Computation of the Johnston Power Index in Weighted Majority Games

Ingo Wilms[1]

Abstract

Power indices can be very useful to reveal the actual power of voters in a voting system. In weighted majority games each voter has a weight and the overall vote is "yes", if the sum of weights of all "yes"-voters is greater or equal the quota. One of the power indices that is regularly the subject of research is the Johnston index. In this paper I propose an algorithm for the computation of the Johnston power index in weighted majority games. The presented algorithm computes the index in pseudo-polynomial time, if the single weights are all integers.

Keywords: Game theory, Power indices, Johnston index, Generating functions, Weighted majority games

1. Introduction

Power indices can be used to describe the power of players in a voting system. Each player votes "yes" or "no" over a proposal. All "yes"-voters form a coalition and all "no"-voters form a coalition. According to predefined rules it is determined, which coalition is winning. Although the concept of power is somewhat abstract, in the domain of power indices a voter usually is said to have power, if the voter can influence the outcome with his vote (Penrose, 1946). In weighted majority games each player has a voting weight and a coalition is considered winning if the sum of the weights of all members in that coalition is equal or bigger than a certain quota q.

[1]ingo.wilms@gmail.com

Several power indices were suggested in the past. Two of the best researched and most often applied power indices are Banzhaf index (Penrose, 1946; Banzhaf III., 1965) and Shapley-Shubik index (Shapley & Shubik, 1954). The Deegan-Packel index (Deegan & Packel, 1978) and the Johnston index (Johnston, 1978) played a minor up to today. On the one hand the Johnston index has been criticized for some of its properties by Felsenthal & Machover (1998), but on the other hand Brams (2003) advocated the index especially for the measurement of presidential power. Brams (2003) applied the Johnston index to the United States federal system and Mercik (2009) applied the index to the polish president.

Although the Johnston index is regularly the subject of theoretical research (for instance Hirokawa & Vlach (2006), Lorenzo-Freire et al. (2007), Alonso-Meijide et al. (2008), Freixas et al. (2012), Freixas & Marciniak (2013) and Kurz (2016)), except for Alonso-Meijide et al. (2008) no effort has been made to establish computational methods. Alonso-Meijide et al. (2008) describe an algorithm for the computation of the Johnston index using the multilinear extension method introduced by Owen (1972). In general this method has its strength in approximating power indices for large number of players n as the exact computation has exponential time complexity.

Due to lack of exact computational methods even for moderate n it is not surprising there are only very few cases in the literature, where the Johnston index was applied to real world weighted majority games. Two examples are Strand (2003) and Strand & Rapkin (2005). In both cases the authors used an own program combining an enumeration method and the generating functions technique.

With this article I want to propose a new algorithm for the exact computation of the Johnston index in weighted majority games. For the calculation of Banzhaf index, Shapley-Shubik index and Deegan-Packel index for all n players there exist algorithms with time complexity $\mathcal{O}(nq)$, $\mathcal{O}(n^2q)$ and $\mathcal{O}(n^2q)$ respectively (Uno, 2003). I will use similar techniques to develop an algorithm for the computation of the Johnston index of all players within time complexity

$\mathcal{O}(n^3 q)$, which uses $\mathcal{O}(nq)$ memory.

2. Preliminaries

Let $N = \{p_1, p_2, \ldots, p_n\}$ be a set of players. Every subset $S \subseteq N$ is called a coalition. Let $w_i \in \mathbb{N}$ be the weight of player p_i, $i = 1, 2, \ldots, n$. Then we can define the weight function

$$w : \{S \subseteq N\} \to \mathbb{N}, S \mapsto w(S) := \sum_{p_i \in S} w_i.$$

A weighted majority game is represented by $[q; w_1, w_2, \ldots, w_n]$ where $q \in \mathbb{N}$ is the quota satisfying $\frac{1}{2}\sum_{i=1}^{n} w_i < q \leq w(N)$. A coalition $S \subseteq N$ is called winning if and only if $w(S) \geq q$. The set of all winning coalitions is denoted by W. A player p_i is said to be *crucial* for S, if $w(S) \geq q$, $p_i \in S$ and $w(S \setminus \{p_i\}) < q$. I will denote the set of all winning coalitions containing at least one crucial player with W_{qmin}.

