games

MDPI

Article

Monte Carlo Methods for the Shapley–Shubik Power Index †

Yuto Ushioda, Masato Tanaka and Tomomi Matsui *

Graduate School of Engineering, Tokyo Institute of Technology, Ookayama, Meguro-ku, Tokyo 152-8552, Japan; yutoyuto0928@gmail.com (Y.U.); adgjmpt.w02020202@gmail.com (M.T.)
* Correspondence: matsui.t.af@m.titech.ac.jp

† This paper is an extended version of our paper published in the 21st Japan-Korea Joint Workshop on Algorithms and Computation (WAAC), Fukuoka, Japan, 26–27 August 2018.

Abstract: This paper deals with the problem of calculating the Shapley–Shubik power index in weighted majority games. We propose an efficient Monte Carlo algorithm based on an implicit hierarchical structure of permutations of players. Our algorithm outputs a vector of power indices preserving the monotonicity, with respect to the voting weights. We show that our algorithm reduces the required number of samples, compared with the naive algorithm.

Keywords: voting game; weighted majority game; power index; Monte Carlo algorithm

check for updates

Citation: Ushioda, Y.; Tanaka, M.; Matsui, T. Monte Carlo Methods for the Shapley–Shubik Power Index. *Games* **2022**, *13*, 44. https://doi.org/10.3390/g13030044

Academic Editor: Ulrich Berger

Received: 16 April 2022
Accepted: 26 May 2022
Published: 2 June 2022

1. Introduction

The analysis of power is a central issue in political science. In general, it is difficult to define the idea of power, even in restricted classes of the voting rules commonly considered by political scientists. The use of game theory to study the power distribution in voting systems can be traced back to the invention of "simple games" by von Neumann and Morgenstern [1]. A simple game is an abstraction of the constitutional political machinery for voting.

In 1954, Shapley and Shubik [2] proposed the specialization of the Shapley value [3] to assess the a priori measure of the power of each player in a simple game. Since then, the Shapley–Shubik power index (S–S index) has become widely known as a mathematical tool for measuring the relative power of the players in a simple game.

In this paper, we consider a special class of simple games, called *weighted majority games*, which constitute a familiar example of voting systems. Let N be a set of players. Each player $i \in N$ has a positive integer voting weight w_i as the number of votes or weight of the players. A positive integer q is the quota needed for a coalition to win. A coalition $N' \subseteq N$ is a *winning coalition* if $\sum_{i \in N'} w_i \geq q$ holds; otherwise, it is a *losing coalition*. Many works analyze weighted majority games in a variety of settings, including the Council of the European Union [4–6], the U.S. Electoral College [7,8], and the International Monetary Fund [9,10]. Weighted majority games and power indices are applicable beyond classical voting situations in politics. Applications of power indices include cost allocation [11], analyses of genetic networks [12], analyses of social networks [13,14], and reliability problems in the maintenance of computer networks [15].

The difficulty involved in calculating the S–S index in weighted majority games is described in [16] without a proof (see p. 280, problem [MS8]). Deng and Papadimitriou [17] showed the problem of computing the S–S index in weighted majority games to be #P-complete. Prasad and Kelly [18] proved the NP-completeness of the problem of verifying the positivity of a given player's S–S index in weighted majority games. The problem of verifying the asymmetry of a given pair of players was also shown to be NP-complete [19]. It is known that even approximating the S–S index within a constant factor is intractable, unless P $=$ NP [20].

There are variations of methods for calculating the S–S index. These include algorithms based on the Monte Carlo method [21–26], multilinear extensions [27,28], dynamic programming [22,29–32], generating functions [33], binary decision diagrams [34], the Karnaugh map [35], relation algebra [36], or the enumeration technique [37]. A survey of algorithms for calculating power indices in weighted voting games is presented in [22].

In the classical theory of cooperative games, it is assumed that all players can communicate freely. Alonso-Meijide et al. [38] consider situations where some players are incompatible, i.e., some players cannot cooperate among themselves for ideological or economic reasons. They proposed a method for calculating the S—S and Banzhaf–Coleman power indices when some players are incompatible, using generating functions. Courtin et al. [39] discussed multi-type games in which there are several non-ordered types in the input, while the output consists of a single real value. When the output is dichotomous, they extend and fully characterize the S–S index.

This paper addresses Monte Carlo algorithms for calculating the S–S index in weighted majority games. In the following section, we describe the notations and definitions used in this paper. In Section 3, we analyze a naive Monte Carlo algorithm (Algorithm 1) and extend some results obtained in the study reported in [25]. In Section 4, we propose an efficient Monte Carlo algorithm (Algorithm 2) and show that our algorithm reduces the required number of samples compared to the naive algorithm. Table 1 summarizes the results of this study, where $(\varphi_1, \varphi_2, \dots, \varphi_n)$ denotes the S–S index and $(\varphi_1^{\mathrm{A}}, \varphi_2^{\mathrm{A}}, \dots, \varphi_n^{\mathrm{A}})$ denotes the estimator obtained by Algorithm 1 or 2.

Table 1. Required Number of Samples.

Property	Required Number of Samples	
	Algorithm 1 (Naive Algorithm)	**Algorithm 2 (Our Algorithm)**
$\Pr\left[\left\lvert\varphi_i^{\mathrm{A}} - \varphi_i\right\rvert < \varepsilon\right] \geq 1-\delta$	$\dfrac{\ln 2 + \ln(1/\delta)}{2\varepsilon^2}$ [25]	$\left(\dfrac{\ln 2 + \ln(1/\delta)}{2\varepsilon^2}\right)\left(\dfrac{1}{i^2}\right)$ (assume $w_1 \geq \cdots \geq w_n$)
$\Pr\left[\forall i \in N, \left\lvert\varphi_i^{\mathrm{A}} - \varphi_i\right\rvert < \varepsilon\right] \geq 1-\delta$	$\dfrac{\ln 2 + \ln(1/\delta) + \ln n}{2\varepsilon^2}$	$\dfrac{\ln 2 + \ln(1/\delta) + \ln 1.129}{2\varepsilon^2}$
$\Pr\left[\dfrac{1}{2}\sum_{i \in N}\left\lvert\varphi_i^{\mathrm{A}} - \varphi_i\right\rvert < \varepsilon\right] \geq 1-\delta$	$\dfrac{n \ln 2 + \ln(1/\delta)}{2\varepsilon^2}$	$\dfrac{n'' \ln 2 + \ln(1/\delta)}{2\varepsilon^2}$

An integer n'' denotes the size of a maximal player subset with mutually different weights.

2. Notations and Definitions

In this paper, we consider a special class of cooperative games called *weighted majority games*. Let $N = \{1, 2, \dots, n\}$ be a set of *players*. A subset of players is called a *coalition*. A weighted majority game G is defined by a sequence of positive integers $G = [q; w_1, w_2, \dots, w_n]$, where we may think of w_i as the number of votes or the weight of player i, and q as the quota needed for a coalition to win. In this paper, we assume that $0 < q \leq w_1 + w_2 + \cdots + w_n$.

A coalition $S \subseteq N$ is called a *winning coalition* when the inequality $q \leq \sum_{i \in S} w_i$ holds. The inequality $q \leq w_1 + w_2 + \cdots + w_n$ implies that N is a winning coalition. A coalition S is called a *losing coalition* if S is not winning. We define an empty set as a losing coalition.

