

モンテカルロ法を用いた投票力指数の計算

潮田 優斗, 田中 雅人, 松井 知己

本稿では, 重み付き多数決ゲームにおける, Shapley-Shubik 指数を計算するモンテカルロ法について扱う. 最初に, 単純なモンテカルロ法について, その解析を行う. 次に, 新たな方法を提案し, モンテカルロ法の実行におけるサンプル数が, 単純なモンテカルロ法に比べ少ないことを示す.

キーワード: 重み付き多数決ゲーム, 投票力指数, Shapley-Shubik 指数, モンテカルロ法

1. はじめに

本稿では, 重み付き多数決ゲームにおける, Shapley-Shubik 指数を計算するモンテカルロ法について扱う. 重み付き多数決ゲームは, von Neumann and Morgenstern の著書 [1] において扱われている simple game の特殊ケースであり, われわれの身の周りでよく用いられる投票方法である『多数決』をモデル化したゲームである [1]. Shapley-Shubik 指数 (以下 S-S 指数) は, 1954 年に論文 [2] において提案された投票力指数であり, 特性関数ゲームに対する Shapley 値 [3] を拡張して得られる解概念である. 現在でも, 投票における参加者の力を評価する投票力指数として, さまざまな分野で用いられている.

重み付き多数決ゲームにおける S-S 指数の計算は NP-完全と呼ばれるクラスに属し, 簡単ではないことが知られている [4] (p. 280, problem [MS8] には証明なしでこれが掲載されている). また Deng and Papadimitriou [5] は, 重み付き多数決ゲームの S-S 指数の計算が #P-完全と呼ばれる計算困難なクラスに含まれることを示している. 重み付き多数決ゲームにおいて, 指定されたプレイヤーの S-S 指数が正の値をもつかを問う問題が NP-完全であることを Prasad and Kelly [6] は示しており, 指定されたプレイヤー対が対称的であるかを問う問題が NP-完全であることが Matsui and Matsui [7] で示されている. 重み付き多数決ゲームの S-S 指数を近似的に求めることも難しい問題となることが Elkind らによって示されている [8].

上記のように S-S 指数の計算が (理論的に) 困難で

あるためか, S-S 指数を計算するさまざまな方法が提案・研究されている. 最初期のものとしては, Mann and Shapley [9] においてモンテカルロ法が提案されている. モンテカルロ法に関するほかの論文としては文献 [10–14] などがある. ほかの手法としては, 多重線形展開 [15, 16], 動的計画法 [14, 17–20], 母関数 [21], 二分決定図 (BDD) [22], カルノー図 [23], 関係度数 [24], 列挙法 [25] などを用いた方法が研究されている. 投票力指数の計算法に関する概説論文として文献 [14] がある.

次節では, 本稿で用いる概念と記号を定義する. 3 節では, 単純なモンテカルロ法 (Algorithm A1) を記述し, 論文 [10] の解析を拡充する. 4 節では, 新たなモンテカルロ法 (Algorithm A2) を提案し, 単純なモンテカルロ法に比べ必要サンプル数が少ないことを示す. 二つのモンテカルロ法の比較をまとめたものを表 1 に示す. ただし, $(\varphi_1, \varphi_2, \dots, \varphi_n)$ は (真の) S-S 指数, $(\varphi_1^A, \varphi_2^A, \dots, \varphi_n^A)$ は Algorithm A1 または A2 によって得られる値を表している.

本稿では, 紙面の都合で証明をすべて省略しているため, 詳細は文献 [26] を確認されたい.

2. 記号と定義

本稿では, 協力ゲームの一種である 重み付き多数決ゲームを扱う. プレイヤーの集合を $N = \{1, 2, \dots, n\}$ とし, プレイヤーの部分集合を *提携* と呼ぶ. 重み付き多数決ゲーム G は, 正整数の列 $G = [q; w_1, w_2, \dots, w_n]$ で定義される. ここで w_i はプレイヤー i のもつ票数を表し, q は投票に勝つために必要な票数 (*必要票数* と呼ぶ) を表す. 本稿では不等式 $0 < q \leq w_1 + w_2 + \dots + w_n$ が成り立つことを仮定する. 提携 $S \subseteq N$ が $q \leq \sum_{i \in S} w_i$ を満たすとき, S は *勝利提携* であるという. 仮定 $q \leq w_1 + w_2 + \dots + w_n$ より, N (全体提携) は勝利提携である. 勝利提携でない提携は, *敗北提携* と呼ぶ. 空集合に対応する提携は敗北提携であるとする.