Definition 1. Let $[q; w_1, w_2, \ldots, w_n]$ be a weighted majority game. The Johnston index of player p_j is defined by

$$\gamma_j \cdot |W_{\text{qmin}}| = \sum_{\substack{S \in W_{\text{qmin}}: \\ w(S \setminus p_j) < q}} \frac{1}{\#\{p_i \in S : w(S \setminus p_i) < q\}}. \tag{1}$$

3. Computation of the Johnston Index

Remark 2. Without loss of generality we assume $w_1 \geq w_2 \geq \cdots \geq w_n$. If the weights are not sorted, they have to be sorted by using a sorting algorithm. The time complexity of well known sorting algorithms is bounded by $\mathcal{O}(n \log n)$.

We define two subsets of N by

$$P_i = \{p_1, p_2, \ldots, p_i\}, \qquad \overline{P}_i = \{p_i, p_{i+1}, \ldots, p_n\}.$$

Also we define $P_0 = \overline{P}_{n+1} = \emptyset$.

The idea of the algorithm. We want to calculate the Johnston index γ_j for player p_j. The idea is to partition every winning coalition S, where p_j is crucial, into two disjoint subsets. Let i be the biggest index of a crucial player in S. We define two subsets of S by

$$S_1 = S \cap P_i \quad \text{and} \quad S_2 = S \cap \overline{P}_{i+1}.$$

It follows $S = S_1 \cup S_2$ and $S_1 \cap S_2 = \emptyset$. Due to sorted weights every player in S_1 is crucial while no player of S_2 is. So, for $i = j, j+1, \ldots, n$ we pre-calculate the numbers of all possible coalitions $S_1 \subseteq P_i$ with $\{p_i, p_j\} \subseteq S_1$. Also, for each combination of i and $w(S_1)$ we pre-calculate the numbers of all possible coalitions $S_2 \subseteq \overline{P}_{i+1}$ so that following properties hold

1. $S = S_1 \cup S_2$ is a winning coalition,
2. p_i is crucial and
3. no player in S_2 is crucial.

Once we have done that, we can calculate γ_j from

$$\gamma_j \cdot |W_{\text{qmin}}| = \sum_{i=j}^{n} \sum_{y=w(\{p_i,p_j\})}^{q+w_i-1} \sum_{k=|\{p_i,p_j\}|}^{i} \frac{\#\{S_1 : |S_1| = k, w(S_1) = y\} \cdot \#\{S_2\}}{k},$$

where S_1 and S_2 fulfil the above properties. The limits of the sums are chosen according to our requirements. For counting possible coalitions S_1 we will introduce functions f and f_j. For counting possible coalitions S_2 we will introduce functions c, d and h. The functions f, c and d were already used by Uno (2003).

I will start with showing how to count the relevant coalitions S_2 for given i and $y = w(S_1)$. Let c be a function defined by

$$c(i, y) = \#\{S \subseteq \overline{P}_i : w(S) = y\}.$$

Then $c(i, y)$ indicates the number of subsets in $\overline{P}_i$ with weight y. For i, y with

$1 \leq i \leq n$ and $0 \leq y \leq w(N)$ we derive the property

$$\begin{aligned} c(i,y) &= \#\{S \subseteq \overline{P}_i : w(S) = y\} \\ &= \#\{S \subseteq \overline{P}_i : p_i \in S, w(S) = y\} + \#\{S \subseteq \overline{P}_{i+1} : w(S) = y\} \\ &= \#\{S \subseteq \overline{P}_{i+1} : w(S) = y - w_i\} + \#\{S \subseteq \overline{P}_{i+1} : w(S) = y\} \\ &= \begin{cases} c(i+1,y) + c(i+1,y-w_i) & \text{if } y \geq w_i, \\ c(i+1,y) & \text{if } y < w_i. \end{cases} \end{aligned}$$

With this recursion we can calculate all $c(i,y)$ for $i \in \{1,2,\ldots,n+1\}$ and $y \in \{0,1,\ldots,\min\{w(\overline{P}_i), q-w_i\}\}$ from the the initial values $c(n+1,0) = 1$ and $c(n+1,y) = 0$ for $y > 0$. The time complexity for that is $\mathcal{O}(nq)$.

