Journal of the Operations Research
Society of Japan
Vol. 43, No. 1, March 2000

A SURVEY OF ALGORITHMS FOR CALCULATING POWER INDICES OF WEIGHTED MAJORITY GAMES

Tomomi Matsui
University of Tokyo

Yasuko Matsui
Tokai University

(Received December 21, 1998; Revised September 2, 1999)

Abstract For measuring an individual's voting power of a voting game, some power indices are proposed. In this paper, we discuss the problems for calculating the Shapley-Shubik index, the Banzhaf index and the Deegan-Packel index of weighted majority games.

1. Introdution

In 1944, John von Neumann and Oskar Morgenstern studied the distribution of power in voting systems in their book *Theory of Games and Economic Behavior* [19]. They dealt with a "simple game" in which the only goal is "winning". This is an abstraction of the constitutional political machinery for voting. This paper deals with the weighted majority game, which is a familiar example of voting systems.

In 1960s, U.S. Supreme Court handed down a series of "one person one vote" decisions. After that, calculations of power indices using real data were carried out and presented as evidence in the courtroom. For example, the courts in New York State have accepted the Banzhaf index (also called the Coleman value or Chow parameters) as an appropriate measure for weighted voting systems. The calculation normally requires the aid of a computer and so many counties in U.S. hire specialized consultants, mathematicians or computer scientists (see [13]).

In this paper, we discuss some algorithms for calculating power indices. In Section 2, we define weighted majority games and related concepts. Section 3 defines three power indices, the Shapley-Shubik power index, the Banzhaf index and the Deegan-Packel index. Section 4 shows complexity classes of the problems for calculating power indices. In Sections 5,6 and 7, we discuss dynamic programming techniques, enumeration methods and Monte Carlo methods.

In papers [21,22], Owen proposed approximation algorithms for calculating the Shapley-Shubik indices and the Banzhaf indices based on *multilinear extensions*. The methods are written in Owen's book [23] (Chapter XII) in detail with a numerical example of the Presidential Election Game in United States. So, we omit Owen's approximation algorithms.

2. Weighted Majority Games

In this paper, we consider a special class of cooperative games called *weighted majority games*. Let $N = \{1, 2, \ldots, n\}$ be a set of players. A subset of players is called a *coalition*. A weighted majority game G is defined by a sequence of nonnegative numbers

$$G = [q; w_1, w_2, \ldots, w_n],$$

where we may think of w_i as the number of votes, or *weight* of player i and q as the threshold or *quota* needed for a coalition to win. In this paper, we assume that $q, w_1, w_2, \ldots, w_n$ are nonnegative integers and $(1/2)\sum_{i=1}^{n} w_i < q \leq \sum_{i=1}^{n} w_i$. We also assume the following property.

Assumption 1 *The set of players are arranged to satisfy the inequalities*

$$w_1 \geq w_2 \geq \cdots \geq w_n.$$

A coalition S is called a *winning* coalition when the inequality $\sum_{i \in S} w_i \geq q$ holds. The family of all the winning coalitions is denoted by $\mathcal{W}(G)$, or $\mathcal{W}$ when there is no ambiguity. A coalition S is called a *losing* coalition if S is not winning. A minimal coalition in the family of winning coalitions is called a *minimal winning coalition.* For any family of coalitions $\mathcal{F} \subseteq 2^N$, we denote the family of minimal coalitions in $\mathcal{F}$ by $\min \mathcal{F}$. So, the family of minimal winning coalitions is denoted by $\min \mathcal{W}$. For any player i, the family of all the winning coalitions including i is denoted by $\mathcal{W}^i_+$ and the family of all the winning coalitions excluding i is denoted by $\mathcal{W}^i_-$. For any coalition $S \in \mathcal{W}^i_-$, the coalition $S \cup \{i\}$ is winning and so we have the inequality $|\mathcal{W}^i_-| \leq |\mathcal{W}^i_+|$. Given a weighted majority game G defined on the set of players N, the *characteristic function* $v_G : 2^N \to \{0,1\}$ is defined as

$$v_G(S) = \begin{cases} 1 & (S \in \mathcal{W}), \\ 0 & (S \notin \mathcal{W}). \end{cases}$$

When there is no ambiguity, we denote the characteristic function by v. For any set S and a singleton $\{e\}$, we denote $S \cup \{e\}$ by $S + e$ and $S \setminus \{e\}$ by $S - e$. For any set S, both $|S|$ and $\#S$ denote the cardinality (the number of elements) of S.

For further discussions of (general) voting games, see books [23], [18] and [27] for example.

3. The Power Indices

Sometimes, similar weighted majority games have very different structures of winning coalitions. For example, let us consider games $G_n = [5n-4; 10, 10, \ldots, 10, 1]$ with $n \geq 2$ players. If n is an even number, $v(S+n) = v(S-n)$ for all $S \subseteq N$ and so player n can contribute nothing to any coalition. If n is an odd number, a coalition is winning if and only if the size is greater than $n/2$ and so each player seems to have the same power.

Measure of power plays a useful role in assessing the character of players in the weighted majority games. In this paper, we deal with three measures of power or *power indices*; (1) the Shapley-Shubik index [25,26], (2) the Banzhaf index [1,8], and (3) the Deegan-Packel index [5].

3.1. The Shapley-Shubik index

The Shapley value is a solution concept of n-person cooperative games derived from a set of axioms [25,7]. In 1954, Shapley and Shubik discussed the Shapley value of voting games, and so the Shapley value of voting games is called the *Shapley-Shubik index* (S-S index) [26].

Let $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ be a permutation defined on the set of players N. We say that player σ_i is the *pivot* of the permutation σ, if $\{\sigma_1, \sigma_2, \ldots, \sigma_{i-1}\}$ is a losing coalition and $\{\sigma_1, \sigma_2, \ldots, \sigma_{i-1}, \sigma_i\}$ is a winning coalition. When we assume that all the permutations have the same probability $1/n!$, the pivot probability

$$\varphi_i = (1/n!) \sum \{(|S|-1)!(n-|S|)! : S \in \mathcal{W}, S - i \notin \mathcal{W}\}$$

is the Shapley-Shubik index (S-S index) of player i. We denote the S-S index of player i by φ_i. It is easy to show that the inequalities $w_1 \geq w_2 \geq \cdots \geq w_n$ imply that the S-S indices satisfy the inequalities $\varphi_1 \geq \varphi_2 \geq \cdots \geq \varphi_n$.

3.2. The Banzhaf index

The Banzhaf index (Bz index) is proposed by Banzhaf in 1965 [1]. The index is also called the Coleman value or Chow parameters. Dubey and Shapley discussed axioms to derive the Bz index [8].

A pair of coalitions $(S+i, S)$ is called a *swing* for player i, if $S+i$ is winning and $S \subseteq N-i$ is losing. The number of swings of player i is called the *raw Banzhaf index* of player i. If we assign probability $1/2^{n-1}$ to each coalition $S \subseteq N-i$, the swing probability

$$\begin{aligned} \beta_i &= (1/2^{n-1})\#\{S \subseteq N-i : S+i \in \mathcal{W},\ \ S \notin \mathcal{W}\} \\ &= (1/2^{n-1})\sum\{v(S+i)-v(S) : S \subseteq N-i\} \end{aligned}$$

is called the *Banzhaf index* (Bz index) of player i and denoted by β_i.

