G'_j\})-v(S))\right)$$

This is the number of coalitions that i is critical to in G'_j multiplied by the number of coalitions that G'_j is critical to in G'_0. Thus, by Lemma 3.3,

$$\left(\sum_{S\subseteq N'_j\setminus\{i\}}(v(S\cup\{i\})-v(S))\right)\left(\sum_{S\subseteq E_r\setminus\{G'_j\}}(v(S\cup\{G'_j\})-v(S))\right) = \sum_{S\subseteq N'_0\setminus\{i\}}(v(S\cup\{i\})-v(S))$$

Putting our numerator and denominator together, we get:

$$\frac{\sum_{S \in N_0' \setminus \{i\}} (v(S \cup \{i\}) - v(S))}{2^{|N_0'|-1}}$$

$$= p_i^{\text{BPI}}(G_0')$$

Which, by Lemma 3.2,

$$= p_i^{\text{BPI}}(G_0)$$

So we have:

$$p_i^{\text{MBPI}}(G_0) = p_i^{\text{BPI}}(G_0)$$

Thus, the equality holds in the inductive step, and the theorem holds in general. □

3.2 Computation

The structure of the MBPI leads to a more computationally efficient way to compute it.

If the hierarchical voting game is balanced, we can compute it much faster. We introduce BH Mult Algorithm to efficiently compute the MBPI.

Algorithm 2: BH Mult Algorithm

Result: p_i^{MBPI}, the power of each player $i \in N$ players

Input: root r, parent_bh b
if $E_r == \emptyset$ **then**
    $p_j^{\text{BPI}} = b$
else
    for $i \in E_r$ **do**
        for $S \subseteq E_r$ **do**
            $p_i^{\text{BPI}} = p_i^{\text{BPI}} + |v(S \cup \{i\}) - v(S)|$
        end
        $p_i^{\text{BPI}} = p_i^{\text{BPI}} / 2^{|E_r|-1}$
    end
    for $i \in E_r$ **do**
        BH Mult Algorithm($i, b\, p_i^{\text{BPI}}$)
    end
end

Note that in order to calculate the MBPI for all players, this algorithm passes over every node once, making one computation that requires looking at each permutation of its children.

Thus the runtime is $O(nS2^S)$, where n is the number of nodes in the tree and S is the branching factor. As it must be the case that $S \leq n$, we can be sure that BH Mult Algorithm is at least as fast as BH Flat Algorithm, and often faster. The smaller the branching factor is, the greater the speed-up The scenario where $S = n$ is the one where the tree is only of depth one. In that case, the algorithms are effectively the same and thus have the same runtime ($O(n^2 2^n) = O(nS2^S)$ when $n = S$).

Depth-first

A small modification of BH Mult Algorithm allows it to be run to calculate the power for one player very quickly.

Algorithm 3: BH Mult Depth-First Algorithm

Result: p_j^{MBPI}, the power of some player $j \in N$ players

Input: root r, parent_bh b, player j
if $E_r == \emptyset$ **then**
    $p_j^{\text{BPI}} = b$
else
    for $i \in E_r \cap A_j$ **do**
        for $S \subseteq E_r$ **do**
            $p_i^{\text{BPI}} = p_i^{\text{BPI}} + |v(S \cup \{i\}) - v(S)|$
        end
        $p_i^{\text{BPI}} = p_i^{\text{BPI}} / 2^{|E_r|-1}$
    end
    for $i \in E_r \cap A_j$ **do**
        BH Mult Depth-First Algorithm($i, b\, p_i^{\text{BPI}}, j$)
    end
end

BH Mult Depth-First Algorithm runs as a depth-first algorithm. This reduces the computation time of one player i to $O(|A_i|S2^S)$, where $|A_i|$ is the number of ascendants of player i and S is the branching factor. The runtime of BH Flat Algorithm run for one player i is $O(n2^n)$, where n is the number of players in the game.

This is a major advantage in large-scale computing, as algorithms that require the entire tree quickly become intractable with a large tree. A depth-first search allows us to calculate the BPI of one player in a much shorter amount of time.