うしおだ ゆうと
株式会社リクルート
たなか まさと, まつい ともみ
東京工業大学工学院経営工学系
〒152-8552 東京都目黒区大岡山 2-12-1
matsui.t.af@m.titech.ac.jp

表 1 必要なサンプル数の比較表

要求する性質	必要なサンプル数	
	Algorithm A1	Algorithm A2
$\Pr \left[\left \varphi_i^A - \varphi_i \right < \varepsilon \right] \geq 1 - \delta$	$\frac{\ln 2 + \ln(1/\delta)}{2\varepsilon^2}$ (Bachrach et al. [10])	$\frac{\ln 2 + \ln(1/\delta)}{2\varepsilon^2} \left(\frac{1}{i^2} \right)$ ($w_1 \geq \dots \geq w_n$ を仮定)
$\Pr \left[\forall i \in N, \left \varphi_i^A - \varphi_i \right < \varepsilon \right] \geq 1 - \delta$	$\frac{\ln 2 + \ln(1/\delta) + \ln n}{2\varepsilon^2}$	$\frac{\ln 2 + \ln(1/\delta) + \ln 1.129}{2\varepsilon^2}$
$\Pr \left[\frac{1}{2} \sum_{i \in N} \left \varphi_i^A - \varphi_i \right < \varepsilon \right] \geq 1 - \delta$	$\frac{n \ln 2 + \ln(1/\delta)}{2\varepsilon^2}$	$\frac{n'' \ln 2 + \ln(1/\delta)}{2\varepsilon^2}$

ただし n'' は、票数が互い異なるプレイヤーの集合の最も大きなもののサイズ、すなわち票数の種類数である。

次に S-S 指数を定義する。プレイヤー集合 N 上に定義された順列 $\pi : \{1, 2, \dots, n\} \rightarrow N$ は、プレイヤーの全順序 $(\pi(1), \pi(2), \dots, \pi(n))$ を与えるとする。順列すべての集合を Π_N と書く。空集合が敗北提携で、 N が勝利提携であることから、与えられた順列に従ってプレイヤーが一人ずつ加わって提携が形成されるとすると、それまで敗北提携であった提携を勝利提携に変えるプレイヤーがただ一人存在する。このプレイヤーを、与えられた $\pi \in \Pi_N$ のピヴォットと呼び、 $\text{piv}(\pi) \in N$ と書く。すなわち、プレイヤー $\pi(i) \in N$ が順列 $\pi \in \Pi_N$ のピヴォットであるとは、 $\{\pi(1), \pi(2), \dots, \pi(i-1)\}$ が敗北提携であり、かつ、 $\{\pi(1), \pi(2), \dots, \pi(i-1), \pi(i)\}$ が勝利提携となることである。各プレイヤー $i \in N$ に対し、 i がピヴォットとなる順列を $\Pi_i = \{\pi \in \Pi_N \mid \text{piv}(\pi) = i\}$ と書く。明らかに $\{\Pi_1, \Pi_2, \dots, \Pi_n\}$ は Π_N の分割 (直和分解) となる。プレイヤー i の S-S 指数 φ_i は $\varphi_i = |\Pi_i|/n!$ と定義される。不等式 $0 \leq \varphi_i \leq 1$ ($\forall i \in N$) と等式 $\sum_{i \in N} \varphi_i = 1$ が成り立つことは明らかであろう。

本稿では、以下の仮定をおく。

仮定 1. プレイヤーの番号 (名前) は、 $w_1 \geq w_2 \geq \dots \geq w_n$ を満たすとする。

上記の仮定より $\varphi_1 \geq \varphi_2 \geq \dots \geq \varphi_n$ が成立することは、容易に示すことができる。

3. 簡単な解法とその解析

本節では、簡単なモンテカルロ法を記載し、その性能について述べる。下記において M は総サンプル数である。

Algorithm A1

Step 0: $m := 1, \varphi'_i := 0$ ($\forall i \in N$) とする。

Step 1: 順列 $\pi \in \Pi_N$ をランダムに選択する。

確率変数 $I^{(m)}$ を $\text{piv}(\pi)$ とし、 $\varphi'_{I^{(m)}}$ を $\varphi'_{I^{(m)}} + 1$ に更新する。