Clearly, the Bz index β_i is equivalent to the value $\beta_i = (1/2^{n-1})(|\mathcal{W}^i_+| - |\mathcal{W}^i_-|)$, which is called the Dahlingham index [8]. The Bz indices do not add up to 1. The assumption $w_1 \geq w_2 \geq \cdots \geq w_n$ implies the inequalities $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_n$.

3.3. The Deegan-Packel index

In 1978, Deegan and Packel proposed a power index based on the minimal winning coalitions [5]. The Deegan-Packel index (D-P index) γ_i of player i is defined by

$$\gamma_i = \begin{cases} (1/|\min \mathcal{W}|)\sum\{1/|S| : i \in S \in \min \mathcal{W}\} & (\text{ if } \{S : i \in S \in \min \mathcal{W}\} \neq \emptyset), \\ 0 & (\text{ if } \{S : i \in S \in \min \mathcal{W}\} = \emptyset). \end{cases}$$

The sum total of D-P indices is equal to 1. Even if the weights satisfy the inequalities $w_1 \geq w_2 \geq \cdots \geq w_n$, the D-P indices do not always satisfy the inequalities $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_n$. For example, the weighted majority game $[26; 20, 6, 5, 2, 1, 1, 1, 1]$ with eight players has seven minimal winning coalitions

$$\{\{1,2\},\{1,3,4\},\{1,3,5\},\{1,3,6\},\{1,3,7\},\{1,3,8\},\{1,4,5,6,7,8\}\}$$

and so corresponding D-P indices are

$$\gamma = \left(\frac{1}{3}, \frac{1}{14}, \frac{5}{21}, \frac{1}{14}, \frac{1}{14}, \frac{1}{14}, \frac{1}{14}, \frac{1}{14}\right).$$

In this case, inequalities $w_2 > w_3 > w_4$ and $\gamma_3 > \gamma_2 = \gamma_4$ hold.

3.4. Numerical example

Here we show a numerical example of a weighted majority game and corresponding power indices. The present method of choosing a president for the United States of America is an interesting example of the weighted majority games. The voters within each state elect members of Electoral College, who in turn vote for the president. We assume that all electors from a given state vote together. Then it becomes a 51-player weighted majority game defined as;

$$[270; 54, 33, 32, \ldots, 5, 5, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3],$$

(for complete description, see Table 1) where the weights (the numbers of electoral votes) are obtained from 1996 census data on the following internet home page.

Table 1: Electoral College Game.

Electoral votes	Number of states	S-S	Bz	D-P
54	1	0.10813	0.46645	0.02125
33	1	0.06297	0.25959	0.02020
32	1	0.06093	0.25121	0.02017
25	1	0.04693	0.19386	0.01998
23	1	0.04301	0.17784	0.01992
22	1	0.04106	0.16989	0.01990
21	1	0.03911	0.16196	0.01987
18	1	0.03333	0.13835	0.01979
15	1	0.02762	0.11496	0.01971
14	1	0.02573	0.10721	0.01969
13	2	0.02385	0.09948	0.01966
12	2	0.02197	0.09176	0.01964
11	4	0.02011	0.08406	0.01961
10	2	0.01825	0.07637	0.01959
9	2	0.01639	0.06870	0.01956
8	6	0.01454	0.06104	0.01953
7	3	0.01270	0.05338	0.01951
6	2	0.01087	0.04574	0.01948
5	4	0.00904	0.03811	0.01946
4	6	0.00722	0.03048	0.01943
3	8	0.00541	0.02285	0.01938

http://www.fec.gov/pubrec/map1.htm
Table 1 shows the power indices of this game. The number of minimal winning coalitions of this game is 51,476,401,254,318.

The above numerical result is calculated by Ogawa [20]. His computer program is based on the dynamic programming technique in Section 5 and calculates the power indices on the table in a few minutes by using a popular engineering workstation. One can calculate the power indices on the following internet homepages.

http://www.misojiro.t.u-tokyo.ac.jp/~ tomomi/voting/voting.html
http://algo.kuamp.kyoto-u.ac.jp/tokutei98/database/index.html

4. Complexity for Calculating Power Indices

When player i satisfies that $\forall S \subseteq N - i$, $v(S) = v(S + i)$, player i is called a *dummy player*. When a pair of players i, j satisfies $\forall S \subseteq N \setminus \{i, j\}$, $v(S+i) = v(S+j)$, we say that the pair is symmetric. The following theorem shows the hardness for checking these characters of players.

Theorem 2 *Under Assumption 1, both of the following two problems are $\mathcal{NP}$-complete.*

(1) Is player n not a dummy player?

(2) Is the pair of players 1 and 2 not symmetric?

Proof : Player n is not a dummy player if and only if there exists a coalition $S \subseteq N - \{n\}$ satisfying that S is losing and $S \cup \{n\}$ is winning. The pair of players 1 and 2 is not symmetric if and only if there exists a coalition $S \subseteq N - \{1, 2\}$ satisfying that $S \cup \{1\}$ is winning and $S \cup \{2\}$ is losing, since $w_1 \geq w_2$. Thus, both problems have polynomial size YES certificates and so belong to class $\mathcal{NP}$.

We show a polynomial time reduction of the *partition problem* [9,11] defined below to problems (1) and (2). Given a set of positive integers $\{a_1, a_2, \ldots, a_k\}$, the partition problem checks the existence of an index subset $T \subseteq \{1, 2, \ldots, k\}$ satisfying the condition that $\sum_{i \in T} a_i = (1/2)M$ where $M = \sum_{i=1}^{k} a_i$. To prove $\mathcal{NP}$-completeness of (1) and (2), we construct two games

$$G_1 = [q_1; a_1, a_2, \ldots, a_n, 1] \text{ and } G_2 = [q_2; M+1, M, a_1, \ldots, a_n]$$

where q_1 and q_2 are the minimum integers which is greater than the half of the sum of all the voting weights, respectively. Then it is easy to show that the following three statements are equivalent; (a) the given partition problem has YES answer, (b) player n of game G_1 is not a dummy player, and (c) the pair of players 1 and 2 of G_2 is not symmetric. □

The $\mathcal{NP}$-completeness of problem (1) is described in the book [9] by Garey and Johnson without proof (see p.280, problem [MS8]). Problem (2) is discussed in [16].

For the S-S index, the Bz index, and the D-P index, player i is dummy if and only if the corresponding index of player i is equal to 0. It implies the following.

Corollary 3 *Under Assumption 1, all the problems for calculating player n's indices φ_n(the S-S index), β_n(the Bz index), and γ_n(the D-P index) are $\mathcal{NP}$-hard.*

The above corollary says that it is hard to calculate power indices of a weighted majority game when the input size is large.

For the S-S index and the Bz index, a pair of players is symmetric if and only if the corresponding pairs of indices are equivalent. So, we have the following.

Corollary 4 *Under Assumption 1, both of the problems for calculating the pair of S-S indices (φ_1, φ_2) and the pair of Bz indices (β_1, β_2) are $\mathcal{NP}$-hard.*

The above result implies that even if we need to calculate indices of biggest and second biggest players, it is still hard. For the D-P indices, the situation is complicated. Even if the D-P indices of a pair of players are equivalent, the pair is not always symmetric (see the numerical example in the subsection of the definition of the D-P index).