Let now d be a function defined by

$$d(i,z) = \begin{cases} \#\{S \subseteq \overline{P}_i : w(S) \leq z\} & \text{for } z \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Using the recursion

$$d(i,z) = \sum_{y=0}^{z} c(i,y) = \sum_{y=0}^{z-1} c(i,y) + c(i,z) = d(i,z-1) + c(i,z)$$

all values of $d(i,z)$ for $i \in \{1,2,\ldots,n+1\}$ and $z \in \{1,2,\ldots,\min\{w(\overline{P}_i), q-1\}\}$ can be calculated in $\mathcal{O}(nq)$.

Now let h be a function defined by

$$h(i,y) = \#\{S \subseteq \overline{P}_{i+1} : y + w(S) - \max_{p_k \in S} w_k \geq q, y + w(S) - w_i < q\}.$$

Then h is exactly the number of S_2 for given i and $w(S_1) = y$ that fulfil the properties above.

Proposition 3. *Let $G = [q; w_1, w_2, \ldots, w_n]$ be a weighted majority game with $w_1 \geq w_2 \geq \cdots \geq w_n$. Then*

$$\begin{aligned} h(i,y) = \sum_{j=i+1}^{n} \Big[& d(j+1, \min\{q + w_i - w_j - 1 - y, w(\overline{P}_{j+1})\}) \\ & - d(j+1, \max\{0, q-y\} - 1)\Big] + \begin{cases} 1 & \textit{if } q + w_i - 1 \geq y \textit{ und } y \geq q \\ 0 & \textit{otherwise.} \end{cases} \end{aligned}$$

Proof. We write $h(i,y)$ as

$$\begin{aligned} h(i,y) &= \#\{S \subseteq \overline{P}_{i+1} : y + w(S) - \max_{p_k \in S} w_k \geq q, y + w(S) - w_i < q\} \\ &= \#\{S \subseteq \overline{P}_{i+1} : y + w(S) - \max_{p_k \in S} w_k \geq q, y + w(S) - w_i \leq q - 1\} \\ &= \#\{S \subseteq \overline{P}_{i+1} : q + w_i - 1 - y \geq w(S) \geq q - y + \max_{p_k \in S} w_k\}. \end{aligned}$$

Using the identity $\{S \subseteq \overline{P}_{i+1}\} = \bigcup_{j=i+1}^{n+1} \{S \subseteq \overline{P}_j : p_j \in S\}$ we obtain

$$\begin{aligned} &= \#\{ \bigcup_{j=i+1}^{n+1} \{S \subseteq \overline{P}_j : p_j \in S, q + w_i - 1 - y \geq w(S) \geq q - y + w_j\}\} \\ &= \#\{ \bigcup_{j=i+1}^{n} \{S \subseteq \overline{P}_j : p_j \in S, q + w_i - 1 - y \geq w(S) \geq q - y + w_j\} \\ &\quad \cup \{S : S = \emptyset, q + w_i - 1 - y \geq 0 \geq q - y\}\} \\ &= \#\{ \bigcup_{j=i+1}^{n} \{S \cup p_j : S \subseteq \overline{P}_{j+1}, q + w_i - w_j - 1 - y \geq w(S) \geq q - y\} \\ &\quad \cup \{S : S = \emptyset, q + w_i - 1 - y \geq 0 \geq q - y\}\}. \end{aligned}$$

The $\{S : S \subseteq \overline{P}_j, p_j \in S\}$ are pairwise distinct for different j. Therefore

$$\begin{aligned} &= \sum_{j=i+1}^{n} \#\{S \subseteq \overline{P}_{j+1} : q + w_i - w_j - 1 - y \geq w(S) \geq q - y\} \\ &\quad + \#\{S : S = \emptyset, q + w_i - 1 - y \geq 0 \geq q - y\}. \end{aligned}$$

Now substituting d into the equation completes the proof. □

This means, after computing d we can compute $h(i,y)$ for $i \in \{1, 2, \ldots, n\}$ and $y \in \{w_i, w_i + 1, \ldots, \min\{q + w_i - 1, w(P_i)\}\}$ in $\mathcal{O}(n)$. Therefore, all values of $h(i,y)$ can be computed in $\mathcal{O}(n^2 q)$.