Let $\pi : \{1, 2, \dots, n\} \to N$ be a permutation defined on the set of players N, which provides a sequence of players $(\pi(1), \pi(2), \dots, \pi(n))$. We denote the set of all the permutations by Π_N. We say that the player $\pi(i) \in N$ is the *pivot* of the permutation $\pi \in \Pi_N$, if $\{\pi(1), \pi(2), \dots, \pi(i-1)\}$ is a losing coalition and $\{\pi(1), \pi(2), \dots, \pi(i-1), \pi(i)\}$ is a winning coalition. For any permutation $\pi \in \Pi_N$, $\mathrm{piv}(\pi) \in N$ denotes the pivot of π. For each player $i \in N$, we define $\Pi_i = \{\pi \in \Pi_N \mid \mathrm{piv}(\pi) = i\}$. Obviously, $\{\Pi_1, \Pi_2, \dots, \Pi_n\}$ becomes a partition of Π_N. The S–S index of player i, denoted by φ_i, is defined by $|\Pi_i|/n!$. We know that $0 \leq \varphi_i \leq 1$ $(\forall i \in N)$ and $\sum_{i \in N} \varphi_i = 1$.

Assumption 1. *The set of players is arranged to satisfy $w_1 \geq w_2 \geq \cdots \geq w_n$.*

Clearly, this assumption implies that $\varphi_1 \geq \varphi_2 \geq \cdots \geq \varphi_n$.

3. Naive Algorithm and Its Analysis

In this section, we describe a naive Monte Carlo algorithm and analyze its theoretical performance. In the following, M denotes the number of samples generated in our algorithm.

Algorithm 1 Naive Algorithm.

Step 0: Set $m := 1$, $\varphi'_i := 0 \ \ (\forall i \in N)$.
Step 1: Choose $\pi \in \Pi_N$ uniformly at random.
Put (the random variable) $I^{(m)} := \mathrm{piv}(\pi)$. Update $\varphi'_{I^{(m)}} := \varphi'_{I^{(m)}} + 1$.
Step 2: If $m = M$, then output $\varphi'_i / M \ \ (\forall i \in N)$ and stop.
Else, update $m := m + 1$ and go to Step 1.

For each permutation $\pi \in \Pi_N$, we can find the pivot $\mathrm{piv}(\pi) \in N$ in $\mathrm{O}(n)$ time. Thus, the time complexity of Algorithm 1 is bounded by $\mathrm{O}(M(\tau(n) + n))$, where $\tau(n)$ denotes the computational effort required for the random generation of a permutation.

We denote the vector (of random variables) obtained by executing Algorithm 1 by $(\varphi_1^{\mathrm{A1}}, \varphi_2^{\mathrm{A1}}, \ldots, \varphi_n^{\mathrm{A1}})$. The following theorem is obvious.

Theorem 1. *For each player $i \in N$, $\mathrm{E}\left[\varphi_i^{\mathrm{A1}}\right] = \varphi_i$.*

The following theorem provides the number of samples required in Algorithm 1.

Theorem 2. *For any $\varepsilon > 0$ and $0 < \delta < 1$, we have the following.*
(1) Ref. [25] *If we set $M \geq \dfrac{\ln 2 + \ln(1/\delta)}{2\varepsilon^2}$, then each player $i \in N$ satisfies that*

$$\Pr\left[\left|\varphi_i^{\mathrm{A1}} - \varphi_i\right| < \varepsilon\right] \geq 1 - \delta.$$

(2) *If we set $M \geq \dfrac{\ln 2 + \ln(1/\delta) + \ln n}{2\varepsilon^2}$, then*

$$\Pr\left[\forall i \in N, \left|\varphi_i^{\mathrm{A1}} - \varphi_i\right| < \varepsilon\right] \geq 1 - \delta.$$

(3) *If we set $M \geq \dfrac{n \ln 2 + \ln(1/\delta)}{2\varepsilon^2}$, then*

$$\Pr\left[\frac{1}{2}\sum_{i \in N}\left|\varphi_i^{\mathrm{A1}} - \varphi_i\right| < \varepsilon\right] \geq 1 - \delta.$$

The distance measure $\frac{1}{2}\sum_{i \in N}\left|\varphi_i^{\mathrm{A1}} - \varphi_i\right|$ appearing in (3) is called the *total variation distance*.

Proof. Let us introduce random variables $X_i^{(m)}$ $(\forall m \in \{1, 2, \ldots, M\}, \forall i \in N)$ in Step 1 of Algorithm 1, defined by

$$X_i^{(m)} = \begin{cases} 1 & (\text{if } i = I^{(m)}), \\ 0 & (\text{otherwise}). \end{cases}$$

It is obvious that, for each player $i \in N$, $\{X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(M)}\}$ is a Bernoulli process satisfying $\varphi_i^{\mathrm{A1}} = \sum_{m=1}^{M} X_i^{(m)} / M$, $\mathrm{E}\left[\varphi_i^{\mathrm{A1}}\right] = \mathrm{E}\left[X_i^{(m)}\right] = \varphi_i$ $(\forall m \in \{1, 2, \ldots, M\})$. Hoeffding's inequality [40] implies that each player $i \in N$ satisfies

$$\Pr\left[\left|\varphi_i^{\mathrm{A1}} - \varphi_i\right| \geq \varepsilon\right] \leq 2\exp\left(-\frac{2M^2\varepsilon^2}{\sum_{m=1}^{M}(1-0)^2}\right) = 2\exp(-2M\varepsilon^2).$$

(1) If we set $M \geq \dfrac{\ln(2/\delta)}{2\varepsilon^2}$, then

$$\Pr\left[\left|\varphi_i^{\mathrm{A1}} - \varphi_i\right| < \varepsilon\right] \geq 1 - 2\exp\left(-2\frac{\ln(2/\delta)}{2\varepsilon^2}\varepsilon^2\right) = 1 - \delta.$$

(2) If we set $M \geq \dfrac{\ln(2n/\delta)}{2\varepsilon^2}$, then we have that

$$\begin{aligned}
\Pr\left[\forall i \in N, \left|\varphi_i^{\mathrm{A1}} - \varphi_i\right| < \varepsilon\right] &= 1 - \Pr\left[\exists i \in N, \left|\varphi_i^{\mathrm{A1}} - \varphi_i\right| \geq \varepsilon\right] \\
&\geq 1 - \sum_{i \in N} \Pr\left[\left|\varphi_i^{\mathrm{A1}} - \varphi_i\right| \geq \varepsilon\right] \geq 1 - \sum_{i=1}^{n} 2\exp(-2M\varepsilon^2) \\
&\geq 1 - \sum_{i=1}^{n} 2\exp\left(-2\frac{\ln(2n/\delta)}{2\varepsilon^2}\varepsilon^2\right) = 1 - \sum_{i=1}^{n}\frac{\delta}{n} = 1 - \delta.
\end{aligned}$$