Step 2: もし $m = M$ ならば、 φ'_i/M ($\forall i \in N$) を出力して終了。

それ以外は $m := m + 1$ と更新し Step 1 へ戻る。

任意の順列 $\pi \in \Pi_N$ に対し、そのピヴォット $\text{piv}(\pi) \in N$ は $O(n)$ の計算量で見つけることができる。ゆえに、Algorithm A1 の時間計算量は $O(M(\tau(n) + n))$ である。ここで $\tau(n)$ は N 上の順列一つをランダムに生成する計算量である。

Algorithm A1 が出力する (確率変数) ベクトルを $(\varphi_1^{A1}, \varphi_2^{A1}, \dots, \varphi_n^{A1})$ と書く。以下が成り立つことは明らかであろう。

定理 1. $\forall i \in N, E[\varphi_i^{A1}] = \varphi_i$ 。

次の定理を用いると、Algorithm A1 を実行する際の必要サンプル数を算定できる。

定理 2. 任意の $\varepsilon > 0$ と $0 < \delta < 1$ に対し、以下が成り立つ。

(1) サンプル数を $M \geq \frac{\ln 2 + \ln(1/\delta)}{2\varepsilon^2}$ とすると、各プレイヤー $i \in N$ は以下を満たす [10] :

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

(2) サンプル数を $M \geq \frac{\ln 2 + \ln(1/\delta) + \ln n}{2\varepsilon^2}$ とすると、以下が成り立つ :

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

(3) サンプル数を $M \geq \frac{n \ln 2 + \ln(1/\delta)}{2\varepsilon^2}$ とすると、以下が成り立つ :

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

上記 (3) での値 $\frac{1}{2} \sum_{i \in N} \left| \varphi_i^{\text{A1}} - \varphi_i \right|$ は分布間距離の一種であり全変動距離 (total variation distance) と呼ばれる。

4. 提案解法

本節では、順列の分割 $\{\Pi_1, \Pi_2, \dots, \Pi_n\}$ 上の階層構造を利用した新たな解法の提案を行う。1 以外の各プレイヤー $i \in N \setminus \{1\}$ に対し、解法の鍵となる写像 $f_i : \Pi_i \rightarrow \Pi_N$ を導入する。任意の順列 $\pi \in \Pi_i$ に対し $f_i(\pi)$ は、列 $(\pi(1), \pi(2), \dots, \pi(n))$ 中のプレイヤー i と $i-1$ の場所を交換して得られる順列とする。仮定 1 より $w_{i-1} \geq w_i$ が成り立つことから、順列 $f_i(\pi)$ のピヴォットがプレイヤー $(i-1)$ になることが容易に示される。さらに、写像 f_i の定義より下記が成り立つ。

補題 3. 任意の $i \in N \setminus \{1\}$ に対し、写像 $f_i : \Pi_i \rightarrow \Pi_{i-1}$ は単射である。

図 1 に、重み付き多数決ゲーム $G = [50; 40, 30, 20, 10]$ から作られる単射 f_2, f_3, f_4 を示す。

順列の順序対 (π, π') が $\pi \in \Pi_i, \pi' \in \Pi_j, i \leq j$ と $\pi = f_{i-1} \circ \dots \circ f_{j-1} \circ f_j(\pi')$ を満たすとき、 π' は π の祖先であるという。また、任意の順列 π は自分自身 π の祖先であるとする。補題 3 より、任意の順列 $\pi \in \Pi_N$ は $[j = n \text{ または } f_{j+1}^{-1}(\pi') = \emptyset]$ を満たす唯一の祖先 $\pi' \in \Pi_j$ をもつ。上記の性質をもつ唯一の先祖を**始祖**と呼び、任意の順列 $\pi \in \Pi_N$ に対し、 π の始祖の順列のピヴォットを $\text{org}(\pi) \in N$ と書く、すなわち $\Pi_{\text{org}(\pi)}$ は π の始祖の順列を含む。