I will now address how to count the numbers of $S_1 \subseteq P_i$ with $p_i \in S_1$ and $p_j \in S_1$. For that we will use two functions f and f_j. Let f be a function

defined by

$$f(i,k,y)$$
$$= \begin{cases} \#\{S : S \subseteq P_i, w(S) = y, |S| = k\} & \text{for } y \in \{0,1,\ldots,w(P_i)\}, \\ 0 & \text{otherwise.} \end{cases}$$

f is indicating the number of coalitions in P_i for given weight y and number of players k. For i,k,y with $1 \leq i \leq n$, $1 \leq k \leq i$ and $0 \leq y \leq w(P_i)$ we can derive the following property

$$\begin{aligned} f(i,k,y) &= \#\{S : S \subseteq P_i, w(S) = y, |S| = k\} \\ &= \#\{S : S \subseteq P_{i-1}, w(S) = y, |S| = k\} \\ &\quad + \#\{S : S \subseteq P_i, p_i \in S, w(S) = y, |S| = k\} \\ &= \#\{S : S \subseteq P_{i-1}, w(S) = y, |S| = k\} \\ &\quad + \#\{S : S \subseteq P_{i-1}, w(S) = y - w_i, |S| = k-1\} \\ &= \begin{cases} f(i-1,k,y) + f(i-1,k-1,y-w_i) & \text{if } y \geq w_i, \\ f(i-1,k,y) & \text{if } y < w_i. \end{cases} \end{aligned}$$

With the initial values $f(i,k,0) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{else.} \end{cases}$ we can calculate $f(i,k,y)$ for all $(i,k,y) \in \{0,1,\ldots,n\} \times \{0,1,\ldots,i\} \times \{0,1,\ldots,q+w_1-1\}$ in $\mathcal{O}(n^2 q)$.

Now let f_j be a function defined by

$$f_j(i,k,y)$$
$$= \begin{cases} \#\{S : S \subseteq P_i \setminus \{p_j\}, w(S) = y, |S| = k\} & \text{for } y \in \{0,1,\ldots,w(P_i \setminus \{p_j\})\}, \\ 0 & \text{otherwise.} \end{cases}$$

f_j is corresponding to $f(i,k,y)$ setting $w_j = 0$. Completely analogous to before

we can show the property

$$f_j(i,k,y) = \begin{cases} f_j(i-1,k,y) & \text{for } i = j, \\ \begin{cases} f_j(i-1,k,y) + f_j(i-1,k-1,y-w_i) & \text{if } y \geq w_i, \\ f_j(i-1,k,y) & \text{if } y < w_i. \end{cases} & \text{for } i \neq j. \end{cases}$$

Obviously we can calculate $f_j(i,k,y)$ for all $(i,k,y) \in \{0,1,\ldots,n\} \times \{0,1,\ldots,i\} \times \{0,1,\ldots,q+w_1-1\}$ also in $\mathcal{O}(n^2 q)$.

Lemma 4. *Let $[q; w_1, w_2, \ldots, w_n]$ be a weighted majority game with $i, j \in \{1,2,\ldots,n\}$ with $i \geq j$. Then*

$$\begin{aligned} &\#\{S \subseteq P_i : \{p_i, p_j\} \subseteq S\} \\ &= f(i,k,y) - f_j(i,k,y) - f(i-1,k,y) + f_j(i-1,k,y). \end{aligned}$$