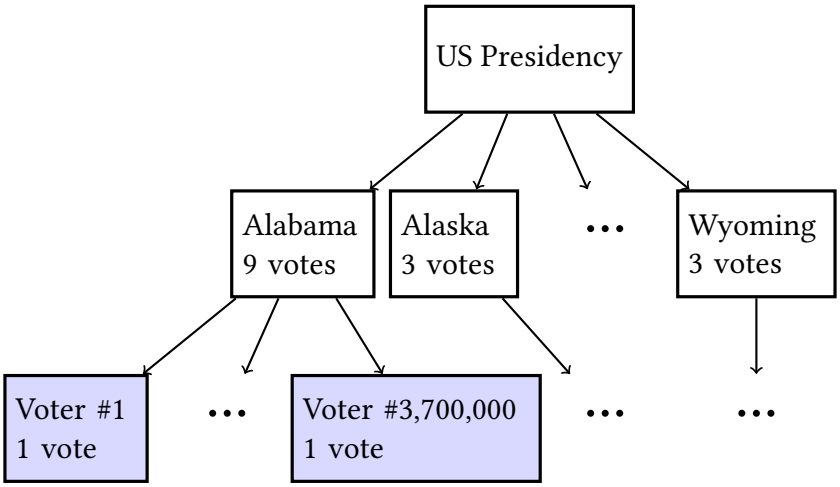
4 Example: French Senate

One specific case where the BPI is useful is in situations where people vote for elected representatives and there are varying vote weights. One classic example is the United States electoral college, which is a weighted majority game. In this weighted majority game, the top level is a root node which represents the final outcome of the electoral college. The next level is a set of nodes that are each a state, whose weight is equal to the number of electoral votes.

This game is small enough to run the complete flat BPI, which has been done several times. This results in a calculation of power for each state [5].

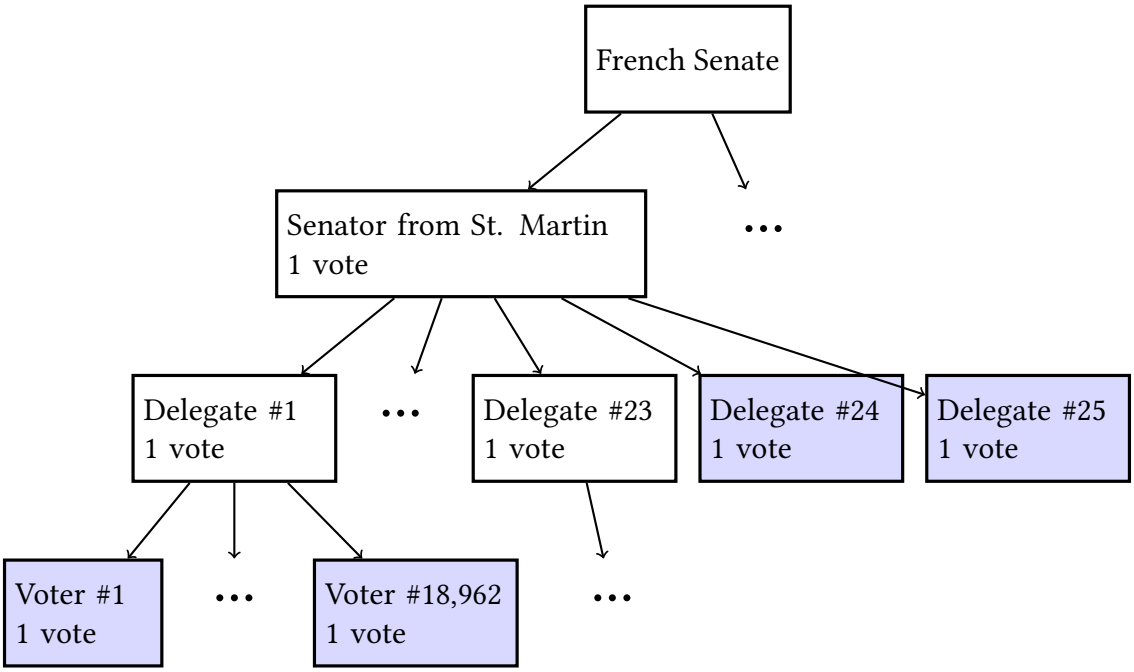
One voting system that is more complicated than the US electoral college is the election of French Senators. In France, there is universal suffrage for electing these members of parliament, but there is a complex voting mechanism. Each of the 34,965 communes in France are allotted representatives based on their population. For example, communes with 1,500-2,499 people each are allotted 5 delegates, and communes with a population of more than 30,000 are allotted 69 delegates plus one delegate for each 800 people more than 30,000 [29]. These representatives from communes then vote in electoral colleges within their district to elect the Senators from their district. There are also a few other politicians that are included in these electoral colleges, but delegates from communes make up more than 95% of the voters in the electoral colleges [30]. There are a few members of parliament that are elected in slightly different ways, but the vast majority of the 348 members of the French senate are elected from the 128 electoral colleges. This is a hierarchical weighted majority game with several levels: voters in a commune who elect delegates, commune delegates and other politicians in the electoral college who elect members of parliament,

Figure 5: The United States Electoral College as a hierarchical weighted majority game.



and members of parliament in the Senate.

Figure 6: The French Senate as a hierarchical weighted majority game.