Π_1	f_2	Π_2	f_3	Π_3	f_4	Π_4
(2, ①, 3, 4)	←	(1, ②, 3, 4)	←	(1, ③, 2, 4)	←	(1, ④, 2, 3)
(2, ①, 4, 3)	←	(1, ②, 4, 3)	←	(1, ③, 4, 2)	←	(1, ④, 3, 2)
(4, 3, ①, 2)	←	(4, 3, ②, 1)	←	(4, 2, ③, 1)		
(3, 4, ①, 2)	←	(3, 4, ②, 1)	←	(2, 4, ③, 1)		
(3, ①, 4, 2)	←	(3, ②, 4, 1)	←	(2, ③, 4, 1)		
(3, ①, 2, 4)	←	(3, ②, 1, 4)	←	(2, ③, 1, 4)		
(4, ①, 3, 2)						
(4, ①, 2, 3)						
(4, 2, ①, 3)						
(2, 4, ①, 3)						

図 1 重み付き多数決ゲーム $G = [50; 40, 30, 20, 10]$ から得られる単射 f_2, f_3, f_4
丸囲いの数字は、ピヴォットとなるプレイヤーである。

本節で提案する解法は以下のものである。

Algorithm A2

Step 0: $m := 1, \varphi'_i := 0 \ (\forall i \in N)$ とする。

Step 1: 順列 $\pi \in \Pi_N$ をランダムに選択する。確率変数 $L^{(m)}$ を $\text{org}(\pi)$ とし、

$$\varphi'_i := \begin{cases} \varphi'_i + 1/L^{(m)} & (\text{if } 1 \leq i \leq L^{(m)}), \\ \varphi'_i & (\text{if } L^{(m)} < i), \end{cases} \quad \text{と更新する.}$$

Step 2: もし $m = M$ ならば、 $\varphi'_i/M \ (\forall i \in N)$ を出力して終了。

それ以外は $m := m+1$ と更新し Step 1 へ戻る。
たとえば図 1 の例において、Algorithm A2 の Step 1 で順列 $\pi = (3, \textcircled{2}, 4, 1)$ を選択したならば、その始祖のピヴォットは $\text{org}(\pi) = 3$ であり、Algorithm A2 は以下の更新

$$\begin{aligned} &(\varphi'_1, \varphi'_2, \varphi'_3, \varphi'_4) \\ &:= (\varphi'_1 + (1/3), \varphi'_2 + (1/3), \varphi'_3 + (1/3), \varphi'_4) \end{aligned}$$

を行う。任意の順列 $\pi \in \Pi_N$ に対し、その始祖のピヴォット $\text{org}(\pi) \in N$ を見つける時間計算量は $O(n)$ である。ゆえに Algorithm A2 の時間計算量は Algorithm A1 と同じ $O(M(\tau(n) + n))$ となる。

Algorithm A2 が出力する (確率変数) ベクトルを $(\varphi_1^{\text{A2}}, \varphi_2^{\text{A2}}, \dots, \varphi_n^{\text{A2}})$ と書く。以下が成り立つことは明らかであろう。

定理 4. (1) 任意の $i \in N$ に対し、 $E[\varphi_i^{\text{A2}}] = \varphi_i$.
(2) 任意のプレイヤー対 $\{i, j\} \subseteq N$ に対し、 $\varphi_i > \varphi_j$ ならば $\varphi_i^{\text{A2}} \geq \varphi_j^{\text{A2}}$ が成り立つ。
(3) 任意のプレイヤー対 $\{i, j\} \subseteq N$ に対し、 $\varphi_i = \varphi_j$ ならば $\varphi_i^{\text{A2}} = \varphi_j^{\text{A2}}$ が成り立つ。

次の定理を用いると、Algorithm A2 を実行する際の必要サンプル数を算定できる。

定理 5. 任意の $\varepsilon > 0$ と $0 < \delta < 1$ に対し、以下が成り立つ。

(1) プレイヤー $i \in N = \{1, 2, \dots, n\}$ に対し、サンプル数を $M \geq \frac{\ln 2 + \ln(1/\delta)}{2\varepsilon^2 i^2}$ とすると、以下が成り立つ：

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

(2) サンプル数を $M \geq \frac{\ln 2 + \ln(1/\delta)}{2\varepsilon^2}$ とすると、以下が成り立つ：

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

$$= 1 - 2 \left(\left(\frac{\delta}{2} \right) + \left(\frac{\delta}{2} \right)^4 + \left(\frac{\delta}{2} \right)^9 + \cdots + \left(\frac{\delta}{2} \right)^{n^2} \right).$$