Proof. Consider the identity

$$\begin{aligned} &\{S : S \subseteq P_i\} \\ &= \{S : S \subseteq P_i \setminus \{p_i\}\} \cup \{S : S \subseteq P_i \setminus \{p_j\}\} \cup \{S : S \subseteq P_i, \{p_i, p_j\} \subseteq S\}, \end{aligned}$$

or rather the identity of the cardinalities

$$\begin{aligned} &|\{S : S \subseteq P_i\}| \\ &= |\{S : S \subseteq P_i \setminus \{p_i\}\}| + |\{S : S \subseteq P_i \setminus \{p_j\}\}| - |\{S : S \subseteq P_i \setminus \{p_i, p_j\}\}| \\ &\quad + |\{S : S \subseteq P_i, \{p_i, p_j\} \subseteq S\}|. \end{aligned}$$

If we restrict all sets S to $|S| = k, w(S) = y$ we have

$$\begin{aligned} &|\{S : S \subseteq P_i, \{p_i, p_j\} \subseteq S, |S| = k, w(S) = y\}| \\ &= |\{S : S \subseteq P_i, |S| = k, w(S) = y\}| \\ &\quad + |\{S : S \subseteq P_i \setminus \{p_i, p_j\}, |S| = k, w(S) = y\}| \\ &\quad - |\{S : S \subseteq P_i \setminus \{p_i\}, |S| = k, w(S) = y\}| \\ &\quad - |\{S : S \subseteq P_i \setminus \{p_j\}, |S| = k, w(S) = y\}| \\ &= f(i,k,y) + f_j(i-1,k,y) - f(i-1,k,y) - f_j(i,k,y). \end{aligned}$$

□

Theorem 5. *Let* $[q; w_1, w_2, \ldots, w_n]$ *be a weighted majority game with* $w_1 \geq w_2 \geq \cdots \geq w_n]$. *Then*

$$\gamma_j = \sum_{i=j}^{n} \sum_{y=w(\{p_i,p_j\})}^{q+w_i-1} \sum_{k=|\{p_i,p_j\}|}^{i} [f(i,k,y) - f_j(i,k,y) - f(i-1,k,y) + f_j(i-1,k,y)] \tfrac{1}{k} h(i,y).$$

Proof. We rewrite the definition of the Johnston index for player p_j

$$\begin{aligned} &\gamma_j \cdot \#\{S \subseteq N : w(S) \geq q, \exists p_k \in S : w(S \setminus p_k) < q\} \\ &= \sum_{S \subseteq N : w(S) \geq q, p_j \in S, w(S \setminus p_j) < q} \frac{1}{\#\{p_k \in S : w(S \setminus p_k) < q\}}. \end{aligned}$$

Partitioning $S = S_1 \cup S_2$ as described before leads to

$$\begin{aligned} &= \sum_{i=j}^{n} \sum_{\substack{(S_1,S_2): \\ S_1 \subseteq P_i : \{p_i,p_j\} \subseteq S_1, \\ S_2 \subseteq \overline{P}_{i+1} : p_k \in S_2 \Rightarrow w(S \setminus p_k) \geq q, \\ q + w_i - 1 \geq w(S_1 \cup S_2) \geq q}} \frac{1}{|S_1|} \\ &= \sum_{i=j}^{n} \sum_{\substack{S_1 \subseteq P_i : \{p_i,p_j\} \subseteq S_1, \\ q + w_i - 1 \geq w(S_1) \geq w(\{p_i,p_j\})}} \frac{1}{|S_1|} \# \left\{ \begin{array}{c} S_2 \subseteq \overline{P}_{i+1}: \\ q + w_i - 1 \geq w(S_1) + w(S_2) \geq q, \\ p_k \in S_2 \Rightarrow w(S_1) + w(S_2) - w_k \geq q \end{array} \right\}. \end{aligned}$$

Introducing weight $y = w(S_1)$ we can write

$$= \sum_{i=j}^{n} \sum_{y=w(\{p_i,p_j\})}^{q+w_i-1} \sum_{\substack{S_1 \subseteq P_i : \\ \{p_i,p_j\} \subseteq S_1, \\ w(S_1) = y}} \frac{1}{|S_1|} \# \left\{ \begin{array}{c} S_2 \subseteq \overline{P}_{i+1}: \\ q + w_i - 1 \geq y + w(S_2) \geq y + w(S_2) - w_k \geq q \; \forall p_k \in S_2 \end{array} \right\}$$