(3) The vector of random variables

$$(M\varphi_1^{\mathrm{A1}}, M\varphi_2^{\mathrm{A1}}, \cdots, M\varphi_n^{\mathrm{A1}}) = \left(\sum_{m=1}^{M} X_1^{(m)}, \sum_{m=1}^{M} X_2^{(m)}, \cdots, \sum_{m=1}^{M} X_n^{(m)}\right)$$

is multinomially distributed with parameters M and $(\varphi_1, \varphi_2, \cdots, \varphi_n)$. Then, the Bretagnolle–Huber–Carol inequality [41] (Theorem A1 in Appendix A) implies that

$$\begin{aligned}
\Pr\left[\frac{1}{2}\sum_{i \in N}\left|\varphi_i^{\mathrm{A1}} - \varphi_i\right| \geq \varepsilon\right] &= \Pr\left[\sum_{i \in N}\left|M\varphi_i^{\mathrm{A1}} - M\varphi_i\right| \geq 2M\varepsilon\right] \leq 2^n \exp\left(-2M\varepsilon^2\right) \\
&\leq 2^n \exp\left(-2\left(\frac{\ln(2^n/\delta)}{2\varepsilon^2}\right)\varepsilon^2\right) = \delta,
\end{aligned}$$

and thus, we have the desired result. □

4. Our Algorithm

In this section, we propose a new algorithm based on the hierarchical structure of the partition $\{\Pi_1, \Pi_2, \ldots, \Pi_n\}$. First, we introduce a map $f_i : \Pi_i \to \Pi_{i-1}$ for each $i \in N \setminus \{1\}$. For any $\pi \in \Pi_i$, $f_i(\pi)$ denotes a permutation obtained by swapping the positions of players i and $i-1$ in the permutation $(\pi(1), \pi(2), \ldots, \pi(n))$. Because $w_{i-1} \geq w_i$ (Assumption 1), it is easy to show that the pivot of $f_i(\pi)$ becomes the player $i-1$. The definition of f_i directly implies that $\forall\{\pi, \pi'\} \subseteq \Pi_i$; if $\pi \neq \pi'$, then $f_i(\pi) \neq f_i(\pi')$. Thus, we have the following.

Lemma 1. *For any $i \in N \setminus \{1\}$, the map $f_i : \Pi_i \to \Pi_{i-1}$ is injective.*

Figure 1 shows injective maps f_2, f_3, f_4 induced by $G = [50; 40, 30, 20, 10]$.

$$
\begin{array}{ccccccc}
\Pi_1 & f_2 & \Pi_2 & f_3 & \Pi_3 & f_4 & \Pi_4 \\
(2,1,3,4) & \leftarrow & (1,2,3,4) & \leftarrow & (1,3,2,4) & \leftarrow & (1,4,2,3) \\
(2,1,4,3) & \leftarrow & (1,2,4,3) & \leftarrow & (1,3,4,2) & \leftarrow & (1,4,3,2) \\
(4,3,1,2) & \leftarrow & (4,3,2,1) & \leftarrow & (4,2,3,1) & & \\
(3,4,1,2) & \leftarrow & (3,4,2,1) & \leftarrow & (2,4,3,1) & & \\
(3,1,4,2) & \leftarrow & (3,2,4,1) & \leftarrow & (2,3,4,1) & & \\
(3,1,2,4) & \leftarrow & (3,2,1,4) & \leftarrow & (2,3,1,4) & & \\
(4,1,3,2) & & & & & & \\
(4,1,2,3) & & & & & & \\
(4,2,1,3) & & & & & & \\
(2,4,1,3) & & & & & &
\end{array}
$$

Figure 1. Injective maps f_2, f_3, f_4 induced by $G = [50; 40, 30, 20, 10]$. The circled number (player) denotes the pivot player.

When an ordered pair of permutations (π, π') satisfies that either (1) $\pi = \pi'$ or (2) $\pi \in \Pi_i$, $\pi' \in \Pi_j$, $i < j$, and $\pi = f_{i-1} \circ \cdots \circ f_{j-1} \circ f_j(\pi')$, we say that π' is an *ancestor* of π. Here, we note that π is always an ancestor of π itself. Lemma 1 implies that every permutation $\pi \in \Pi_N$ has a unique ancestor, called the *originator*, $\pi' \in \Pi_j$ satisfying either that $j = n$ or that its inverse image $f_{j+1}^{-1}(\pi') = \emptyset$. For each permutation $\pi \in \Pi_N$, $\mathrm{org}(\pi) \in N$ denotes the pivot of the originator of π; i.e., $\Pi_{\mathrm{org}(\pi)}$ includes the originator of π.

Now, we describe our algorithm.

Algorithm 2 Our Algorithm.

Step 0: Set $m := 1$, $\varphi'_i := 0 \ \ (\forall i \in N)$.
Step 1: Choose $\pi \in \Pi_N$ uniformly at random. Put the random variable $L^{(m)} := \mathrm{org}(\pi)$.
Update $\varphi'_i := \begin{cases} \varphi'_i + 1/L^{(m)} & (\text{if } i \leq L^{(m)}), \\ \varphi'_i & (\text{if } L^{(m)} < i). \end{cases}$
Step 2: If $m = M$, then output $\varphi'_i / M \ \ (\forall i \in N)$ and stop.
Else, update $m := m + 1$ and go to Step 1.

In the example shown in Figure 1, if we choose $\pi = (3, 2, 4, 1)$ at Step 1 of Algorithm 2, then $\mathrm{org}(\pi) = 3$, and Algorithm 2 updates

$$(\varphi'_1, \varphi'_2, \varphi'_3, \varphi'_4) := (\varphi'_1 + (1/3), \varphi'_2 + (1/3), \varphi'_3 + (1/3), \varphi'_4).$$

For each permutation $\pi \in \Pi_N$, we can find the originator $\mathrm{org}(\pi) \in N$ in $\mathrm{O}(n)$ time. Thus, the time complexity of Algorithm 2 is also bounded by $\mathrm{O}(M(\tau(n) + n))$, where $\tau(n)$ denotes the computational effort required for the random generation of a permutation.

We denote the vector (of random variables) obtained by executing Algorithm 2 by $(\varphi_1^{\mathrm{A2}}, \varphi_2^{\mathrm{A2}}, \ldots, \varphi_n^{\mathrm{A2}})$. We have the following properties.

Theorem 3. (1) *For each player $i \in N$, $\mathrm{E}\left[\varphi_i^{\mathrm{A2}}\right] = \varphi_i$.*
(2) *For each pair of players $\{i, j\} \subseteq N$, if $\varphi_i > \varphi_j$, then $\varphi_i^{\mathrm{A2}} \geq \varphi_j^{\mathrm{A2}}$.*
(3) *For each pair of players $\{i, j\} \subseteq N$, if $\varphi_i = \varphi_j$, then $\varphi_i^{\mathrm{A2}} = \varphi_j^{\mathrm{A2}}$.*

Proof. (1) For each $i \in N$, we define that $\xi_i = |\{\pi \in \Pi_N \mid \mathrm{org}(\pi) = i\}|$. It is obvious that $|\Pi_i| = \xi_i + \xi_{i+1} + \cdots + \xi_n \ (\forall i \in N)$. If we choose $\pi \in \Pi_N$ uniformly at random, then