(3) サンプル数を $M \geq \frac{|N^*| \ln 2 + \ln(1/\delta)}{2\varepsilon^2}$ とすると、以下が成り立つ：

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

ただし $N^* = \{i \in N \setminus \{n\} \mid \varphi_i > \varphi_{i+1}\} \cup \{n\}$ である、すなわち $|N^*|$ は、S-S 指数が互い異なるプレイヤーの集合の最も大きなもののサイズである。

定理 5 (2) の結果を簡略化したものが、以下の系である。この系より、必要なサンプル数 M がプレイヤー数 n に依存しないことがわかる。

系 6. 任意の $\varepsilon > 0$ と $0 < \delta' < 1$ に対し、サンプル数を $M \geq \frac{\ln 2 + \ln(1/\delta') + \ln 1.129}{2\varepsilon^2}$ とすると、以下が成り立つ：

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

上記の定数は次のような値 $\ln 2 \simeq 0.69314$, $\ln 1.129 \simeq 0.12133$ である。定理 5 (3) では集合 N^* のサイズが必要となるが、プレイヤーの対称性を確認する問題が NP-完全であることから [7]、現実の状況では N^* のサイズの算定は困難である。性質「 $\varphi_i > \varphi_{i+1}$ ならば $w_i > w_{i+1}$ である」が成り立つことより、 N^* のサイズは、票数が互い異なるプレイヤーの集合の最も大きなもののサイズ、すなわち票数の種類数（これを n'' と記す）以下である。ゆえに Algorithm A2 の実際の

実行においては、 $|N^*|$ の代替として、票数の種類数である n'' を用いることができる。二つの解法の必要サンプル数を表 1 にまとめる。

5. 計算実験

本節では、実際の例についての計算例を示す。EU 議会における各国をプレイヤーとした投票ゲーム（の一つ）は、

[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, 3]

と書くことができる [27, 28]。このゲームのプレイヤー数は $n = 27$ であり、票数の種類数は $n'' = 9$ となる。Owen [29](Section 12.4) において提示された、アメリカ合衆国の投票ゲームは

[270; 45, 41, 27, 26, 26, 25, 21, 17, 17, 14, 13, 13, 12, 12, 12, 11, 10, ..., 10, 9, ..., 9, 8, 8, 7, ..., 7, 6, ..., 6, 5, 4, ..., 4, 3, ..., 3]

4 人 4 人
4 人 4 人 9 人 7 人

と書くことができる。このゲームのプレイヤー数は $n = 51$ であり、票数の種類数は $n'' = 19$ となる。

計算実験における環境は次の通りである；i7-7700 CPU@3.6 GHz Memory (RAM) 16 GB. Algorithms A1 と A2 の実装には Python 3.6.5 を用いた。

上記の EU 議会とアメリカ合衆国の 2 例に対し、Algorithms A1 と A2 におけるサンプル数 M は、 $M \in \{1 \times 10^5, 2 \times 10^5, \dots, 24 \times 10^5\}$ と設定した。各 M に対し Algorithms A1 と A2 を 100 回実行した。プレイヤー i について、サンプル数 M を固定して 100 回実行した際の、値 $|\varphi_i - \varphi_i^A|$ の平均