and further introducing coalition size $k = |S_1|$ we write

$$\begin{aligned} &= \sum_{i=j}^{n} \sum_{y=w(\{p_i,p_j\})}^{q+w_i-1} \sum_{k=|\{p_i,p_j\}|}^{i} \sum_{\substack{S_1 \subseteq P_i : \{p_i,p_j\} \subseteq S_1, \\ w(S_1) = y, |S_1| = k}} \frac{1}{k} \# \left\{ \begin{array}{c} S_2 \subseteq \overline{P}_{i+1}: \\ q + w_i - 1 \geq y + w(S_2) \\ \geq y + w(S_2) - \max\limits_{p_k \in S_2} w_k \geq q \end{array} \right\} \\ &= \sum_{i=j}^{n} \sum_{y=w(\{p_i,p_j\})}^{q+w_i-1} \sum_{k=|\{p_i,p_j\}|}^{i} \sum_{\substack{S_1 \subseteq P_i : \{p_i,p_j\} \subseteq S_1, \\ w(S_1) = y, |S_1| = k}} \frac{1}{k} \# \left\{ \begin{array}{c} S_2 \subseteq \overline{P}_{i+1}: \\ q + w_i - 1 - y \geq w(S_2) \\ \geq q - y + \max\limits_{p_k \in S_2} w_k \end{array} \right\}. \end{aligned}$$

Substituting f and f_j into the equation using lemma 4 and bringing in function h leads us to

$$= \sum_{i=j}^{n} \sum_{y=w(\{p_i,p_j\})}^{q+w_i-1} \sum_{k=|\{p_i,p_j\}|}^{i} [f(i,k,y) - f_j(i,k,y) - f(i-1,k,y) + f_j(i-1,k,y)] \tfrac{1}{k} h(i,y).$$

□

Corollary 6. *Let* $[q; w_1, w_2, \ldots, w_n]$ *be a weighted majority game with* $w_1 \geq w_2 \geq \cdots \geq w_n$. *The Johnston index for all* n *players can be calculated in* $\mathcal{O}(n^3 q)$ *time with* $\mathcal{O}(nq)$ *space.*

Proof. As $f(i,k,y)$, $f_j(i,k,y)$ and $h(i,y)$ are all computed in $\mathcal{O}(n^2 q)$ time, γ_j can be computed in $\mathcal{O}(n^2 q)$ using theorem 5. Iterating $j = 1, 2, \ldots, n$ we can therefore calculate the indices of all players in $\mathcal{O}(n^3 q)$.

The 3-dimensional arrays f and f_j need $\mathcal{O}(n^2 q)$ space. Luckily we can calculate all relevant values $f(i,\cdot,\cdot)$ completely from $f(i-1,\cdot,\cdot)$. So if we iterate $i = j, j+1, \ldots, n$ during the calculation of γ_j we can use 2-dimensional arrays for given i and delete $f(i-1,\cdot,\cdot)$ once we have calculated $f(i,\cdot,\cdot)$. The same holds for f_j. □

4. Applications

Algorithm 1 in the appendix shows how to calculate the Johnston index using presented methods. I will call this algorithm `Efficient`. In this section I want to discuss some of my applications of the algorithm. For comparison I have also implemented two enumeration methods, which work based on minimal-winning coalitions (in minimal-winning coalitions all players are critical). All presented times were measured on my notebook, which is nothing special (Intel processor i5-3340M with 2.70 GHz and 8 GB RAM).

Of course every algorithm has its strengths and weaknesses. For example, for $q > 2^n$ simple enumeration methods outperform `Efficient`. Also in situations like the inter-americal development bank (Strand, 2003) and the IMF (Strand & Rapkin, 2005), where the shareholder of banks have a very high absolute number of votes leading to a big q, `Efficient` becomes slow, although $n = 14$ and $n = 24$ respectively are small. Here my enumeration algorithms work faster, as the number of minimal-winning coalitions is smaller than q.