Calculating the power of any specific voter with the BH Flat Algorithm would be too expensive for this whole game, since it would involve looking at all 2^n voters, where n is the voting population of France (around 47 million). Even looking just at the power of each commune in the senate would be to expensive, as it would involve looking at all $2^{34,965}$ subsets of all the communes. However, we can utilize BH Mult Depth-First Algorithm to calculate the MBPI for any specific voter.

4.1 Voters in Toulouse and Saint Martin

Now we can ask an interesting question: who has more power in the French Senate, a voter in the small commune (within a small district) of Saint Martin or the large commune (within a large district) of Toulouse?

As this is a balanced hierarchical voting game, Theorem 3.4 tells us that the power of some voter in the French parliament is equal to the product of the power of the voter in their commune, the power of their commune delegates

in the district's electoral college, and the power of their district's Senators in the Senate. Again, this is a hierarchical weighted majority game where each player's weight is 1 at each level. We obtain the following information from the internet [16] [18] [19].

Commune	Voters	Delegates	Electoral college size	Senators	Total Senators in France
St. Martin	18,962	23	25	1	348
Toulouse	254,538	621	2,955	10	348

However, even calculating the BPI of a Senator in the Senate seems at first glance too expensive. Running the BH Flat Algorithm on just that one level game would involve looking at all 2^{348} subsets of the Senators. However, recall that by lemma 2.1, we know that the power of each Senator is equal, since the Senate is a weighted voting game and each Senator has equal voting weight. Similarly, each delegate has equal power within their electoral college and each voter has equal power within their commune. This is the advantage of the MBPI: voters across the country do not have equal power and their power is difficult to calculate since it would involve running the BH Flat Algorithm on the whole population of France, but using the MBPI we can decompose the game into sub-games which are each easy to calculate since they involve players with equal weights.

So, we can calculate the MBPI of a voter in their commune, the power of a commune's delegates in its district's electoral college, and the power of the district's Senators in the Senate. From this we can obtain the power of a voter in the Senate. Also, we know that there were 16,558,379 voters in municipal elections in France in 2020, so the average French voter has power $\frac{1}{16558379}$ [19].

Voter	P. of voter in commune	P. of commune in E.C.	P. of Senators in Senate	P. voter in Senate
St. Martin Voter	$5.273e^{-5}$	.92	$2.873e^{-3}$	$1.394e^{-7}$
Toulouse Voter	$3.929e^{-6}$	.2101	$2.873e^{-2}$	$2.372e^{-}8$
Avg. French Voter	-	-	-	$6.039e^{-}8$

We find that, via the BPI, a voter in Saint Martin has about double the power of the average French voter, while a voter in Toulouse has about one third of the power of the average French voter.

5 Experiment

To verify that BH Mult Algorithm is indeed an improvement over BH Flat Algorithm, we ran experiments on language data. The objective of the BPI in this situation is determining the most important words in a body of text.

5.1 Vocabulary selection

The problem of vocabulary selection is an important one in the creating NLP neural networks. A smaller vocabulary makes a model more interpretable [1] [27], requires less memory [31], is more able to be used in a resource-constrained setting [33], and is less prone to over fitting [17] [7]. One way to choose the vocabulary is to determine which words are most important via the BPI [22]. This is a costly problem, as NLP neural networks can be often involve large datasets with long sentences and many words. This problem is approximately a hierarchical voting game, and as such we can utilize the BH Mult Algorithm to get a close approximation of the BPI much quicker than running the BH Flat Algorithm.