$\Pr[\mathrm{org}(\pi) = i] = (i \cdot \xi_i)/n!$ holds. From the above, it is easy to see that, for each $i \in N$, a random variable $\Delta\varphi'_i(\pi)$ defined by

$$\Delta\varphi'_i(\pi) := \begin{cases} 1/\mathrm{org}(\pi) & (\text{if } i \le \mathrm{org}(\pi)), \\ 0 & (\text{if } \mathrm{org}(\pi) < i), \end{cases}$$

satisfies that

$$\begin{aligned} & \mathrm{E}[\Delta\varphi'_i(\pi)] \\ = \; & \Pr[\mathrm{org}(\pi) = i]\left(\frac{1}{i}\right) + \Pr[\mathrm{org}(\pi) = i+1]\left(\frac{1}{i+1}\right) + \cdots + \Pr[\mathrm{org}(\pi) = n]\left(\frac{1}{n}\right) \\ = \; & \left(\frac{i \cdot \xi_i}{n!}\right)\left(\frac{1}{i}\right) + \left(\frac{(i+1)\cdot\xi_{i+1}}{n!}\right)\left(\frac{1}{i+1}\right) + \cdots + \left(\frac{n \cdot \xi_n}{n!}\right)\left(\frac{1}{n}\right) \\ = \; & \frac{\xi_i + \xi_{i+1} + \cdots + \xi_n}{n!} = \frac{|\Pi_i|}{n!} = \varphi_i. \end{aligned}$$

When $(\pi^{(1)}, \pi^{(2)}, \ldots, \pi^{(M)})$ denotes a sequence of permutations generated in Algorithm 2, we have that

$$\begin{aligned} \mathrm{E}\left[\varphi_i^{\mathrm{A2}}\right] &= \mathrm{E}\left[\frac{\Delta\varphi'_i(\pi^{(1)}) + \Delta\varphi'_i(\pi^{(2)}) + \cdots + \Delta\varphi'_i(\pi^{(M)})}{M}\right] = \sum_{m=1}^{M} \frac{\mathrm{E}[\Delta\varphi'_i(\pi^{(m)})]}{M} \\ &= \sum_{m=1}^{M} \frac{\varphi_i}{M} = \varphi_i. \end{aligned}$$

(2) Assumption 1 implies that, if $\varphi_i > \varphi_j$, then $w_i > w_j$, and thus, $i < j$. The update formula of φ'_i in Algorithm 2 directly implies that inequalities $\varphi'_1 \ge \varphi'_2 \ge \cdots \ge \varphi'_n$ hold throughout the iterations of Algorithm 2, which leads to inequalities $\varphi_1^{\mathrm{A2}} \ge \varphi_2^{\mathrm{A2}} \ge \cdots \ge \varphi_n^{\mathrm{A2}}$. From the above, we obtain that $\varphi_i^{\mathrm{A2}} \ge \varphi_j^{\mathrm{A2}}$.
(3) In the following, we assume that $\varphi_i = \varphi_j$ and $i < j$. Assumption 1 implies that $\varphi_i = \varphi_{i+1} = \cdots = \varphi_j$, and thus, $|\Pi_i| = |\Pi_{i+1}| = \cdots = |\Pi_j|$. From Lemma 1, $f_{i+1}, \ldots, f_j$ are bijections; thus, $\forall \pi \in \Pi, \mathrm{org}(\pi) \notin \{i, i+1, \ldots, j-1\}$. Then, the update formula of φ'_i in Algorithm 2 implies that equalities $\varphi'_i = \varphi'_{i+1} = \cdots = \varphi'_j$ hold throughout the iterations of Algorithm 2, which leads to the desired result: $\varphi_i^{\mathrm{A2}} = \varphi_{i+1}^{\mathrm{A2}} = \cdots = \varphi_j^{\mathrm{A2}}$. □

The following theorem provides the number of samples required in Algorithm 2.

Theorem 4. *For any $\varepsilon > 0$ and $0 < \delta < 1$, we have the following.*
(1) *For each player $i \in N = \{1, 2, \ldots, n\}$, if we set $M \ge \dfrac{\ln 2 + \ln(1/\delta)}{2\varepsilon^2 i^2}$, then*

$$\Pr\left[\left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| < \varepsilon\right] \ge 1 - \delta.$$

(2) *If we set $M \ge \dfrac{\ln 2 + \ln(1/\delta)}{2\varepsilon^2}$, then*

$$\begin{aligned} \Pr\left[\forall i \in N, \left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| < \varepsilon\right] &\ge 1 - 2\sum_{i=1}^{n}\left(\frac{\delta}{2}\right)^{i^2} \\ &= 1 - 2\left(\left(\frac{\delta}{2}\right) + \left(\frac{\delta}{2}\right)^4 + \left(\frac{\delta}{2}\right)^9 + \cdots + \left(\frac{\delta}{2}\right)^{n^2}\right). \end{aligned}$$

(3) *If we set* $M \geq \dfrac{|N^*| \ln 2 + \ln(1/\delta)}{2\varepsilon^2}$, *then*

$$\Pr\left[\frac{1}{2}\sum_{i \in N}\left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| < \varepsilon\right] \geq 1 - \delta,$$

where $N^* = \{i \in N \setminus \{n\} \mid \varphi_i > \varphi_{i+1}\} \cup \{n\}$, *i.e.,* $|N^*|$ *is equal to the size of the maximal player subset, the S–S indices of which are mutually different.*

Proof. Let us introduce random variables $X_i^{(m)}$ ($\forall m \in \{1, 2, \ldots, M\}, \forall i \in N$) in Step 2 of Algorithm 2, defined by

$$X_i^{(m)} = \begin{cases} 1/L^{(m)} & (\text{if } 1 \leq i \leq L^{(m)}), \\ 0 & (\text{if } L^{(m)} < i). \end{cases}$$

It is obvious that, for each player $i \in N$, $\{X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(M)}\}$ is a collection of independent and identically distributed random variables satisfying

$$\varphi_i^{\mathrm{A2}} = \sum_{m=1}^{M} X_i^{(m)}/M, \quad \mathrm{E}\left[\varphi_i^{\mathrm{A2}}\right] = \mathrm{E}\left[X_i^{(m)}\right] = \varphi_i, \text{ and } (1/i) \geq X_i^{(m)} \geq (1/n)$$

for all $m \in \{1, 2, \ldots, M\}$. Hoeffding's inequality [40] implies that each player $i \in N$ satisfies

$$\Pr\left[\left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| \geq \varepsilon\right] \leq 2\exp\left(-\frac{2M^2\varepsilon^2}{\sum_{m=1}^{M}(1/i - 0)^2}\right) = 2\exp(-2M\varepsilon^2 i^2).$$

(1) If we set $M \geq \dfrac{\ln(2/\delta)}{2\varepsilon^2 i^2}$, then

$$\Pr\left[\left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| < \varepsilon\right] \geq 1 - 2\exp\left(-2\frac{\ln(2/\delta)}{2\varepsilon^2 i^2}\varepsilon^2 i^2\right) = 1 - \delta.$$

(2) If we set $M \geq \dfrac{\ln(2/\delta)}{2\varepsilon^2}$, then we have that

$$\begin{aligned} &\Pr\left[\forall i \in N, \left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| < \varepsilon\right] = 1 - \Pr\left[\exists i \in N, \left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| \geq \varepsilon\right] \\ \geq\ & 1 - \sum_{i \in N}\Pr\left[\left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| \geq \varepsilon\right] \geq 1 - \sum_{i=1}^{n} 2\exp(-2M\varepsilon^2 i^2) \\ \geq\ & 1 - 2\sum_{i=1}^{n}\exp\left(-2\frac{\ln(2/\delta)}{2\varepsilon^2}\varepsilon^2 i^2\right) = 1 - 2\sum_{i=1}^{n}\left(\frac{\delta}{2}\right)^{i^2}. \end{aligned}$$