`Efficient` has its strengths, roughly spoken, for moderate q compared to n and when the players have distinct weights. For example, I will now define the game $G = [\left\lfloor \frac{n(n+1)}{4} + 1 \right\rfloor; n, n-1, \ldots, 1]$. This game has all numbers from

1 to n as weights and the quota is just above the half of the sum of all weights. Figure 1 shows the time necessary for the computation of the Johnston indices of all players for varying n. The diagram shows that the algorithm in this case is well applicable for n up to 200, where the enumeration algorithms stand no chance going through all the $1.56 \cdot 10^{57}$ minimal-winning coalitions.

The case of the United States Electoral College in 2016 could be executed in $< 0.03\,\mathrm{s}$ using `Efficient`. One of my two implemented enumeration methods needed circa 30 s, while the other would have run for months.

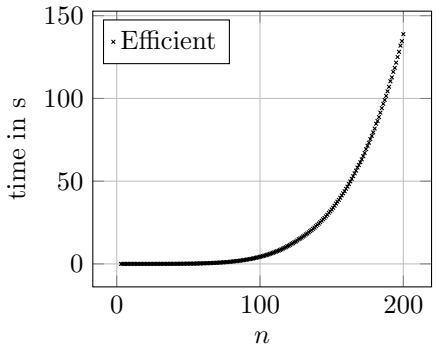

Figure 1: Execution time of algorithm 1 for $G = [\left\lfloor \frac{n(n+1)}{4} + 1 \right\rfloor; n, n-1, \ldots, 1]$ at different n.

5. Conclusions

For the case of weighted majority games a computational method was proposed that computes the Johnston index for all players. The time complexity and memory space complexity is bounded by $\mathcal{O}(n^3 q)$ and $\mathcal{O}(nq)$ respectively. Some examples were discussed regarding the performance. With this method at hand practical applications of the Johnston index can be easier in the future.

References

References

Alonso-Meijide, J. M., Casas-Méndez, B., Holler, M., & Lorenzo-Freire, S. (2008). Computing power indices: Multilinear extensions and new characterizations. *European Journal of Operational Research*, *188*, 540–554.

Banzhaf III., J. F. (1965). Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, *19*, 317–343.

Brams, S. J. (2003). *Negotiation games: applying game theory to bargaining and arbitration*. (2nd ed.). Routledge.

Deegan, J., & Packel, E. W. (1978). A new index of power for simple n-person games. *International Journal of Game Theory*, *7*, 113–123.

Felsenthal, D. S., & Machover, M. (1998). *The Measurement of Voting Power: Theory and Practice, Problems and Paradoxes.*. Cheltenham: Edward Elgar.

Freixas, J., & Marciniak, D. (2013). Egalitarian property for power indices. *Social Choice and Welfare*, *40*, 207–227.

Freixas, J., Marciniak, D., & Pons, M. (2012). On the ordinal equivalence of the johnston, banzhaf and shapley power indices. *European Journal of Operational Research*, *216*, 367–375.

Hirokawa, M., & Vlach, M. (2006). Power analysis of voting by count and account. *Kybernetika*, *42*, 483–493.

Johnston, R. (1978). On the measurement of power: Some reactions to laver. *Environment and Planning A*, *10*, 907–914.

Kurz, S. (2016). The inverse problem for power distributions in committees. *Social Choice and Welfare*, *47*, 65–88.

Lorenzo-Freire, S., Alonso-Meijide, J. M., Casas-Méndez, B., & Fiestras-Janeiro, M. (2007). Characterizations of the deegan–packel and johnston power indices. *European Journal of Operational Research*, *177*, 431–444.

Mercik, J. (2009). A priori veto power of the president of poland. *Operations Research and Decisions*, *19*, 61–75.

Owen, G. (1972). Multilinear extensions of games. *Management Science*, *18*, 64–79.

Penrose, L. S. (1946). The elementary statistics of majority voting. *Journal of the Royal Statistical Society*, *109*, 53–57.

Shapley, L. S., & Shubik, M. (1954). A method of evaluating the distribution of power in a committee system. *American Political Science Review*, *48*, 787–792.

Strand, J. R. (2003). Measuring voting power in an international institution: the united states and the inter-american development bank. *Economics of Governance*, *4*, 19–36.