In [6], Deng and Papadimitriou proved the following result.

Theorem 5 *Under Assumption 1, both of the problems to calculate power indices φ_n and β_n are #P-complete.*

The paper [6] proved the above theorem in the case of the S-S index. However, we can show the case of the Bz index in a similar way.

Lastly, we consider a well-solvable special case.

Theorem 6 *Under Assumption 1, the following statements are equivalent.*
(1) The pair of players 1 and n is symmetric.
(2) Each pair of players is symmetric.
(3) There exists a positive integer k satisfying that $\mathcal{W} = \{S \subseteq N : |S| \geq k\}$.
(4) There exists a positive integer k satisfying that

$$w_{n-(k-1)} + \cdots + w_{n-1} + w_n \geq q > w_1 + w_2 + \cdots + w_{k-1}. \tag{4.1}$$

Proof: Obviously, (1) and (2) are equivalent. It is easy to show that (3) implies (2). Next, we show (2) implies (3). Put $k = \min\{|S| : S \in \mathcal{W}\}$ and $S^* = \arg\min\{|S| : S \in \mathcal{W}\}$.

If there exists a losing coalition S' satisfying $|S'| \geq |S^*|$, then any pair of players (i, j) satisfying $i \in S' \setminus S^*$, and $j \in S^* \setminus S'$ implies $S' - i + j \notin \mathcal{W}$, since the pair i, j is symmetric. By updating S' by $S' - i + j$ and applying the above procedure iteratively, we can construct a losing coalition containing S^*. Contradiction.

It is easy to show that (3) implies (4). Lastly, we show that (4) implies (3). For any coalition S satisfying $|S| \geq k$, the inequality $\sum_{i \in S} w_i \geq w_{n-(k-1)} + \cdots + w_n \geq q$ holds and so $S \in \mathcal{W}$. If a coalition S' satisfies $|S'| < k$, then $\sum_{i \in S'} w_i \leq w_1 + \cdots + w_{k-1} < q$ and so S' is losing. □

The above theorem shows that we can check the symmetricity of player 1 and player n in O(n) time.

5. Dynamic Programming

In this section, we assume that $w_i < q$ for each player i. We partition the set of players N into coalitions $N_1, N_2, \ldots, N_z$ satisfying that:

$$
\begin{aligned}
&(1) \quad N_1 \cup N_2 \cup \cdots \cup N_z = N, \quad N_x \cap N_y = \emptyset \ (x \neq y), \\
&(2) \quad 1 \leq \forall x < \forall y \leq z, \ \forall i \in N_x, \ \forall j \in N_y, \ w_i > w_j, \\
&(3) \quad 1 \leq \forall x \leq z, \ \forall i, \forall j \in N_x, \ w_i = w_j.
\end{aligned}
$$

In addition, we denote the cardinality $|N_x|$ by n_x and the weight of a player in N_x by $\overline{w}_x$.

5.1. Preprocedure

Here we describe the preprocedure. The preprocedure solves the following two problems; (1) problem for constructing the set of all dummy players, and (2) problem for counting the number of all the minimal winning coalitions. If there exist dummy players, we can delete the players without changing power indices (the S-S index, the Bz index and the D-P index). Thus, by removing all the dummy players, we may reduce the computational efforts required at the main procedure described in the next subsection. We use the result of the problem (2) when we calculate the D-P index in the main procedure.

The following property plays an important role to construct the set of all the dummy players.

Theorem 7 *Under Assumption 1, player j is a dummy player if and only if the inequality $\alpha_j + w_j + w_{j+1} + \cdots + w_n < q$ holds where*

$$
\alpha_j = \max\left\{\sum_{i \in S} w_i \ : S \subseteq \{1, 2, \ldots, j-1\}, \ S \notin \mathcal{W}\right\}.
$$

Proof: First, we show that $\alpha_j + w_j + w_{j+1} + \cdots + w_n \geq q$ implies that player j is not a dummy player. From the assumption, there exists a player $k \in \{j, j+1, \ldots, n\}$ satisfying that

$$
\alpha_j + w_j + w_{j+1} + \cdots + w_{k-1} < q \leq \alpha_j + w_j + w_{j+1} + \cdots + w_k.
$$

Thus we have

$$
\alpha_j + w_{j+1} + \cdots + w_k \leq \alpha_j + w_j + w_{j+1} + \cdots + w_{k-1} < q \leq \alpha_j + w_j + w_{j+1} + \cdots + w_k
$$

and so player j is not a dummy player.

Next, we show the inverse implication. Since player j is not a dummy player, there exists a coalition $S \subseteq N - j$ such that $S \notin \mathcal{W}$ and $S + j \in \mathcal{W}$. Put $S' = S \cap \{1, 2, \ldots, j-1\}$. Then we have the inequalities

$$\sum_{i \in S'} w_i \leq \sum_{i \in S} w_i < q \leq \sum_{i \in S} w_i + w_j \leq \sum_{i \in S'} w_i + w_j + w_{j+1} + \cdots + w_n.$$

The definition of α_j implies that $\sum_{i \in S'} w_i \leq \alpha_j$ and so the following inequalities

$$q \leq \sum_{i \in S'} w_i + w_j + w_{j+1} + \cdots + w_n \leq \alpha_j + w_j + w_{j+1} + \cdots + w_n$$

are satisfied. ❐

From the above theorem, we can identify the set of all the dummy players from the sequence of values $(\alpha_2, \alpha_3, \ldots, \alpha_n)$. The problems for calculating the values $(\alpha_2, \alpha_3, \ldots, \alpha_n)$ are called knapsack problems. There exists a pseudo polynomial time algorithm for knapsack problems [3]. The following preprocedure finds the sequence $(\alpha_2, \alpha_3, \ldots, \alpha_n)$ simultaneously.

For any pair of non-negative integers w and x, $c(w, x)$ denotes the number

$$\#\{S \subseteq N : \sum_{i \in S} w_i = w,\ S \cap N_x \neq \emptyset,\ S \cap N_{x+1} = \cdots = S \cap N_z = \emptyset\}.$$

Then the number of minimal winning coalitions is denoted by $\sum_{x=1}^{z} \sum_{w=q}^{q+\overline{w}_x-1} c(w, x)$. A player $i \in N_x$ is a dummy player if and only if

$$\alpha'_x + \overline{w}_x n_x + \overline{w}_{x+1} n_{x+1} + \cdots + \overline{w}_z n_z < q,$$

where

$$\alpha'_x = \max\{w : 0 \leq w \leq q-1 \text{ and } 1 \leq \exists y \leq x-1, c(w, y) > 0\}.$$