(3) We introduce random variables $Y_\ell^{(m)}$ ($\forall m \in \{1, 2, \ldots, M\}, \forall \ell \in N$) in Step 2 of Algorithm 2, defined by

$$Y_\ell^{(m)} = \begin{cases} 1 & (\text{if } \ell = L^{(m)}), \\ 0 & (\text{otherwise}). \end{cases}$$

Because $\sum_{\ell=1}^{n} Y_\ell^{(m)} = 1$ ($\forall m$), the above definition directly implies that

$$X_i^{(m)} = \frac{1}{i}Y_i^{(m)} + \frac{1}{i+1}Y_{i+1}^{(m)} + \cdots + \frac{1}{n}Y_n^{(m)}.$$

For each player $i \in N$ and $i \leq \forall \ell \leq n$, we define $\Pi_{i\ell} = \{\pi \in \Pi_i \mid \mathrm{org}(\pi) = \ell\}$. It is easy to show that $|\Pi_{1\ell}| = |\Pi_{2\ell}| = \cdots = |\Pi_{\ell\ell}|$ for each $\ell \in \{1, 2, \ldots, n\}$. The above definitions imply that

$$\begin{aligned}
\frac{1}{2}\sum_{i \in N}\left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| &= \frac{1}{2M}\sum_{i \in N}\left|M\varphi_i^{\mathrm{A2}} - M\varphi_i\right| = \frac{1}{2M}\sum_{i \in N}\left|\sum_{m=1}^{M} X_i^{(m)} - M\frac{|\Pi_i|}{n!}\right| \\
&= \frac{1}{2M}\sum_{i \in N}\left|\sum_{m=1}^{M}\sum_{\ell=i}^{n}\frac{1}{\ell}Y_\ell^{(m)} - \frac{M}{n!}\sum_{\ell=i}^{n}|\Pi_{i\ell}|\right| \\
&= \frac{1}{2M}\sum_{i \in N}\left|\sum_{\ell=i}^{n}\left(\sum_{m=1}^{M}\frac{1}{\ell}Y_\ell^{(m)} - \frac{M}{n!}|\Pi_{i\ell}|\right)\right| \\
&\leq \frac{1}{2M}\sum_{i=1}^{n}\sum_{\ell=i}^{n}\left|\sum_{m=1}^{M}\frac{1}{\ell}Y_\ell^{(m)} - \frac{M}{n!}|\Pi_{i\ell}|\right| = \frac{1}{2M}\sum_{\ell=1}^{n}\sum_{i=1}^{\ell}\left|\sum_{m=1}^{M}\frac{1}{\ell}Y_\ell^{(m)} - \frac{M}{n!}|\Pi_{i\ell}|\right| \\
&= \frac{1}{2M}\sum_{\ell=1}^{n}\ell\left|\sum_{m=1}^{M}\frac{1}{\ell}Y_\ell^{(m)} - \frac{M}{n!}|\Pi_{1\ell}|\right| \quad (\text{since } |\Pi_{1\ell}| = |\Pi_{2\ell}| = \cdots = |\Pi_{\ell\ell}|) \\
&= \frac{1}{2M}\sum_{\ell=1}^{n}\left|\sum_{m=1}^{M}Y_\ell^{(m)} - \frac{M\ell}{n!}|\Pi_{1\ell}|\right|.
\end{aligned}$$

For each player $\ell \notin N^*$, we have the equalities $|\Pi_\ell| = n!\varphi_\ell = n!\varphi_{\ell+1} = |\Pi_{\ell+1}|$, which yields that $f_{\ell+1} : \Pi_{\ell+1} \to \Pi_\ell$ is a bijection; thus, Π_ℓ does not include any originator. From the above, it is obvious that, if $\ell \notin N^*$, then $\Pi_{1\ell} = \Pi_{2\ell} = \cdots = \Pi_{\ell\ell} = \emptyset$. For each $\ell \in \{1, 2, \ldots, n\}$, $\{Y_\ell^{(1)}, Y_\ell^{(2)}, \ldots, Y_\ell^{(M)}\}$ is a Bernoulli process satisfying $\mathrm{E}[Y_\ell^{(m)}] = \frac{1}{n!}\sum_{i=1}^{\ell}|\Pi_{i\ell}| = \frac{\ell}{n!}|\Pi_{1\ell}|$ $(\forall m)$. Thus, $\ell \notin N^*$ implies that $Y_\ell^{(m)} = 0$ for any $m \in \{1, 2, \ldots, M\}$. To summarize the above, we have shown that

$$\text{if } \ell \notin N^* \text{ then } \sum_{m=1}^{M} Y_\ell^{(m)} - \frac{M\ell}{n!}|\Pi_{1\ell}| = \sum_{m=1}^{M} 0 - \frac{M\ell}{n!}0 = 0.$$

Now, we have an upper bound of the total variation distance

$$\begin{aligned}
\frac{1}{2}\sum_{i \in N}\left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| &\leq \frac{1}{2M}\sum_{\ell=1}^{n}\left|\sum_{m=1}^{M}Y_\ell^{(m)} - \frac{M\ell}{n!}|\Pi_{1\ell}|\right| \\
&= \frac{1}{2M}\sum_{\ell \in N^*}\left|\sum_{m=1}^{M}Y_\ell^{(m)} - \sum_{m=1}^{M}\mathrm{E}[Y_\ell^{(m)}]\right|.
\end{aligned}$$

The vector of random variables $\left(\sum_{m=1}^{M} Y_\ell^{(m)}\right)_{\ell \in N^*}$ is multinomially distributed and satisfies that the total sum is equal to M. Then, the Bretagnolle–Huber–Carol inequality [41] (Theorem A1 in Appendix A) implies that

$$\begin{aligned}
\Pr\left[\frac{1}{2}\sum_{i \in N}\left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| \geq \varepsilon\right] &\leq \Pr\left[\frac{1}{2M}\sum_{\ell \in N^*}\left|\sum_{m=1}^{M}Y_\ell^{(m)} - \sum_{m=1}^{M}\mathrm{E}[Y_\ell^{(m)}]\right| \geq \varepsilon\right] \\
&= \Pr\left[\sum_{\ell \in N^*}\left|\sum_{m=1}^{M}Y_\ell^{(m)} - \sum_{m=1}^{M}\mathrm{E}[Y_\ell^{(m)}]\right| \geq 2M\varepsilon\right] \leq 2^{|N^*|}\exp\left(-2M\varepsilon^2\right) \\
&\leq 2^{|N^*|}\exp\left(-2\frac{\ln\left(2^{|N^*|}/\delta\right)}{2\varepsilon^2}\varepsilon^2\right) = \delta
\end{aligned}$$

and thus, we have the desired result. □

The following corollary provides an approximate version of Theorem 4 (2). Surprisingly, it says that the required number of samples is irrelevant to n (number of players).