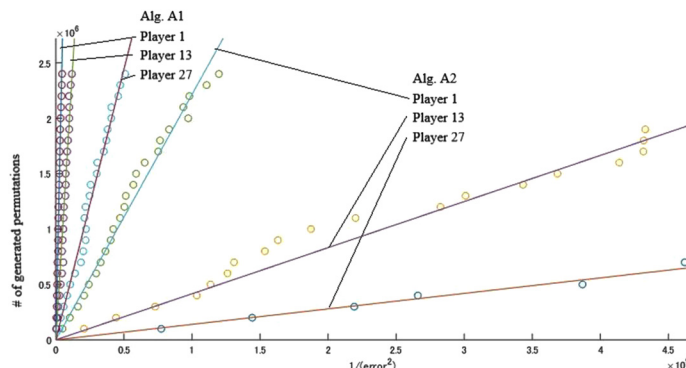


図 2 EU 議会

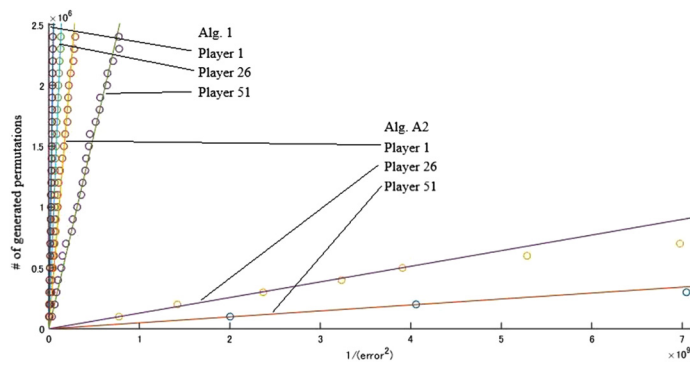


図 3 アメリカ合衆国

表 2 Algorithms A1 と A2 の比較

EU 議会	α		比
	Alg. A1	Alg. A2	
プレイヤー 1	0.0557	0.0022	25.318
プレイヤー 13	0.0199	4.1615×10^{-4}	47.819
プレイヤー 27	0.0049	1.3987×10^{-4}	35.033

アメリカ合衆国	α		ratio
	Alg. A1	Alg. A2	
プレイヤー 1	0.0489	0.0181	2.7017
プレイヤー 26	0.0088	1.2837×10^{-4}	68.552
プレイヤー 51	0.0032	4.8911×10^{-5}	65.424

値を $\hat{\varepsilon}_i$ とする. 図 2 と図 3 の横軸は, $1/\hat{\varepsilon}_i^2$ を表している. 表 2 において, 式 $M = \alpha/\hat{\varepsilon}_i^2$ が成り立つとの仮定の下で, 定数 α の値を最小二乗法で求めた結果と, 二つの解法における定数 α の比を記載した.

生成順列一つに対し, Algorithms A1 と A2 の計算量はどちらも $O(n)$ である. この $O(n)$ の計算量の定数部分について確認の実験を行った. 実験では票数 w_i は, 区間 $[1, 10]$ または $[1, 20]$ から一様ランダムに抽出し, 必要票数は $(1/2) \sum_{i \in N} w_i$ とした. プレイヤー数を $n \in \{10, 20, \dots, 100\}$, サンプル数 $M = 10,000$ として, Algorithms A1 と A2 を実行した. 計算時間が $an + b$ となるとの仮定の下で, a と b の値を最小二乗法で求めた結果を図 4 にまとめた. 実験の結果より, 生成順列一つに対する Algorithm A2 の計算量は, Algorithm A1 の 5 倍程度であることがわかる.

6. まとめ

本稿では, 重み付き多数決ゲームの S-S 指数を求める単純なモンテカルロ法 (Algorithm A1) に対し, 論文 [10] における解析を拡充した. 特に, 全変動距離に関

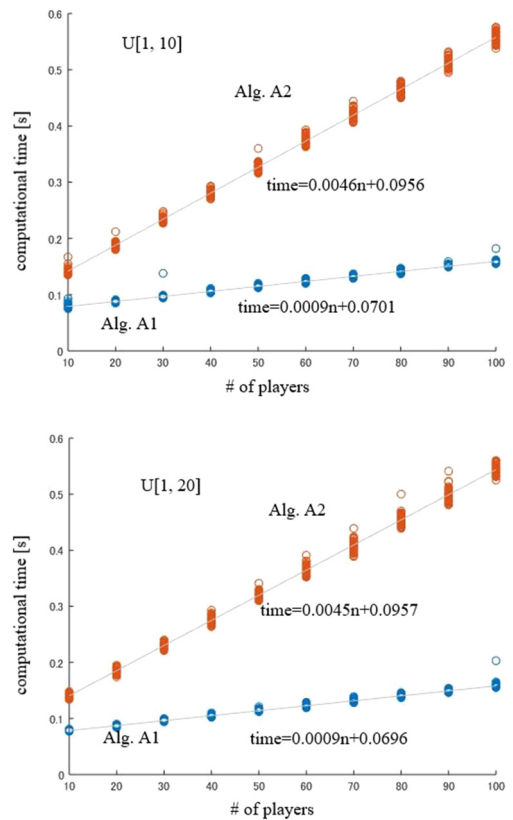


図 4 計算時間

する解析を新たに追加した. 本稿では証明を記載していないが, この解析において Bretagnolle-Huber-Carol 不等式 [30] と呼ばれる不等式を用いている.