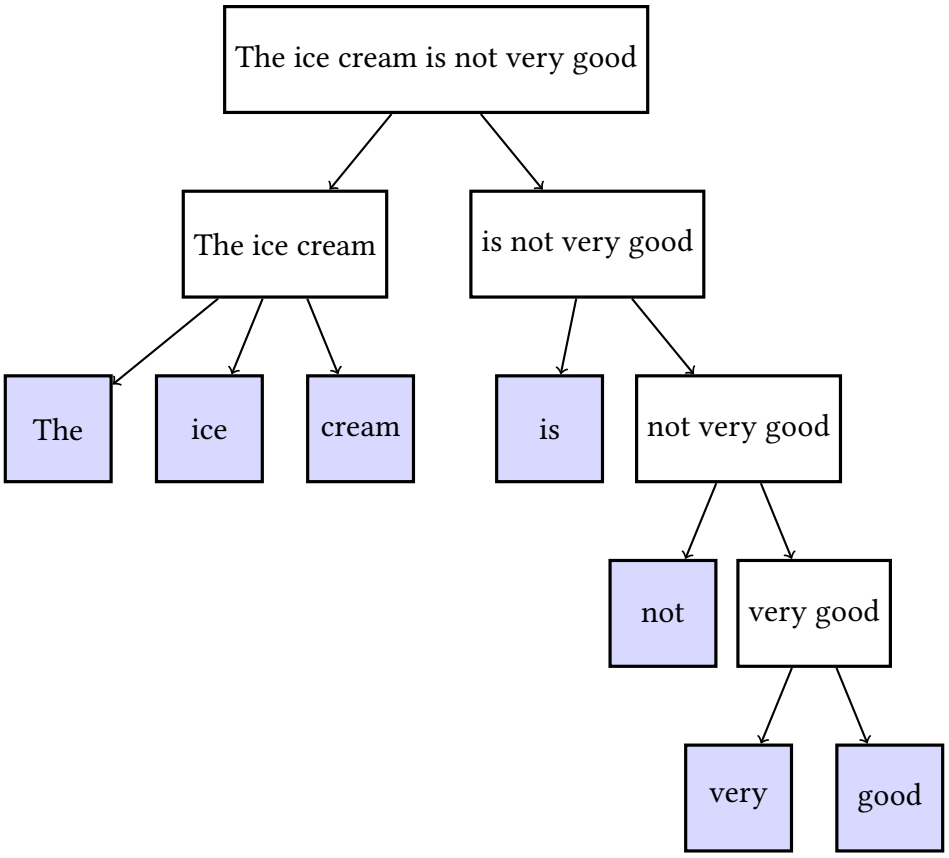
5.2 As a hierarchical voting game

We model a paragraph as a voting game where the output is the sentiment of the paragraph and the players are the words.

For our characteristic function, we use the sentiment analysis classifier in the Stanford NLP Group's Stanza library [24]. Our binary output is a 1 for positive sentiment and 0 for neutral or negative sentiment. This means that in running the algorithm, words are given higher power when their addition or removal from the set causes the sentiment to flip from positive to neutral/negative or from neutral/negative to positive. This aligns with our dataset, which is yelp reviews, which generally have clear positive or negative sentiment.

We can further model any paragraph as a hierarchical voting game by dividing it into a parse tree using Stanza [24]. Stanza's constituency parser divides the paragraph into blocks that contain related words. One example is below:

Figure 7: Simple Parse Tree



The straighforward approach would be to run the BH Flat Algorithm on the entire paragraph, using the sentiment analysis classifier as our characteristic function. However, since we have the paragraph as a hierarchical voting game, we can use BH Mult Algorithm.

Note that for the BPI to be equal to the MBPI, the game needs to be a hierarchical voting game that is balanced. However, there are two key differences between the situation in this experiment and a balanced hierarchical voting game:

- This hierarchical voting game is not balanced. For every collection of words in a sentence, the opposite collection of words does not produce the opposite sentiment. For example, in the sentence "The food is not good," "not good" produces a negative sentiment but "the food is" produces a neutral sentiment rather than a positive sentiment.

- The overall characteristic function (sentiment analysis) does not exactly match the combination of characteristic functions at each level of the tree (which are also sentiment analysis). For example, both the phrase "I feel not" and the phrase "bad" would register as negative sentiment. However, all together, "I feel not bad" would register as positive sentiment. Fortunately, this situation is fairly rare. Most English sentences do follow the expected result of combining their constituent parts, such as the in the example above, where "the ice cream" (neutral) and "is not very good" (negative), combine to form "the ice cream is not very good" (negative), just as we would expect.

Due to these two reasons, the BPI is not equal to the MBPI. However, for English sentences, the two models of power are close enough to each other that they produce very similar results. The savings in time are large enough that some inaccuracy may be tolerable. The complexity / accuracy tradeoff is discussed more below.

5.3 Accuracy metric

To asses the accuracy of our approximate algorithm, we calculate two metrics: the mean squared error and Spearman's ρ. The mean squared error is established as a measure of the distance between two vectors and has been used as a measure of an estimation of the BPI before [26].

But, in the case of vocabulary selection, what we care most about is the ordering of the words, since we are likely to take the top few of them to be the vocabulary in our machine learning model. Thus, we use the rank-based metric Spearman's ρ, that depends only on the final ranking of the words. Spearman's ρ is a widely-used correlation coefficient between two rankings of the same length [6].

First, for each player i, we calculate the BPI as: $p_w^{\text{BPI}} = \frac{1}{|c|} \sum_{r \in c} p_w^{\text{BPI}}(r)$, where c is the set of paragraphs and $p_w^{\text{BPI}}(r)$ is the power of word w in paragraph r. From that, we obtain a vector of weights from both the BPI and MBPI.