Here we describe an algorithm for calculating the values $\{c(w, x) : 0 \leq w \leq q - 1,\ 1 \leq x \leq z\}$ by using ordinary dynamic programming techniques. At the end of xth iteration of the following algorithm, the variables c^* and α^* represent the values $c(w, 1) + c(w, 2) + \cdots + c(w, x)$ and α'_x, respectively.

```
begin
  for w = 1, 2, ..., q − 1 do  c(w) := 0;
  c(0) := 1;  c* := 0;  α* := 0;  w' := w_1 + ··· + w_n;
  for x = 1, 2, ..., z do  {
    w' := w' − w̄_x * n_x;
    for  w = α*, α* − 1, ..., 0 do  {
      if  c(w) > 0 then  {
        c' := 1;  (In the following iterations, c' = _{n_x}C_y.)
        for  y = 1, 2, ..., n_x do  {
          c' := c' * (n_x − y + 1)/y;
          if  w + w̄_x * y ≤ q − 1 then  {
            c(w + w̄_x * y) := c(w + w̄_x * y) + c(w) * c';
            α* := max{α*, w + w̄_x * y}
          };
          else if  q ≤ w + w̄_x * y ≤ q − 1 + w̄_x then  c* := c* + c(w) * c'
```

```
      } } };
      if α* + w' ≤ q − 1 then {output (c*, x + 1); STOP }
   };
   output (c*, "no dummy player")
end
```

If the above algorithm output the pair of values $(c^*, x+1)$ and stopped, then the value c^* is equal to the number of all the minimal winning coalitions and the set of all the dummy players is equal to $N_{x+1} \cup N_{x+2} \cup \cdots \cup N_z$. The time complexity of this algorithm is bounded by $\mathrm{O}(nq)$ and the space complexity is bounded by $\mathrm{O}(n+q)$.

5.2. Dynamic programming for calculating power indices

There exist dynamic programming algorithms for calculating the S-S index and the Bz index [4,15,12].

In the following, we modify the algorithms in [4,15,12] and construct a dynamic programming algorithm for calculating the S-S index, the Bz index and the D-P index simultaneously in pseudo polynomial time.

For any player i, $c_i(w,t,x)$ denotes the number

$$\#\{S \subseteq N - i : \sum_{j \in S} w_j = w,\ |S| = t,\ S \cap N_x \neq \emptyset,\ S \cap N_{x+1} = \cdots = S \cap N_z = \emptyset\}.$$

Then the power indices of the player $i \in N_y$ is described as follows,

$$\begin{aligned}
\varphi_i &= \sum_{t=1}^{n-1} \sum_{w=q-w_i}^{q-1} \sum_{x=1}^{z} \frac{t!(n-t-1)!}{n!} c_i(w,t,x),\\
\beta_i &= (1/2^{n-1}) \sum_{t=1}^{n-1} \sum_{w=q-w_i}^{q-1} \sum_{x=1}^{z} c_i(w,t,x),\\
\gamma_i &= \left(\sum_{x=1}^{y} \sum_{w=q-w_i}^{q-1} \sum_{t=1}^{n-1} \frac{c_i(w,t,x)}{t+1} + \sum_{x=y+1}^{z} \sum_{w=q-w_i}^{q-1-w_i+w_x} \sum_{t=1}^{n-1} \frac{c_i(w,t,x)}{t+1} \right) \frac{1}{|\min \mathcal{W}|}.
\end{aligned}$$

In the above equalities, we used the property i ● N_y in the description of γ_i.

Now we describe our algorithm.

```
begin
   for (w, t) ∈ {1, 2, ..., q − 1} × {1, 2, ..., n − 1} do c(w, t) = 0;
   c(0, 0) := 1; t* := 0; φ* := 0; β* := 0; γ* := 0;
   for t = 0, 1, ..., n − 1 do α(t) := 0;
   for x = 1, 2, ..., z do {
      y' := |N_x − i|;
      for t = t*, t* − 1, ..., 0 do {
         for w = α(t), α(t) − 1, ..., 0 do {
            if c(w, t) > 0 then {
               c' := 1; (In the following iterations, c' =_{y'} C_y.)
               for y = 1, 2, ..., y' do {
                  c' := c' * (y' − y + 1)/y;
                  if w + w̄_x * y ≤ q − 1 then {
                     c(w + w̄_x * y, t + y) := c(w + w̄_x * y, t + y) + c(w, t) * c';
                     α(t + y) := max{α(t + y), w + w̄_x * y}
```

```
            };
            if q − w_i ≤ w + w̄_x * y ≤ q − 1 then {
               φ* := φ* + (t + y)!(n − t − y − 1)! c(w,t) * c';
               β* := β* + c(w,t) * c'};
            if q − w_i ≤ w + w̄_x * y ≤ min{q − 1, q − 1 − w_i + w̄_x} then {
               γ* := γ* + (1/(t + y + 1)) * c(w,t) * c'}
   } } } };
   t* := t* + y'
};
output (φ*/n!, β*/2^{n−1}, γ*/|min W|) end
```

The time complexity and space requirement of the above algorithm is bounded by $O(n^2q)$ and $O(nq)$, respectively.

6. Enumeration Algorithm

In this section, we propose enumeration algorithms for calculating the power indices.

First, we define some notations. Given a positive real vector $\boldsymbol{a} = (a_1, a_2, \ldots, a_k)^\top$ and a real number b, we define a family of indices $\mathcal{F}(\boldsymbol{a}, b) \equiv \{F \subseteq \{1, 2, \ldots, k\} : \sum_{i \in F} a_i \geq b\}$. The vector $\boldsymbol{f}(\boldsymbol{a}, b) \equiv (f_1, f_2, \ldots, f_k)^\top$ consists of elements

$$f_i = \#\{S \in \mathcal{F}(\boldsymbol{a}, b) : |S| = i\} \quad (i = 1, 2, \ldots, k).$$

6.1. Enumeration algorithm for calculating D-P indices

Clearly from the definition, we can calculate the D-P indices by generating all the minimal winning coalitions. In this subsection, we consider an efficient algorithm for generating all the minimal winning coalitions.

We consider the problem for generating all the minimal sets in $\mathcal{F}(\boldsymbol{a}, b)$ where $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)^\top$ is a positive integer vector and b is an integer. The algorithm uses the divide and conquer technique based on the following theorem.

Theorem 8 *Let $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)^\top$ be a positive real vector and b be a real number satisfying the conditions that $a_1 \geq a_2 \geq \cdots \geq a_n > 0$ and $a_1 + a_2 + \cdots + a_n \geq b$. Then the pair of families*

$$\mathcal{F}^1_+ \equiv \{S : 1 \in S \in \mathcal{F}(\boldsymbol{a}, b)\}, \quad \mathcal{F}^1_- \equiv \{S : 1 \notin S \in \mathcal{F}(\boldsymbol{a}, b)\},$$

satisfies that if $\emptyset \notin \mathcal{F}(\boldsymbol{a}, b)$, then $\min \mathcal{F}(\boldsymbol{a}, b)$ is equal to the direct sum of $\min \mathcal{F}^1_+$ and $\min \mathcal{F}^1_-$.