Corollary 1. *For any $\varepsilon > 0$ and $0 < \delta' < 1$, we have the following.*
If we set $M \geq \dfrac{\ln 2 + \ln(1/\delta') + \ln 1.129}{2\varepsilon^2}$, *then*

$$\Pr\left[\forall i \in N, \left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| < \varepsilon\right] \geq 1 - \delta'.$$

Proof. If we put $\delta = \delta'/1.129$, then Theorem 2 (2) implies that

$$\begin{aligned}
&\Pr\left[\forall i \in N, \left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| < \varepsilon\right] \\
&\geq\ 1 - 2\left(\left(\frac{\delta}{2}\right) + \left(\frac{\delta}{2}\right)^4 + \left(\frac{\delta}{2}\right)^9 + \cdots + \left(\frac{\delta}{2}\right)^{n^2}\right) \\
&\geq\ 1 - \delta\left(1 + \left(\frac{1}{2}\right)^3 + \left(\frac{1}{2}\right)^8 + \left(\frac{1}{2}\right)^{15} + \left(\frac{1}{2}\right)^{24} + \cdots + \left(\frac{1}{2}\right)^{n^2-1}\right) \\
&\geq\ 1 - \delta\left(1 + \left(\frac{1}{2}\right)^3 + \left(\frac{1}{2}\right)^8\left(1 + \left(\frac{1}{2}\right)^7 + \left(\frac{1}{2}\right)^{14} + \left(\frac{1}{2}\right)^{21} + \cdots\right)\right) \\
&=\ 1 - \delta\left(1 + \left(\frac{1}{2}\right)^3 + \left(\frac{1}{2}\right)^8\left(\frac{1}{1-(1/2)^7}\right)\right) \geq 1 - 1.129\delta = 1 - \delta'. \quad \square
\end{aligned}$$

Here, we note that $\ln 2 \simeq 0.69314$ and $\ln 1.129 \simeq 0.12133$.

In a practical setting, it is difficult to estimate the size of N^* defined in Theorem 4 (3) since the problem of verifying the asymmetricity of a given pair of players is NP-complete [19]. The following corollary is useful in some practical situations.

Corollary 2. *For any $\varepsilon > 0$ and $0 < \delta < 1$, we have the following.*
If we set $M \geq \dfrac{n'' \ln 2 + \ln(1/\delta)}{2\varepsilon^2}$, *then*

$$\Pr\left[\frac{1}{2}\sum_{i \in N}\left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| < \varepsilon\right] \geq 1 - \delta,$$

where $n'' = |\{i \in N \setminus \{n\} \mid w_i > w_{i+1}\} \cup \{n\}|$, i.e., n'' is equal to the size of a maximal player subset with mutually different weights.

Proof. Since $\varphi_i > \varphi_{i+1}$ implies $w_i > w_{i+1}$, it is obvious that $|N^*| \leq n''$, and we have the desired result. $\square$

A game of the power of the countries in the EU Council is defined by (255; 29, 29, 29, 29, 27, 27, 14, 13, 12, 12, 12, 12, 12, 10, 10, 10, 7, 7, 7, 7, 7, 4, 4, 4, 4, 4, 3) [42,43]. In this case, $n = 27$ and $n'' = 9$. A weighted majority game defined by [44] (Section 12.4) for a voting process in the United States has a vector of weights

$$\begin{aligned}
&(270; 45, 41, 27, 26, 26, 25, 21, 17, 17, 14, 13, 13, 12, 12, 12, 11, \underbrace{10, \ldots, 10}_{4\text{ times}}, \underbrace{9, \ldots, 9}_{4\text{ times}}, \\
&8, 8, \underbrace{7, \ldots, 7}_{4\text{ times}}, \underbrace{6, \ldots, 6}_{4\text{ times}}, 5, \underbrace{4, \ldots, 4}_{9\text{ times}}, \underbrace{3, \ldots, 3}_{7\text{ times}}), \text{ where } n = 51 \text{ and } n'' = 19.
\end{aligned}$$

5. Computational Experiments

This section reports the results of our preliminary numerical experiments. All the experiments were conducted on a Windows machine, i7-7700 CPU@3.6GHz Memory (RAM) 16 GB. Algorithms 1 and 2 were implemented using Python 3.6.5.

We tested the EU Council instance and the United States instance described in the previous section. In each instance, we set M in Algorithms 1 and 2 (the number of generated permutations) to $M \in \{1 \times 10^5, 2 \times 10^5, \ldots, 24 \times 10^5\}$. For each value of M, we executed Algorithms 1 and 2 100 times. Figures 2 and 3 show the results of some players. For each value of M, we calculated the mean number of $|\varphi_i - \varphi_i^{\mathrm{A}}|$, denoted by $\widehat{\varepsilon}_i$, in an average of 100 trials. The horizontal axes of Figures 2 and 3 show the value $1/\widehat{\varepsilon}_i{}^2$. Under the assumption that $M = \alpha/\widehat{\varepsilon}_i{}^2$, we estimated α by the least squares method. Table 2 shows the results and ratios of α of the two algorithms.

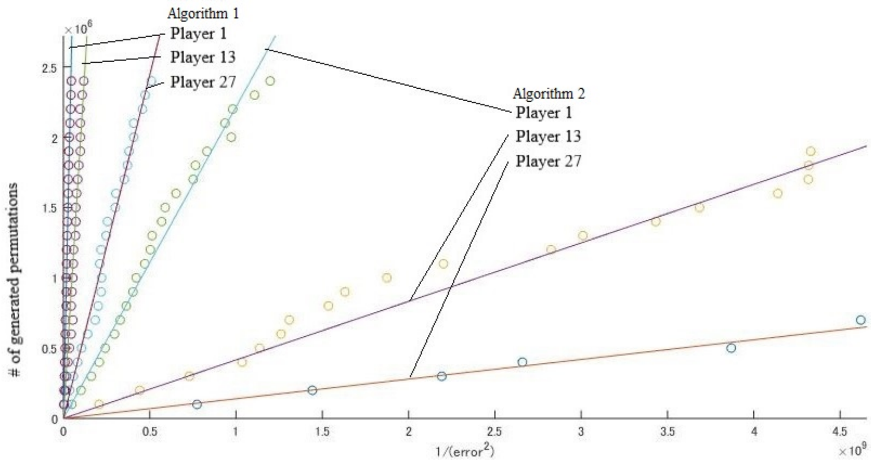

Figure 2. EU Council.

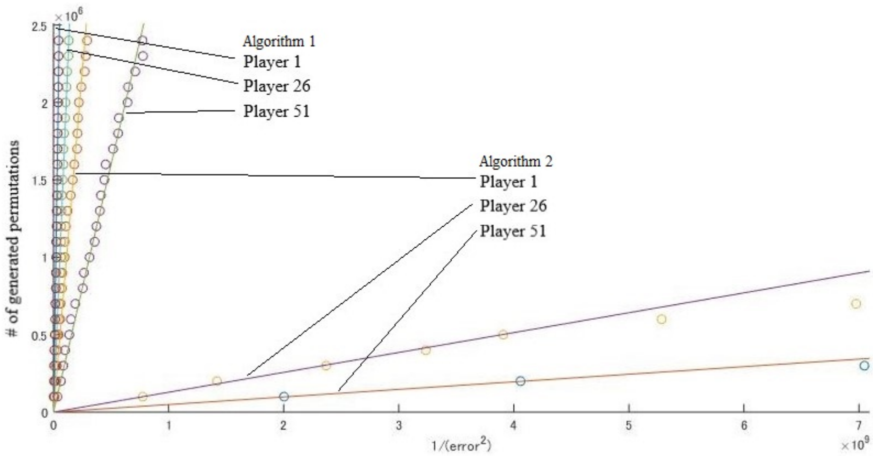

Figure 3. United States.

Table 2. Comparison of Algorithms 1 and 2.