Then we follow the formula for mean squared error, which is $MSE = \sqrt{\frac{\sum_{w \in N} (p_w^{\text{BPI}} - p_w^{\text{MBPI}})^2}{|N|}}$.

To calculate Spearman's ρ, we first calculate the rank of each word $w \in N$ in each BPI, denoted s_w^{BPI} and s_w^{MBPI}. Then use the standard formula, which is $\rho = 1 - \frac{6 \sum_{w \in N} (s_w^{\text{BPI}} - s_w^{\text{MBPI}})^2}{|N|(|N|^2 - 1)}$.

5.4 Results

First, we calculated the runtime of running both algorithms on paragraphs of different lengths. Using data from our yelp dataset, we found sentences of each given length and timed how long the algorithms took. For the BH Flat Algorithm, we used one data point. This is sufficient because for the flat algorithm, the runtime should always be 2^n, where n is the number of words in the paragraph, no matter what the words are. However, for the BH Mult Algorithm, the runtime can vary between paragraphs of the same length, since they are split into different parse trees (via Stanza), and parse trees of different shapes can have different runtimes. So, for the BH Mult Algorithm, we used 20 samples for every different paragraph length.

Next, we calculated the mean squared error as the length of the paragraph increases, to get a sense of how close the two distributions are. Each data point is the average of the algorithm run on ten paragraphs.

Finally, we calculated Spearman's ρ as the length of the paragraph increases, to get a sense of how aligned the two rankings are. Each data point is the average of the algorithm run on ten paragraphs.

5.5 Discussion

As can be seen in Figure 8, the runtime of the BH Flat Algorithm increases exponentially as the length of the paragraph increases. This makes it intractable for longer paragraphs. In contrast, the runtime of the BH Mult Algorithm increases slowly, so it is tractable for much longer paragraphs. Taking each sentence as its own paragraph,