Proof: It is easy to show that $\min \mathcal{F}(\boldsymbol{a}, b) \subseteq \min \mathcal{F}^1_+ \cup \min \mathcal{F}^1_-$ and $\min \mathcal{F}(\boldsymbol{a}, b) \supseteq \min \mathcal{F}^1_-$. If we assume that $\exists S \in \min \mathcal{F}^1_+ - \min \mathcal{F}(\boldsymbol{a}, b)$, then we have $S - 1 \in \mathcal{F}(\boldsymbol{a}, b)$ and so $\sum_{i \in S-1} a_i \geq b$. The inequalities $a_1 \geq \cdots \geq a_n$ imply that if there exists a player $j \in S-1$, then $\sum_{i \in S-j} a_i \geq \sum_{i \in S-1} a_i \geq b$ and it contradicts with the condition $S \in \min \mathcal{F}^1_+$. From the above $S = \{1\}$ and so $S - 1 = \emptyset \in \mathcal{F}(\boldsymbol{a}, b)$. Contradiction. □

By using the $(n-1)$-vector $\boldsymbol{a}' = (a_2, a_3, \ldots, a_n)^\top$, we can describe the pair of families $\mathcal{F}^1_+$ and $\mathcal{F}^1_-$ appeared in the above proof as follows;

$$\mathcal{F}^1_+ = \{S + 1 : S \in \min \mathcal{F}(\boldsymbol{a}', b - a_1)\}, \quad \mathcal{F}^1_- = \mathcal{F}(\boldsymbol{a}', b).$$

The above discussion directly implies a divide and conquer type algorithm.

Input: positive real vector $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)^\top$ and a positive real number b satisfying $a_1 + a_2 + \cdots + a_n \geq b$ where $a_1 \geq a_2 \geq \cdots \geq a_n$
Output: all the minimal sets in the family $\mathcal{F}(\boldsymbol{a}, b)$

```
procedure ENUMERATE (S', n', b');
   begin
      if n' = n then {output S' ∪ {n}; return }
      else {
         if (a_{n'+1} + ··· + a_n) ≥ b' then ENUMERATE (S', n' + 1, b');
         if a_{n'} ≥ b' then {output S' ∪ {n'}}
         else ENUMERATE (S' ∪ {n'}, n' + 1, b' − a_{n'}) ;
         return }
   end ;
begin(main routine)
   ENUMERATE (∅, 1, b)
end.
```

The above recursive algorithm either outputs a minimal set or calls **procedure** ENUMERATE at least once. The height of the recursive tree of the algorithm is less than n and so the number of calls of **procedure** ENUMERATE is less than or equal to n times the number of minimal sets. The time complexity of each line in **procedure** ENUMERATE is bounded by O(1) when we update the value $(a_{n'+1} + \cdots + a_n)$ at each iteration. Thus, the time complexity of the above algorithm is $O(n|\min \mathcal{F}(\boldsymbol{a}, b)|)$ and memory complexity is $O(n)$.

Now we construct our algorithm for calculating the D-P indices as follows. When the above algorithm generates an additional minimal winning coalition, we update the (current) D-P indices. The algorithm updates the current D-P indices of all the players and so we can obtain the D-P indices of all the players in $O(n|\min \mathcal{W}|)$ time and $O(n)$ space.

6.2. Enumeration algorithm for calculating S-S indices and Bz indices

In this subsection, we describe algorithms for calculating the S-S index and the Bz index of player i. We denote the $(n-1)$-vector $(w_1, w_2, \ldots, w_{i-1}, w_{i+1}, \ldots, w_n)^\top$ by $\boldsymbol{w}' = (w'_1, w'_2, \ldots, w'_{n-1})^\top$. The pair of vectors $\boldsymbol{f}(\boldsymbol{w}', q - w_i)$ and $\boldsymbol{f}(\boldsymbol{w}', q)$ are denoted by $\boldsymbol{f}^+$ and $\boldsymbol{f}^-$ respectively. Then the definition of power indices directly imply that

$$\varphi_i = \sum_{t=1}^{n-1} \frac{t!(n-t-1)!}{n!}(f_t^+ - f_t^-), \quad \beta_i = (1/2^{n-1}) \sum_{t=1}^{n-1} (f_t^+ - f_t^-).$$

So, we only need to construct an algorithm for solving the following problem.

Input: a positive real vector $\boldsymbol{a} = (a_1, a_2, \ldots, a_k)^\top$ and an integer number b satisfying $a_1 \geq a_2 \geq \cdots \geq a_n > 0$ and $a_1 + a_2 + \cdots + a_n \geq b$
Output: vector $\boldsymbol{f}(\boldsymbol{a}, b)$

In the following algorithm, we denote the vector of binomial coefficients $({}_k\mathrm{C}_0, {}_k\mathrm{C}_1, \ldots, {}_k\mathrm{C}_k)^\top$ by θ_k. The k-dimensional zero vector is denoted by $\mathbf{0}_k$.

```
function SHELLING (n', b');
(comment : a_{n'} + ··· + a_n ≥ b' holds)
  begin
    if n' = n then return ((0, 1))
```

else {
if $(a_{n'+1} + \cdots + a_n) \geq b'$ **then** $\boldsymbol{f}^1 :=$ SHELLING $(n'+1, b')$ **else** $\boldsymbol{f}^1 := \boldsymbol{0}_{n-n'}$;
(**comment** : all the index subsets excluding n' are considered)
if $a_{n'} \geq b'$ **then** $\boldsymbol{f}^2 := \theta_{n-n'}$ **else** $\boldsymbol{f}^2 :=$ SHELLING $(n'+1, b'-a_{n'})$;
(**comment** : all the index subsets including n' are considered)
$\boldsymbol{f}' := (f_1^1, f_2^1 + f_1^2, f_3^1 + f_2^2, \ldots, f_{n-n'}^1 + f_{n-n'-1}^2, f_{n-n'}^2)$;
return $(\boldsymbol{f}')$ }
end;
begin(main routine)
SHELLING $(1, b)$;
end.

In the above recursive algorithm, the number of calls of **procedure** SHELLING is bounded by n times the number of minimal sets, i.e., the number of outputs. When we construct binomial sequences $\theta_1, \theta_2, \ldots, \theta_n$ in a preprocedure, each line in **function** is bounded by $O(n)$. Thus the total time complexity is $O(n^2|\min \mathcal{F}(\boldsymbol{a}, b)|)$. Since the height of the recursive call tree is less than n, we only need to maintain at most n vectors whose dimension is less than or equal to n. Each element in the vector is an integer less than or equal to $n!$ and so the space complexity is also bounded by a polynomial of n.

If we apply the above algorithm for calculating $\boldsymbol{f}^+$ and $\boldsymbol{f}^-$, the time complexity is $O(n^2|\min \mathcal{W}_+^i| + n^2|\min \mathcal{W}_-^i|) = O(n^2|\min \mathcal{W}|)$, where the preprocedure for constructing the binomial sequences requires additional $O(n^2)$ computational efforts.

From the above discussions, we can construct an algorithm for calculating the S-S index and the Bz index of player i which requires $O(n^2|\min \mathcal{W}|)$ time and $O(n^2)$ space.

7. Monte Carlo Method

In this section, we consider the most likelihood method based on Monte Carlo sampling for calculating the S-S indices and the Bz indices.

7.1. Monte Carlo method for calculating S-S indices

The definition of the S-S index says that the S-S index of player i is equal to the probability that the player i becomes the pivot player under the assumption that all the permutations have the same probability $1/n!$.