EU Council	α Algorithm 1	α Algorithm 2	ratio
Player 1	0.0557	0.0022	25.318
Player 13	0.0199	4.1615×10^{-4}	47.819
Player 27	0.0049	1.3987×10^{-4}	35.033
United States	**α Algorithm 1**	**α Algorithm 2**	**ratio**
Player 1	0.0489	0.0181	2.7017
Player 26	0.0088	1.2837×10^{-4}	68.552
Player 51	0.0032	4.8911×10^{-5}	65.424

For each (generated) permutation, the computational effort of both Algorithms 1 and 2 are bounded by $O(n)$. Here, we discuss the constant factors of $O(n)$ computations. We tested the cases where weights w_i are generated uniformly at random from the intervals $(1, 10)$ or $(1, 20)$, and the quota is equal to $(1/2)\sum_{i \in N} w_i$. For each $n \in \{10, 20, \ldots, 100\}$, we executed Algorithms 1 and 2 by setting M = 10,000. Under the assumption that computational time is equal to $an + b$, we estimated a and b by the least squares method. Figure 4 shows that for each permutation, the computational effort of Algorithm 2 increases about five-fold compared to Algorithm 1.

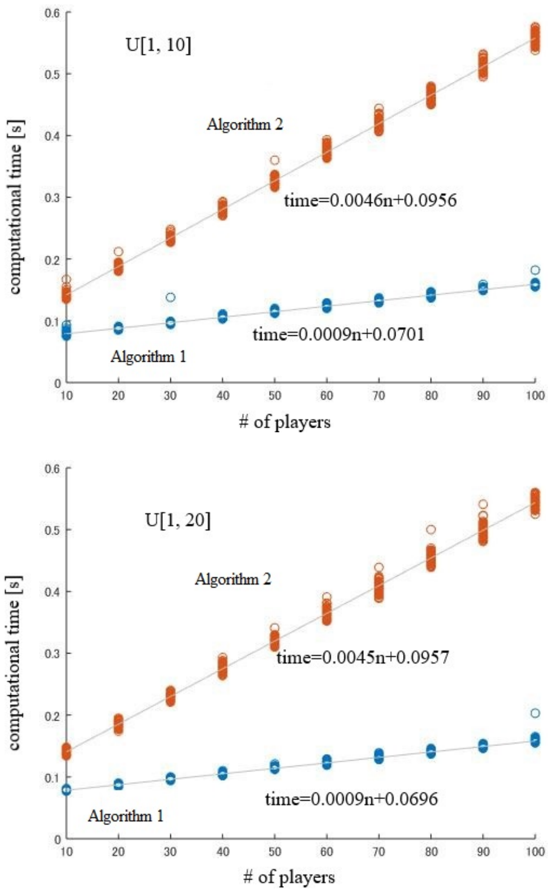

Figure 4. Computational time.

6. Conclusions

In this paper, we analyzed a naive Monte Carlo algorithm (Algorithm 1) for calculating the S–S index denoted by $(\varphi_1, \varphi_2, \ldots, \varphi_n)$ in weighted majority games. By employing the Bretagnolle–Huber–Carol inequality [41] (Theorem A1 in Appendix A), we estimated the required number of samples that gives an upper bound of the total variation distance.

We also proposed an efficient Monte Carlo algorithm (Algorithm 2). The time complexity of each iteration of our algorithm is equal to that of the naive algorithm (Algorithm 1). Our algorithm has the property that the obtained estimator $(\varphi_1^{\mathrm{A2}}, \varphi_2^{\mathrm{A2}}, \ldots, \varphi_n^{\mathrm{A2}})$ satisfies

$$\text{both } [\text{ if } \varphi_i < \varphi_j, \text{ then } \varphi_i^{\mathrm{A2}} \leq \varphi_j^{\mathrm{A2}}] \text{ and } [\text{ if } \varphi_i = \varphi_j, \text{ then } \varphi_i^{\mathrm{A2}} = \varphi_j^{\mathrm{A2}}].$$

We proved that, even if we consider the property

$$\Pr\left[\forall i \in N, \left|\varphi_i^{\mathrm{A2}} - \varphi_i\right| < \varepsilon\right] \geq 1 - \delta,$$

the required number of samples is irrelevant to n (the number of players).

Author Contributions: Y.U.: validation, original draft preparation; M.T.: visualization, coding simulations; T.M.: manuscript writing, project supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by JSPS KAKENHI, Grant Numbers JP26285045, JP26242027, and JP20K04973.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Appendix A. (Bretagnolle–Huber–Carol Inequality)

Theorem A1 ([41])**.** *If the random vector $(Z_1, Z_2, Z_3, \ldots, Z_n)$ is multinomially distributed with parameters $(p_1, p_2, \ldots, p_n)$ and satisfies $Z_1 + Z_2 + \cdots + Z_n = M$, then*

$$\Pr\left[\sum_{i=1}^{n} |Z_i - Mp_i| \geq 2M\varepsilon\right] \leq 2^n \exp(-2M\varepsilon^2).$$

Proof. It is easy to see that

$$\begin{aligned}
\Pr\left[\sum_{i=1}^{n} |Z_i - Mp_i| \geq 2M\varepsilon\right] &= \Pr\left[2 \max_{S \subseteq \{1,2,\ldots,n\}} \sum_{i \in S} (Z_i - Mp_i) \geq 2M\varepsilon\right] \\
&= \Pr\left[\exists S \subseteq \{1,2,\ldots,n\}, \ \sum_{i \in S} (Z_i - Mp_i) \geq M\varepsilon\right] \leq \sum_{S \subseteq \{1,2,\ldots,n\}} \Pr\left[\sum_{i \in S} (Z_i - Mp_i) \geq M\varepsilon\right] \\
&= \sum_{S \subseteq \{1,2,\ldots,n\}} \Pr\left[\sum_{i \in S} Z_i - M \sum_{i \in S} p_i \geq M\varepsilon\right]
\end{aligned}$$

For any subset $S \subseteq \{1,2,\ldots,n\}$, there exists a Bernoulli process $(X_S^{(1)}, X_S^{(2)}, \ldots, X_S^{(M)})$ satisfying $\sum_{i\in S} Z_i = \sum_{m=1}^{M} X_S^{(m)}$ and $\mathrm{E}[X_S^{(m)}] = \sum_{i\in S} p_i$ $(\forall m \in \{1,2,\ldots M\})$. Hoeffding's inequality [40] implies that

$$\begin{aligned}
\sum_{S\subseteq\{1,2,\ldots,n\}} \Pr\left[\sum_{i\in S} Z_i - M\sum_{i\in S} p_i \geq M\varepsilon\right] &= \sum_{S\subseteq\{1,2,\ldots,n\}} \Pr\left[\sum_{m=1}^{M} X_S^{(m)} - \mathrm{E}\left[\sum_{m=1}^{M} X_S^{(m)}\right] \geq M\varepsilon\right] \\
&= \sum_{S\subseteq\{1,2,\ldots,n\}} \Pr\left[\frac{1}{M}\sum_{m=1}^{M} X_S^{(m)} - \frac{1}{M}\mathrm{E}\left[\sum_{m=1}^{M} X_S^{(m)}\right] \geq \varepsilon\right] \leq \sum_{S\subseteq\{1,2,\ldots,n\}} \exp(-2M\varepsilon^2) \\
&= 2^n \exp(-2M\varepsilon^2). \quad \square
\end{aligned}$$