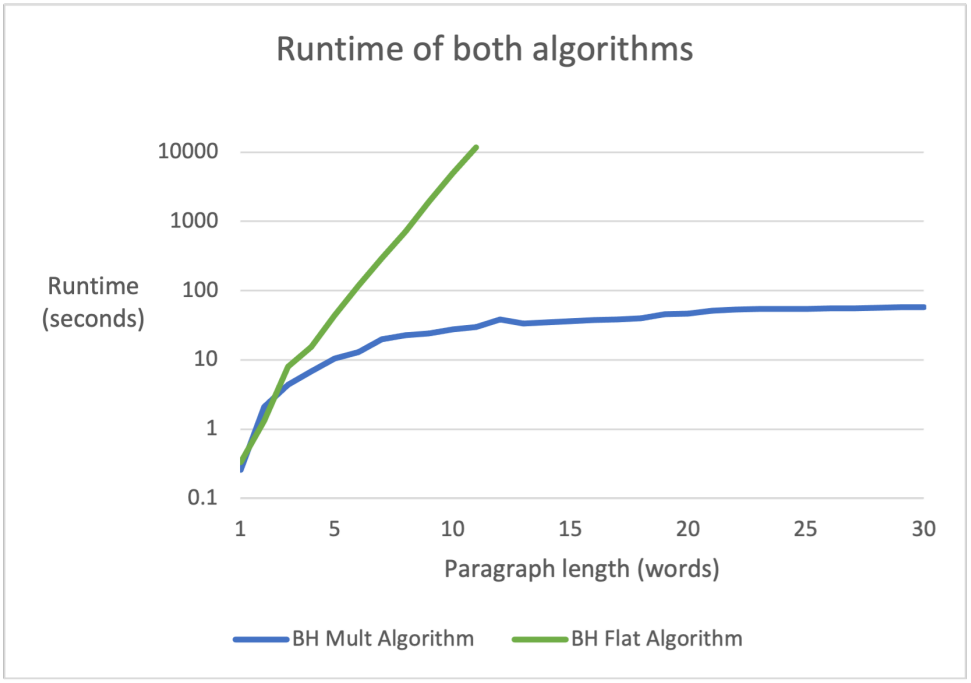


Figure 8: The runtime of the BH Mult and BH Flat Algorithms

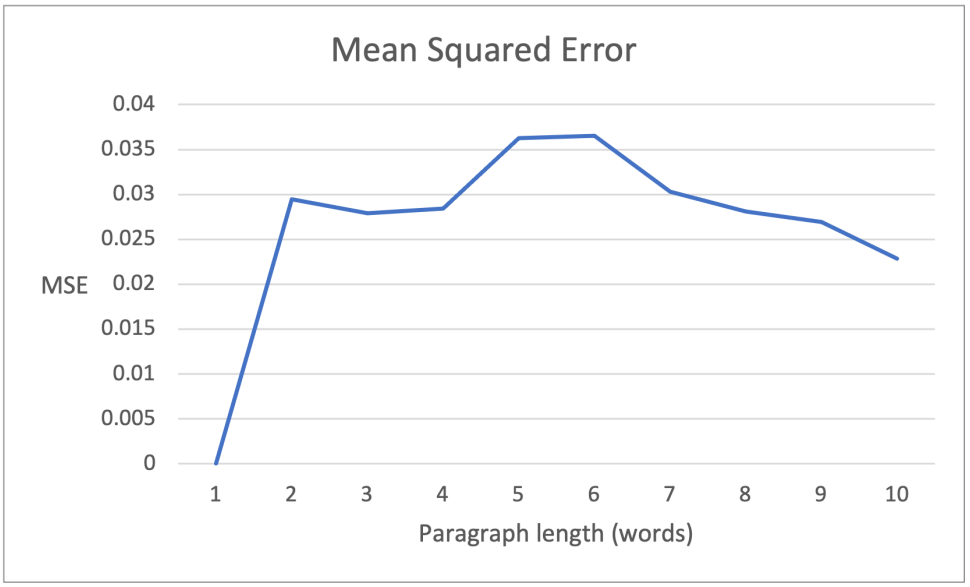


Figure 9: The Mean Squared Error between the outputs of BH Mult and BH Flat Algorithms

since the average sentence in the English language is 15-20 words, the BH Mult Algorithm allows for computing of many more sentences in the same amount of time. Furthermore, the BH Mult Algorithm could allow for multiple sentence paragraphs to become tractable, as most of the yelp reviews in our dataset were. This allows for words to be understood better in their full context, as opposed to in only the single sentence they are in. Thus, in that way, running the Mult BH Algorithm on longer paragraphs can allow for more nuanced understanding of the words - by examining longer paragraphs and more paragraphs total.

There is, however, a trade-off in terms of accuracy, as shown in Figures 9 and 10. The mean squared error increases until a paragraph length of 6 words, then decreases slightly, meaning the approximate algorithm gets farther away then closer to the true value as the length of the paragraph increases. It makes intuitive sense that to start, error would increase as sentences get slightly longer and the error from the approximation increases, but then as the paragraphs get much longer, there is more data for the algorithm take in, so it more accurate they become.

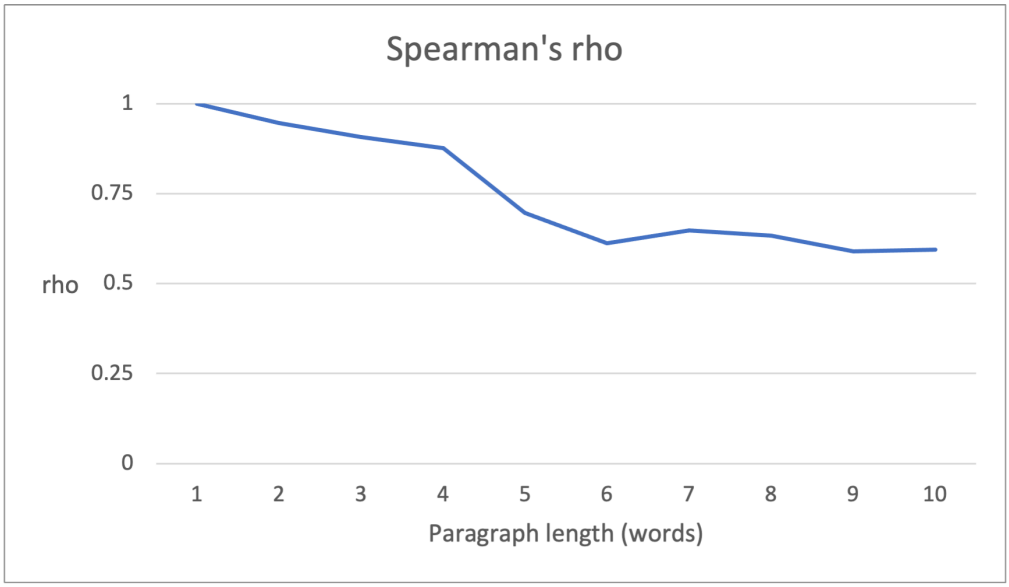


Figure 10: Spearman's ρ between the outputs of the BH Mult and BH Flat Algorithms

On the other hand, Spearman's ρ decreases slightly as the paragraph length increases, which makes sense - as the paragraphs get longer, there are more total words in the ranking, so there is more potential for small deviations in the rankings.

There is certainly a trade-off of accuracy for speed. More examination of the trade-off should be undertaken by anyone who wants to implement this for a natural language processing application, but in general, if a user decides that a mean squared error of about .03 (or lower, if longer paragraphs are used) and Spearman's ρ of about .6 is tolerable, then the Mult BH Algorithm provides a large speedup.