Choose p^* permutations of players at random. Let P_i be the number of permutations such that the player i is the pivot player. Then the random variables $P_1, \ldots, P_n$ have a multinomial distribution:

$$\Pr[P_1 = p_1, P_2 = p_2, \ldots, P_n = p_n] = \frac{p^*!}{p_1! p_2! \cdots p_n!} \varphi_1^{p_1} \varphi_2^{p_2} \cdots \varphi_n^{p_n}$$

where $\boldsymbol{p} = (p_1, \ldots, p_n)^\top$ is a non-negative integer vector satisfying that the sum of its components is equal to p^*. From the above, the maximum likelihood estimates of the S-S indices are $(p_1/p^*, p_2/p^*, \ldots, p_n/p^*)$. It is easy to see that the error of estimate goes to zero like $1/\sqrt{p^*}$. We refer to this method as the *simple Monte Carlo method.*

In 1960, Mann and Shapley proposed some variations of Monte Carlo sampling methods [14]. They applied their Monte Carlo methods for obtaining the S-S indices of the Electoral College game(see Section 3.4) . By calculating a gain of each method, they showed that a "cycling scheme" described below was effectual. A target player i is singled out, and the remaining players are placed in a random order. However, this order is then put through all of its cyclic permutations, and the player i is inserted in each position in each permutation. Thus $(N-1)N$ permutations are generated. We illustrate below the cycling sampling scheme for five players a, b, c, d and i.

i	a	b	c	d	i	b	c	d	a	i	c	d	a	b	i	d	a	b	c
a	i	b	c	d	b	i	c	d	a	c	i	d	a	b	d	i	a	b	c
a	b	i	c	d	b	c	i	d	a	c	d	i	a	b	d	a	i	b	c
a	b	c	i	d	b	c	d	i	a	c	d	a	i	b	d	a	b	i	c
a	b	c	d	i	b	c	d	a	i	c	d	a	b	i	d	a	b	c	i

The above four sets of permutations are based on four permutations $(a, b, c, d), (b, c, d, a)$, (c, d, a, b) and (d, a, b, c). In this example, $(5-1)5 = 20$ permutations are generated. The random variable P_i of player i of the cycling sampling method is defined as

$$P_i = \frac{1}{(N-1)N}\ (\textit{number of permutations such that player } i \textit{ is the pivot}).$$

It is easily seen that P_i is an unbiased estimate of S-S index φ_i.

The definition of S-S index directly shows that inequalities $w_1 \geq w_2 \geq \cdots \geq w_n$ (see Assumption 1) imply the inequalities $\varphi_1 \geq \varphi_2 \geq \cdots \geq \varphi_n$. Thus we need to improve the estimates under the monotonicity restriction. When we employ the simple Monte Carlo method, the set of maximum likelihood estimates of the S-S indices under monotonicity restriction is optimal solutions of the following problem:

$$\begin{aligned} \text{MLE}(\boldsymbol{p}) : \text{maximize} \quad & \sum_{i=1}^{n} p_i \log x_i \\ \text{subject to} \quad & x_1 \geq x_2 \geq \cdots \geq x_n \geq 0, & (7.1) \\ & x_1 + x_2 + \cdots x_n = 1, & (7.2) \end{aligned}$$

where the objective is the maximization of log-likelihood. In the following, we describe the solution method of this problem.

For any non-negative vector $\boldsymbol{p} = (p_1, \ldots, p_n)^\top$, the vector $\boldsymbol{p}^{\mathrm{C}} = (p_0^{\mathrm{C}}, p_1^{\mathrm{C}}, \ldots, p_n^{\mathrm{C}})^\top$ satisfying $p_0^{\mathrm{C}} = 0$ and $p_i^{\mathrm{C}} = p_1 + \cdots + p_i$ is called the *cumulative sum vector* of $\boldsymbol{p}$. Given a non-negative vector $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)^\top$, $f : [0, n] \to \mathbf{R}$ denotes the function defined by $f(x) \equiv \min_{g \in \mathcal{G}} g(x)$ where $\mathcal{G}$ is the set of continuous concave functions satisfying $\forall i \in \{0, 1, \ldots, n\}, \forall g \in \mathcal{G}, g(i) \geq p_i^{\mathrm{C}} = p_1 + p_2 + \cdots + p_i$. We denote the vector

$$(f(1) - f(0), f(2) - f(1), \ldots, f(n) - f(n-1))^\top$$

by $\widehat{\boldsymbol{p}} = (\widehat{p}_1, \widehat{p}_2, \ldots, \widehat{p}_n)^\top$. The vector $\widehat{\boldsymbol{p}}$ is called the *isotonic regression* of $\boldsymbol{p}$. Then we have the following theorem.

Theorem 9 *Given a positive vector $\boldsymbol{p} = (p_1, \ldots, p_n)^\top$, the vector $(1/p^*)\widehat{\boldsymbol{p}}$ is an optimal solution of MLE($\boldsymbol{p}$) where $p^* = p_1 + p_2 + \cdots + p_n$.*

Proof: We denote $h(\boldsymbol{p}, \boldsymbol{x}) \equiv \sum_{i=1}^{n} p_i \log x_i$. For any feasible solution $\boldsymbol{x}$ of MLE($\boldsymbol{p}$), we show that the following equality and inequalities hold;

$$h(\boldsymbol{p}, (1/p^*)\widehat{\boldsymbol{p}}) = h(\widehat{\boldsymbol{p}}, (1/p^*)\widehat{\boldsymbol{p}}) \geq h(\widehat{\boldsymbol{p}}, \boldsymbol{x}) \geq h(\boldsymbol{p}, \boldsymbol{x}).$$

(i) First, we show $h(\boldsymbol{p}, (1/p^*)\widehat{\boldsymbol{p}}) = h(\widehat{\boldsymbol{p}}, (1/p^*)\widehat{\boldsymbol{p}})$.

The definition of isotonic regression says that if $(\widehat{p}_i^{\mathrm{C}} - p_i^{\mathrm{C}})$ is positive, then $\widehat{p}_i = \widehat{p}_{i+1}$. So, for all $i = 1, 2, \ldots, n-1$, we have $(\widehat{p}_i^{\mathrm{C}} - p_i^{\mathrm{C}})(\log \widehat{p}_i - \log \widehat{p}_{i+1}) = 0$. These equalities imply that;

$$\begin{aligned} 0 &= \sum_{i=1}^{n-1}(\widehat{p}_i^{\mathrm{C}} - p_i^{\mathrm{C}})(\log \widehat{p}_i - \log \widehat{p}_{i+1}) + (\widehat{p}_n^{\mathrm{C}} - p_n^{\mathrm{C}}) \log \widehat{p}_n \\ &= (\widehat{p}_1^{\mathrm{C}} - p_1^{\mathrm{C}}) \log \widehat{p}_1 + \sum_{i=2}^{n}(-(\widehat{p}_{i-1}^{\mathrm{C}} - p_{i-1}^{\mathrm{C}}) + (\widehat{p}_i^{\mathrm{C}} - p_i^{\mathrm{C}})) \log \widehat{p}_i \\ &= (\widehat{p}_1 - p_1) \log \widehat{p}_1 + \sum_{i=2}^{n}(\widehat{p}_i - p_i) \log \widehat{p}_i = \sum_{i=1}^{n}(\widehat{p}_i - p_i) \log \widehat{p}_i \end{aligned}$$

and so $\sum_{i=1}^n p_i \log \widehat{p}_i = \sum_{i=1}^n \widehat{p}_i \log \widehat{p}_i$. Then we have the following result;

$$\begin{aligned} h(\boldsymbol{p}, (1/p^*)\widehat{\boldsymbol{p}}) &= \sum_{i=1}^{n} p_i \log(1/p^*)\widehat{p}_i = \log(1/p^*) \sum_{i=1}^{n} p_i + \sum_{i=1}^{n} p_i \log \widehat{p}_i \\ &= \log(1/p^*) \sum_{i=1}^{n} \widehat{p}_i + \sum_{i=1}^{n} \widehat{p}_i \log \widehat{p}_i = \sum_{i=1}^{n} \widehat{p}_i \log(1/p^*)\widehat{p}_i = h(\widehat{\boldsymbol{p}}, (1/p^*)\widehat{\boldsymbol{p}}). \end{aligned}$$

(ii) Next, we show that $h(\widehat{\boldsymbol{p}}, (1/p^*)\widehat{\boldsymbol{p}}) \geq h(\widehat{\boldsymbol{p}}, \boldsymbol{x})$ for any $\boldsymbol{x}$ feasible to MLE($\boldsymbol{p}$).