References

1. Von Neumann, J.; Morgenstern, O. *Theory of Games and Economic Behavior*; Princeton University Press: Princeton, NJ, USA, 1944.
2. Shapley, L.S.; Shubik, M. A method for evaluating the distribution of power in a committee system. *Am. Political Sci. Rev.* **1954**, *48*, 787–792.
3. Shapley, L.S. A value for n-person games. In *Contributions to the Theory of Games II*; Kuhn, H.W., Tucker, A.W., Eds.; Princeton University Press: Princeton, NJ, USA, 1953; pp. 307–317.
4. Laruelle, A.; Widgrén, M. Is the allocation of voting power among EU states fair? *Public Choice* **1998**, *94*, 317–339.
5. Leech, D. Designing the voting system for the Council of the European Union. *Public Choice* **2002**, *113*, 437–464.
6. Życzkowski, K.; Słomczyński, W. Square Root Voting System, Optimal Threshold and π. In *Power, Voting, and Voting Power: 30 Years After*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 573–592.
7. Owen, G. Evaluation of a presidential election game. *Am. Political Sci. Rev.* **1975**, *69*, 947–953.
8. Miller, N.R. A priori voting power and the US Electoral College. In *Power, Voting, and Voting Power: 30 Years After*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 411–442.
9. Leech, D. Voting power in the governance of the International Monetary Fund. *Ann. Oper. Res.* **2002**, *109*, 375–397.
10. Leech, D.; Leech, R. A new analysis of a priori voting power in the IMF: Recent quota reforms give little cause for celebration. In *Power, voting, and voting power: 30 years after*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 389–410.
11. Young, H.P. Cost allocation. *Handb. Game Theory Econ. Appl.* **1994**, *2*, 1193–1235.
12. Lucchetti, R.; Radrizzani, P. Microarray data analysis via weighted indices and weighted majority games. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 179–190.
13. Holler, M.J.; Rupp, F. Power in networks: A PGI analysis of Krackhardt's Kite Network. In *Transactions on Computational Collective Intelligence XXXIV*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 21–34.
14. Holler, M.J.; Rupp, F. Power in Networks: The Medici. *Homo Oeconomicus* **2021**, *38*, 59–75.
15. Bachrach, Y.; Rosenschein, J.S.; Porat, E. Power and stability in connectivity games. In Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, Estoril, Portugal, 12–16 May 2008; 2008; Volume 2, pp. 999–1006.
16. Garey, M.R.; Johnson, D.S. *Computers and Intractability*; Freeman: San Francisco, CA, USA, 1979; Volume 174.
17. Deng, X.; Papadimitriou, C.H. On the complexity of cooperative solution concepts. *Math. Oper. Res.* **1994**, *19*, 257–266.
18. Prasad, K.; Kelly, J.S. NP-completeness of some problems concerning voting games. *Int. J. Game Theory* **1990**, *19*, 1–9.
19. Matsui, Y.; Matsui, T. NP-completeness for calculating power indices of weighted majority games. *Theor. Comput. Sci.* **2001**, *263*, 305–310.
20. Elkind, E.; Goldberg, L.A.; Goldberg, P.W.; Wooldridge, M. Computational complexity of weighted threshold games. In *AAAI*; Association for the Advancement of Artificial Intelligence: Menlo Park, CA, USA, 2007; pp. 718–723.
21. Mann, I.; Shapley, L.S. *Values of Large Games, IV: Evaluating the Electoral College by Montecarlo Techniques*; Technical Report RM-2651; The RAND Corporation: Santa Monica, CA, USA, 1960.
22. Matsui, T.; Matsui, Y. A survey of algorithms for calculating power indices of weighted majority games. *J. Oper. Res. Soc. Jpn.* **2000**, *43*, 71–86.
23. Fatima, S.S.; Wooldridge, M.; Jennings, N.R. A linear approximation method for the Shapley value. *Artif. Intell.* **2008**, *172*, 1673–1699.
24. Castro, J.; Gómez, D.; Tejada, J. Polynomial calculation of the Shapley value based on sampling. *Comput. Oper. Res.* **2009**, *36*, 1726–1730.
25. Bachrach, Y.; Markakis, E.; Resnick, E.; Procaccia, A.D.; Rosenschein, J.S.; Saberi, A. Approximating power indices: Theoretical and empirical analysis. *Auton. Agents-Multi-Agent Syst.* **2010**, *20*, 105–122.
26. Castro, J.; Gómez, D.; Molina, E.; Tejada, J. Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation. *Comput. Oper. Res.* **2017**, *82*, 180–188.
27. Owen, G. Multilinear extensions of games. *Manag. Sci.* **1972**, *18*, 64–79.
28. Leech, D. Computing power indices for large voting games. *Manag. Sci.* **2003**, *49*, 831–837.

29. Brams, S.J.; Affuso, P.J. Power and size: A new paradox. *Theory Decis.* **1976**, *7*, 29–56.
30. Lucas, W.F. Measuring power in weighted voting systems. In *Political and Related Models*; Springer: Berlin/Heidelberg, Germany, 1983; pp. 183–238.
31. Mann, I.; Shapley, L.S. *Values of Large Games, VI: Evaluating the Electoral College Exactly*; Technical Report RM-3158-PR; The RAND Corporation: Santa Monica, CA, USA, 1962.
32. Uno, T. Efficient computation of power indices for weighted majority games. In *International Symposium on Algorithms and Computation*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 679–689.
33. Bilbao, J.M.; Fernandez, J.R.; Jiménez-Losada, A.; Lopez, J.J. Generating functions for computing power indices efficiently. *Top* **2000**, *8*, 191–213.
34. Bolus, S. Power indices of simple games and vector-weighted majority games by means of binary decision diagrams. *Eur. J. Oper. Res.* **2011**, *210*, 258–272.
35. Rushdi, A.M.A.; Ba-Rukab, O.M. Map calculation of the Shapley-Shubik voting powers: An example of the European Economic Community. *Int. J. Math. Eng. Manag. Sci.* **2017**, *2*, 17–29.
36. Berghammer, R.; Bolus, S.; Rusinowska, A.; De Swart, H. A relation-algebraic approach to simple games. *Eur. J. Oper. Res.* **2011**, *210*, 68–80.
37. Klinz, B.; Woeginger, G.J. Faster algorithms for computing power indices in weighted voting games. *Math. Soc. Sci.* **2005**, *49*, 111–116.
38. Alonso-Meijide, J.M.; Casas-Méndez, B.; Fiestras-Janeiro, M.G. Computing Banzhaf–Coleman and Shapley–Shubik power indices with incompatible players. *Appl. Math. Comput.* **2015**, *252*, 377–387.
39. Courtin, S.; Nganmeni, Z.; Tchantcho, B. The Shapley–Shubik power index for dichotomous multi-type games. *Theory Decis.* **2016**, *81*, 413–426.
40. Hoeffding, W. Probability inequalities for sums of bounded random variables. *J. Am. Stat. Assoc.* **1963**, *58*, 13–30.
41. van der Vaart, A.W.; Wellner, J.A. *Weak convergence and empirical processes: With applications to statistics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1996.
42. Felsenthal, D.S.; Machover, M. The Treaty of Nice and qualified majority voting. *Soc. Choice Welf.* **2001**, *18*, 431–464.
43. Bilbao, J.M.; Fernandez, J.R.; Jiménez, N.; Lopez, J.J. Voting power in the European Union enlargement. *Eur. J. Oper. Res.* **2002**, *143*, 181–196.
44. Owen, G. *Game Theory*; Academic Press: Cambridge, MA, USA, 1995.