6 Conclusion

This paper presented a new algorithm, the BH Mult Algorithm, for calculating the Banzhaf power index on hierarchical voting games. The runtime of the new algorithm is in polynomial time, as opposed to exponential time needed to run the Banzhaf power index directly. As such, this allows us to calculate Banzhaf power for large games that were previously intractable.

We presented two specific cases where this result is useful. In the French Senate, we were able to calculate the voting power of voters in Toulouse and Saint Martin, which is an intractable problem when run with the flat Banzhaf power index. This result showed us that under this model, a voter in Saint Martin has about six times as much power as a voter in Toulouse. In the problem of determining the power of words for vocabulary selection, we were able to speed up the algorithm by several orders of magnitude even for small paragraphs (10 words long), while still keeping relatively high accuracy.

For future work, we suggest expanding this line of reasoning to other types of similar voting games. There are many different types of games that do not fall in the strict parameters we have laid out as a voting game. Some games aren't balanced. Some games have inputs and outputs that are not restricted to binary values, such as ternary voting games, which are games with three possible choices, with the third generally being abstaining. Finding a similar algorithmic speedup for those games would be useful in making their computation tractable. Also, the SSPI on hierarchical voting games should be studied with a similar line of reasoning, with the hopes of finding a way to calculate it much quicker.

References

[1] Amina Adadi and Mohammed Berrada. "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6 (2018), pp. 52138–52160. DOI: 10.1109/ACCESS.2018.2870052.

[2] Fuad Aleskerov, Valeriy Kalyagin, and Kirill Pogorelskiy. "Actual voting power of the IMF members based on their political-economic integration". In: *Annals of Economics and Statistics* 48.9 (2008), pp. 1554–1569. DOI: https://doi.org/10.1016/j.mcm.2008.05.020.

[3] E. Algaba, J. M. Bilbao, and J. R. Fernandez. "The distribution of power in the European constitution". In: *European Journal of Operational Research* 176.3 (2007), pp. 1752–1766. DOI: https://doi.org/10.1016/j.ejor.2005.12.002.

[4] John F. Banzhaf. "Weighted Voting Doesn't Work: A Mathematical Analysis". In: *Rutgers Law Review* 19 (1965), p. 317.

[5] Fabrice Barthelemy and Mathieu Martin. "A comparison between the methods of apportionment using power indices: the case of the US presidential elections". In: *Annals of Economics and Statistics* 101 (2011), pp. 87–106. DOI: https://doi.org/10.2307/41615475.

[6] J.C. Caruso and N. Cliff. "Empirical size, converage, and power of confidence intervals for Spearman's rho". In: *Educational and Psychological Measurement* 57 (1997), pp. 637–654. DOI: https://doi.org/10.1177/0013164497057004009.

[7] Wenhu Chen et al. "How Large a Vocabulary Does Text Classification Need? A Variational Approach to Vocabulary Selection". In: *CoRR* abs/1902.10339 (2019). arXiv: 1902.10339. URL: http://arxiv.org/abs/1902.10339.

[8] Pradeep Dubey, Ezra Einy, and Ori Haimanko. "Compound voting and the Banzhaf index". In: *Games and Economic Behavior* 51.1 (2005), pp. 20–30. DOI: https://doi.org/10.1016/j.geb.2004.03.002.

[9] Imre Ferto et al. "The power ranking of the members of the Agricultural Committee of the European Parliament". In: *European Review of Agricultural Economics* 47.5 (2020), pp. 1897–1919. DOI: https://doi.org/10.1093/erae/jbaa011.

[10] Ori Haimanko. "Composition independence in compound games: a characterization of the Banzhaf power index and the Banzhaf value". In: *International Journal of Game Theory* 48 (2019), pp. 755–768. DOI: https://doi.org/10.1007/s00182-019-00660-w.

[11] Adam Karczmarz et al. "Improved Feature Importance Computations for Tree Models: Shapley vs. Banzhaf". In: (2021). DOI: https://arxiv.org/abs/2108.04126.

[12] Werner Kirsch. "Brexit and the Distribution of Power in the Council of the EU". In: *CEPS* (2016). DOI: https://www.ceps.eu/ceps-publications/brexit-and-distribution-power-council-eu/.

[13] Roland Kirstein. "Volkswagen vs. Porsche: a power-index analysis". In: *International Journal of Corporate Governance* 2.1 (2010), pp. 1–20.

[14] Bettina Klinz and Gerhard J. Woeginger. "Faster algorithms for computing power indices in weighted voting games". In: *Mathematical Social Sciences* 49.1 (2005), pp. 111–116. DOI: https://doi.org/10.1016/j.mathsocsci.2004.06.002.