We discuss properties of optimal solutions of MLE($\widehat{\boldsymbol{p}}$). If we remove the constraint (7.1) from MLE($\widehat{\boldsymbol{p}}$), the K-K-T condition of the relaxed problem becomes

$$\widehat{p}_i / x_i - \lambda = 0 \ \ (i = 1, \ldots, n), \quad x_1 + x_2 + \cdots + x_n = 1.$$

The solution $\boldsymbol{x}^* = (1/p^*)\widehat{\boldsymbol{p}} = (\widehat{p}_1/p^*, \ldots, \widehat{p}_n/p^*)^\top$ and the Lagrange multiplier $\lambda^* = p^*$ satisfies the K-K-T condition and $\boldsymbol{x}^* = (1/p^*)\widehat{\boldsymbol{p}}$ is optimal to the relaxed problem. Since the solution $\boldsymbol{x}^* = (1/p^*)\widehat{\boldsymbol{p}}$ satisfies the constraint (7.1), $\boldsymbol{x}^* = (1/p^*)\widehat{\boldsymbol{p}}$ is also optimal to MLE($\widehat{\boldsymbol{p}}$). Any feasible solution $\boldsymbol{x}$ of MLE($\boldsymbol{p}$) is also feasible to MLE($\widehat{\boldsymbol{p}}$) and so we have $h(\widehat{\boldsymbol{p}}, (1/p^*)\widehat{\boldsymbol{p}}) = \sum_{i=1}^n \widehat{p}_i \log(\widehat{p}_i/p^*) \geq \sum_{i=1}^n \widehat{p}_i \log x_i = h(\widehat{\boldsymbol{p}}, \boldsymbol{x})$.

(iii) Lastly, we show that $h(\widehat{\boldsymbol{p}}, \boldsymbol{x}) \geq h(\boldsymbol{p}, \boldsymbol{x})$.

From the definition, the value $(\widehat{p}_{i-1}^{\mathrm{C}} - p_{i-1}^{\mathrm{C}})$ is non-negative. For any feasible solution $\boldsymbol{x}$ of MLE($\boldsymbol{p}$), constraint (7.1) implies that $\log x_i - \log x_{i+1} \geq 0$. Then we can derive the following inequalities;

$$0 \leq \sum_{i=1}^{n-1}(\widehat{p}_i^{\mathrm{C}} - p_i^{\mathrm{C}})(\log x_i - \log x_{i+1}) + (\widehat{p}_n^{\mathrm{C}} - p_n^{\mathrm{C}}) \log x_n = \sum_{i=1}^{n}(\widehat{p}_i - p_i) \log x_i$$

and so $h(\widehat{\boldsymbol{p}}, \boldsymbol{x}) = \sum_{i=1}^n \widehat{p}_i \log x_i \geq \sum_{i=1}^n p_i \log x_i = h(\boldsymbol{p}, \boldsymbol{x})$.

In the above, we have shown that for any feasible solution $\boldsymbol{x}$ of MLE($\boldsymbol{p}$), the following equality and inequalities hold;

$$h(\boldsymbol{p}, (1/p^*)\widehat{\boldsymbol{p}}) = h(\widehat{\boldsymbol{p}}, (1/p^*)\widehat{\boldsymbol{p}}) \geq h(\widehat{\boldsymbol{p}}, \boldsymbol{x}) \geq h(\boldsymbol{p}, \boldsymbol{x}).$$

Thus $(1/p^*)\widehat{\boldsymbol{p}}$ is optimal to MLE($\boldsymbol{p}$). □

The isotonic regression is discussed in the well-known book [2]. The above theorem is also discussed in the book in a more general setting.

From the convex hull of points $Q = \{(i, q_i) : i = 0, 1, \ldots, n\} \subseteq \mathbf{R}^2$, we can construct the isotonic regression of $\boldsymbol{p}$ easily. The combination of bucket sorting algorithm and Graham scan method [10] finds the convex hull of Q in O(n) time (see [24] Section 3.3.2). The pool-adjacent-algorithm described in [2] also finds the isotonic regression of $\boldsymbol{p}$ in O(n) time.

7.2. Monte Carlo method for calculating Bz indices

The definition of Bz index says that the Bz index of player i is equal to the probability that a pair of coalition $(S+i, S)$ becomes a swing for player i under the assumption that all the pairs have the same probability $1/2^{n-1}$.

Choose p^* coalitions excluding i at random. Let P_i be the number of coalitions such that $(S+i, S)$ are swings for player i. Then the set of maximum likelihood estimates of Bz indices under the monotonicity constraints is an optimal solution of the following problem;

$$\begin{array}{rll} \text{MLE'}(\boldsymbol{p}) : \text{maximize} & \sum_{i=1}^{n}(p_i \log x_i + (p^* - p_i)\log(1-x_i)) & \\ \text{subject to} & 1 \geq x_1 \geq x_2 \geq \cdots \geq x_n \geq 0, & (7.3) \end{array}$$

where the objective function is the log-likelihood function. Then the following theorem holds.

Theorem 10 *For any positive vector* $\boldsymbol{p} = (p_1, \ldots, p_n)^\top$, *the vector* $(1/p^*)\widehat{\boldsymbol{p}}$ *is optimal to MLE'*$(\boldsymbol{p})$ *where* $p^* = p_1 + p_2 + \cdots + p_n$.