Strand, J. R., & Rapkin, D. P. (2005). Regionalizing multilateralism: Estimating the power of potential regional voting blocs in the imf. *International Interactions*, *31*, 15–54.

Uno, T. (2003). *Efficient Computation of Power Indices for Weighted Majority Games*. Technical Report National Institute of Informatics Tokyo.

Appendix

Algorithm 1

Input: $[q; w_1, w_2, \ldots, w_n]$ *with* $w_1 \geq w_2 \geq \cdots \geq w_n$.

Calculate c

`For` $(i, y) \in \{1, 2, \ldots, n+1\} \times \{0, 1, \ldots, q-1\}$ `set` $c(i, y) := 0$

$c(n+1, 0) := 1$

```
For i = n, n − 1, . . . , 3
    For y = 0, 1, . . . , min{w(P̄_i), q − 1}
        If y ≥ w_i
            c(i, y) := c(i + 1, y) + c(i + 1, y − w_i)
        Else
            c(i, y) := c(i + 1, y)
        End If
    End For
End For
Calculate d
For i = n + 1, n, . . . , 3
    d(i, 0) := c(i, 0)
    For y = 1, 2, . . . , min{w(P̄_i), q − 1}
        d(i, y) := d(i, y − 1) + c(i, y)
    End For
End For
Calculate h
For y = q, q + 1, . . . , min{q + w_n − 1, w(N)}
    h(n, y) := 1
End For
For i = n − 1, n − 2, . . . , 1
    For y = w_i, w_i + 1, . . . , min{q + w_i − 1, w(P_i)}
        If q + w_i − 1 − y ≥ 0 ≥ q − y
            h(i, y) := 1
        End If
        For j = i + 1, i + 2, . . . , n
            y_0 := max{0, q − y} − 1
            y_0 := min{y_0, w(P̄_{j+1})}
            y_1 := min{q + w_i − w_j − 1 − y, w(P̄_{j+1})}
            If y_1 > y_0
                h(i, y) := h(i, y) + d(j + 1, y_1) − d(j + 1, y_0)
            End If
        End For
```

```
    End For
End For
Calculate all γ_j
For j = 1, 2, ..., n
    γ_j := 0
    For (k, y) ∈ {0, 1, ..., n} × {0, 1, ..., q − 1} set f(k, y) := 0, f_j(k, y) := 0
    For i = 1, 2, ..., n
        Calculate f(i − 1, ·, ·), f_j(i − 1, ·, ·)
        If i = 1
            f(0, 0) := 1, f_j(0, 0) := 1
        Else
            For k = i − 1, i − 2, ..., 1
                For y = q − 1, q − 2, ..., w_{i−1}
                    f(k, y) := f(k, y) + f(k − 1, y − w_{i−1})
                    If i − 1 ≠ j
                        f_j(k, y) := f_j(k, y) + f_j(k − 1, y − w_{i−1})
                    End If
                End For
            End For
        End If
        Update γ_j
        If j ≤ i ≤ n
            For k = |{p_i, p_j}|, ..., i
                For y = w({p_i, p_j}), ..., min{q + w_i − 1, w(P_i)}
                    If h(i, y) > 0
                        f_val := f(k − 1, y − w_i)
                        If i > j And y ≤ w(P_i \ {p_j})
                            f_val := f_val − f_j(k − 1, y − w_i)
                        End If
                        γ_j := γ_j + (1/k) f_val · h(i, y)
                    End If
                End For
            End For
```

```
        End If
    End For
End For
For i = 1, 2, ..., n
    γ_j := γ_j / |W_qmin|
End For
```

Output: $\gamma_1\gamma_2, \ldots, \gamma_n$.

The last step can be done by simply normalizing the indices or one can also use the identity

$$|W_{\mathrm{qmin}}| = \sum_{i=1}^{n} (d(i+1, q-1) - d(i+1, q-w_i-1)),$$

which is only valid for sorted weights $w_1 \geq w_2 \geq \cdots \geq w_n$ and is also useful for validation purposes. Note that we need to calculate c and d down to $i = 1$ for this.

The algorithm is very basic. For instance, obviously one do not need to calculate γ_i for $i > 1$ if $w_i = w_{i-1}$.