[15] Bogdan Kulynych and Carmela Tronscoso. "Feature importance scores and lossless feature pruning using Banzhaf power indices". In: *NIPS Symposium on Interpretable Machine Learning* (2017). DOI: https://arxiv.org/abs/1711.04992.

[16] L'express. "Results of the 2014 Municipal Elections". In: (2014). DOI: https://www.lexpress.fr/resultats-elections/municipales-2014-toulouse-31200_403989.html.

[17] Gurvan L'Hostis, David Grangier, and Michael Auli. "Vocabulary Selection Strategies for Neural Machine Translation". In: *CoRR* abs/1610.00072 (2016). arXiv: 1610.00072. URL: http://arxiv.org/abs/1610.00072.

[18] Ministere De L'interior. "Results of the 2014 Senatorial Elections". In: (2014). DOI: https://www.interieur.gouv.fr/Elections/Les-resultats/Senatoriales/elecresult__SN2014/(path)/SN2014/031/index.html.

[19] Ministere De L'interior. "Results of the 2020 Senatorial Elections". In: (2020). DOI: https://www.interieur.gouv.fr/Elections/Les-resultats/Senatoriales/elecresult__senatoriales-2020/(path)/senatoriales-2020/978/index.html.

[20] Dennis Leech. "The Relationship Between Shareholding Concentration and Shareholder Voting Power in British Companies: A Study of the Application of Power Indices for Simple Games". In: *Management Science* 34.4 (1988), pp. 509–527. DOI: https://doi.org/10.1287/mnsc.34.4.509.

[21] J. Von Neumann and O. Morgenstern. "Theory of Games and Economic Behavior". In: *Princeton University Press* (1944). DOI: https://press.princeton.edu/books/paperback/9780691130613/theory-of-games-and-economic-behavior.

[22] Roma Patel et al. "Game-theoretic Vocabulary Selection via the Shapley Value and Banzhaf Index". In: *NAACL-HLT*. 2021, pp. 2789–2798. URL: https://doi.org/10.18653/v1/2021.naacl-main.223.

[23] Stefan Prigge. "The Performance of Measures of Shareholder Influence". In: *SSRN Electronic Journal* (Feb. 2007). DOI: 10.2139/ssrn.966086.

[24] Peng Qi et al. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages". In: *Association for Computational Linguistics (ACL) System Demonstrations* (2020). URL: https://arxiv.org/abs/2003.07082.

[25] Amnon Rapoport and Esther Golan. "Assessment of Political Power in the Israeli Knesset". In: *American Political Science Review* 79.3 (1985), pp. 673–692. DOI: doi:10.2307/1956837.

[26] A. Saavedra-Nieves and M.G. Fiestras-Janeiro. "Sampling methods to estimate the Banzhaf-Owen value". In: *Ann Oper Res* 301 (May 2021), pp. 199–223. DOI: 10.1007/s10479-020-03614-8.

[27] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. "Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models". In: *CoRR* abs/1708.08296 (2017). arXiv: 1708.08296. URL: http://arxiv.org/abs/1708.08296.

[28] J Antionio Seijas-Marcias. "Power Index of Finnish Parties: Evolution of the Parliament System". In: *Finnish-German Yearbook of Political Economy* 41.61 (2019), pp. 41–61.

[29] French Senate. "2017 Senatoriales". In: (2017). DOI: http://senatoriales2017.senat.fr/#dashboard.

[30] French Senate. "The senatorial elections". In: (2022). DOI: https://www.senat.fr/lng/en/senators/the_senatorial_elections.html.

[31] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Neural Machine Translation of Rare Words with Subword Units". In: *CoRR* abs/1508.07909 (2015). arXiv: 1508.07909. URL: http://arxiv.org/abs/1508.07909.

[32] L. S. Shapley and Martin Shubik. "A Method for Evaluating the Distribution of Power in a Committee System". In: *American Political Science Review* 48.3 (1954), pp. 787–792. DOI: 10.2307/1951053.

[33] Xing Shi and Kevin Knight. "Speeding Up Neural Machine Translation Decoding by Shrinking Run-time Vocabulary". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 574–579. DOI: 10.18653/v1/P17-2091. URL: https://aclanthology.org/P17-2091.

[34] Frantisek Turnovec. "Fair Majorities in Proportional Voting". In: *Kybernetika* 49.3 (2013), pp. 498–505.

[35] John R. Wright. "Pivotal states in the Electoral College, 1880 to 2004". In: *Public Choice* 139 (2009), pp. 21–37. URL: https://link.springer.com/article/10.1007/s11127-008-9374-y.