Proof: The proof is almost the same as that of Theorem 9. Here we describe the outline of the proof. In the following, we denote $h(\boldsymbol{p}, \boldsymbol{x}) \equiv \sum_{i=1}^{n}(p_i \log x_i + (p^* - p_i)\log(1-x_i))$. We show that for any feasible solution $\boldsymbol{x}$ of MLE'$(\boldsymbol{p})$, the following equality and inequalities hold;

$$h(\boldsymbol{p}, (1/p^*)\widehat{\boldsymbol{p}}) = h(\widehat{\boldsymbol{p}}, (1/p^*)\widehat{\boldsymbol{p}}) \geq h(\widehat{\boldsymbol{p}}, \boldsymbol{x}) \geq h(\boldsymbol{p}, \boldsymbol{x}).$$

(i) The definition of the isotonic regression implies that when $(\widehat{p}_i^{\mathrm{C}} - p_i^{\mathrm{C}})$ is positive, then $\widehat{p}_i = \widehat{p}_{i+1}$. Since $h(\boldsymbol{p}, \boldsymbol{x})$ is a linear function of $\boldsymbol{p}$, $h(\boldsymbol{p}, (1/p^*)\widehat{\boldsymbol{p}}) = h(\widehat{\boldsymbol{p}}, (1/p^*)\widehat{\boldsymbol{p}})$.
(ii) It is easy to show that the solution $(1/p^*)\widehat{\boldsymbol{p}}$ is optimal to MLE'$(\widehat{\boldsymbol{p}})$. For any feasible solution $\boldsymbol{x}$ of MLE'$(\boldsymbol{p})$, $\boldsymbol{x}$ is also feasible to MLE'$(\widehat{\boldsymbol{p}})$ and so we have the inequality $h(\widehat{\boldsymbol{p}}, (1/p^*)\widehat{\boldsymbol{p}}) \geq h(\widehat{\boldsymbol{p}}, \boldsymbol{x})$.
(iii) From the definition of the isotonic regression, $(\widehat{p}_{i-1}^{\mathrm{C}} - p_{i-1}^{\mathrm{C}})$ is non-negative. For any feasible solution $\boldsymbol{x}$ of MLE'$(\boldsymbol{p})$, $x_i \geq x_{i+1}$ implies that $\log x_i - \log(1-x_i) \geq \log x_{i+1} - \log(1-x_{i+1})$. From the above, we have the following inequalities;

$$\begin{array}{rcl} 0 & \leq & \sum_{i=1}^{n-1}(\widehat{p}_i^{\mathrm{C}} - p_i^{\mathrm{C}})((\log x_i - \log(1-x_i)) - (\log x_{i+1} - \log(1-x_{i+1}))) \\ & + & (\widehat{p}_n^{\mathrm{C}} - p_n^{\mathrm{C}})(\log x_n - \log(1-x_n)) = \sum_{i=1}^{n}(\widehat{p}_i - p_i)(\log x_i - \log(1-x_i)) \end{array}$$

and so $h(\widehat{\boldsymbol{p}}, \boldsymbol{x}) \geq h(\boldsymbol{p}, \boldsymbol{x})$ is proved. □

8. Discussion

In this paper, we discussed algorithms for calculating some power indices of weighted majority games.

In many practical settings, we need to calculate the power indices for analyzing systems and committees; e.g., the parliamentary systems, electoral systems and the committee of stockholders. Here, we considered ordinary weighted majority games. However, many problems are left unsolved. We need to consider more complicated voting systems, (see [17], [23] Section XII.5, [18] and [27] for example). Efficient sensitivity analysis methods and approximate algorithms are also required.

Acknowledgment. We are indebted to Shigeo Muto and Rie Ono for several comments and suggestions. We are also grateful to Ryuusuke Ogawa for his helpful interest and computer program.

The work of Tomomi Matsui was supported in part by the Grant-in-Aid for Encouragement of Young Scientists (A) 11780323 and the Algorithm Engineering Project, Grant-in-Aid of MESSC Japan. The work of Yasuko Matsui was supported in part by the Grant-in-Aid for Encouragement of Young Scientists (A) 10780279.

References

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

[2] R. E. Barlow, D. J. Bartholomew, J. M. Bremner and H. D. Brunk: *Statistical Inference under Order Restrictions* (John Wiley and Sons, 1972).

[3] R. E. Bellman: Notes on the theory of dynamic programming IV – maximization over discrete sets. *Naval Research Logistics Quarterly*, **3** (1956) 67–70.

[4] S. J. Brams and P. J. Affuso: Power and size; a new paradox. *Mimeographed Paper*, 1975.

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

[6] X. Deng and C. H. Papadimitriou: On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, **19** (1994) 257–266.

[7] P. Dubey: On the uniqueness of the Shapley value. *International Journal of Game Theory*, **4** (1975) 131–139.

[8] P. Dubey and L. S. Shapley: Mathematical properties of the Banzhaf power index. *Mathematics of Operations Research*, **4** (1979) 99–131.

[9] M. R. Garey and D. S. Johnson: *Computers and Intractability: a guide to the theory of $\mathcal{NP}$-completeness* (W. H. Freeman and Company, 1979).

[10] R. L. Graham: An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, **1** (1972) 132–133.

[11] R. M. Karp: Reducibility among combinatorial problems. R. E. Miller and J. W. Thatcher (eds.): *Complexity of Computer Computations* (Plenum Press, New York, 1972), 85–103.

[12] W. F. Lucas: Measuring power in weighted voting systems. S. J. Brams, W. F. Lucas and P. D. Straffin (eds.): *Political and Related Models* (Springer-Verlag, 1983), 183–238.

[13] W. F. Lucas, J. C. Maceli, M. Hilliard and D. Housman: Reapportionment by weighted voting. *Technical report, School of Operations Research and Industrial Engineering, College of Engineering, Cornel University*, **533** (1982).

[14] I. Mann and L. S. Shapley: Values of large games, IV : evaluating the electoral college by Montecarlo techniques. *Technical Report, The RAND Corporation*, **RM-2651** (1960).

[15] I. Mann and L. S. Shapley: Values of large games, VI : evaluating the electoral college exactly. *Technical Report, The RAND Corporation*, **RM-3158-PR** (1962).

[16] Y. Matsui and T. Matsui: $\mathcal{NP}$-completeness for calculating power indices of weighted majority games. *Technical Report, Department of Mathematical Engineering and Information Physics, Faculty of Engineering, University of Tokyo*, **METR98-01**(1998).

[17] N. Megiddo: Nucleoluses of compound simple games. *SIAM Journal of Applied Mathematics*, **26** (1974) 607–621.

[18] 武藤滋夫, 小野理恵: 投票システムのゲーム分析 (日科技連, 1998); S. Muto and R. Ono: *Voting Systems - A Game Theoretic Analysis* (Nikkagiren, 1998).

[19] J. von Neumann and O. Morgenstern: *Theory of Games and Economic Behavior* (Princeton Univ. Press, 1944, 2nd ed. 1947, 3rd ed. 1953).

[20] R. Ogawa: Private communication.

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

[22] G. Owen: Multilinear extensions and the Banzhaf value. *Naval Research Logistic Quarterly*, **22** (1975) 741–750.

[23] G. Owen: *Game Theory, 3rd ed.* (Academic Press, 1995).

[24] F. P. Preparata and M. I. Shamos: *Computational Geometry* (Springer-Verlag, 1985); 浅野孝夫, 浅野哲夫 訳, 計算幾何学入門 (総研出版, 1992).

[25] L. S. Shapley: A value for n-person games. H. W. Kuhn and A. W. Tucker (eds.): *Contributions to the Theory of Games II* (Princeton University Press, 1953), 307–317.

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

[27] 鈴木光男, 武藤滋夫: 協力ゲームの理論 (東京大学出版, 1985); M. Suzuki and S. Muto: *Cooperative Game Theory* (The Tokyo Univ. Press, 1985).

Tomomi Matsui
University of Tokyo
Bunkyo-ku, Tokyo 113-8656, Japan.
E-mail: tomomi@misojiro.t.u-tokyo.ac.jp