また, 新たなモンテカルロ法 (Algorithm A2) を提案し, 解析を行った. 提案解法で得られる指数 $(\varphi_1^{A2}, \varphi_2^{A2}, \dots, \varphi_n^{A2})$ は, $[\varphi_i < \varphi_j \text{ ならば } \varphi_i^{A2} \leq \varphi_j^{A2}]$ かつ $[\varphi_i = \varphi_j \text{ ならば } \varphi_i^{A2} = \varphi_j^{A2}]$ という性質をもつ点が, 単純なモンテカルロ法 (Algorithm A1) との大きな違いである.

参考文献

- [1] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 1944.
- [2] L. S. Shapley and M. Shubik, "An algorithm for evaluating the distribution of power in a committee system," *American Political Science Review*, **48**, pp. 787–792, 1954.
- [3] L. S. Shapley, "A value for n -person games," *Contributions to the Theory of Games II*, H. W. Kuhn and A. W. Tucker (eds.), Princeton University Press, pp. 307–317, 1953.
- [4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, WH Freeman, 1979.
- [5] X. Deng and C. H. Papadimitriou, "On the complexity of cooperative solution concepts," *Mathematics of Operations Research*, **19**, pp. 257–266, 1994.
- [6] K. Prasad and J. S. Kelly, "NP-completeness of some problems concerning voting games," *International Journal of Game Theory*, **19**, pp. 1–9, 1990.
- [7] Y. Matsui and T. Matsui, "NP-completeness for calculating power indices of weighted majority games," *Theoretical Computer Science*, **263**, pp. 305–310, 2001.
- [8] E. Elkind, L. A. Goldberg, P. Goldberg and M. Wooldridge, "Computational complexity of weighted threshold games," In *Proceeding of the National Conference on Artificial Intelligence*, AAAI Press, pp. 718–723, 2007.
- [9] I. Mann and L. S. Shapley, "Values of large games. IV: Evaluating the electoral college by Monte Carlo techniques," *Technical Report, The RAND Corporation*, RM-2651, 1960.
- [10] Y. Bachrach, E. Markakis, E. Resnick, A. D. Procaccia, J. S. Rosenschein and A. Saberi, "Approximating power indices: Theoretical and empirical analysis," *Autonomous Agents and Multi-Agent Systems*, **20**, pp. 105–122, 2010.
- [11] J. Castro, D. Gómez and J. Tejada, "Polynomial calculation of the Shapley value based on sampling," *Computers & Operations Research*, **36**, pp. 1726–1730, 2009.
- [12] J. Castro, D. Gómez, E. Molina and J. Tejada, "Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation," *Computers & Operations Research*, **82**, pp. 180–188, 2017.
- [13] S. S. Fatima, M. Wooldridge and N. R. Jennings, "A linear approximation method for the Shapley value," *Artificial Intelligence*, **172**, pp. 1673–1699, 2008.
- [14] T. Matsui and Y. Matsui, "A survey of algorithms for calculating power indices of weighted majority games," *Journal of the Operations Research Society of Japan*, **43**, pp. 71–86, 2000.
- [15] D. Leech, "Computing power indices for large voting games," *Management Science*, **49**, pp. 831–837, 2003.
- [16] G. Owen, "Multilinear extensions of games," *Management Science*, **18**, pp. 64–79, 1972.
- [17] S. J. Brams and P. J. Affuso, "Power and size: A new paradox," *Mimeographed Paper*, 1975.
- [18] W. F. Lucas, "Measuring power in weighted voting systems," *Political and Related Models*, S. J. Brams, W. F. Lucas and P. D. Straffin (eds.), Springer, pp. 183–238, 1983.
- [19] 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.
- [20] T. Uno, "Efficient computation of power indices for weighted majority games," In *Proceeding of International Symposium on Algorithms and Computation (ISAAC)*, LNCS, **7676**, Springer, pp. 679–689, 2012.
- [21] J. M. Bilbao, J. R. Fernandez, A. J. Losada and J. J. Lopez, "Generating functions for computing power indices efficiently," *Top*, **8**, pp. 191–213, 2000.
- [22] S. Bolus, "Power indices of simple games and vector-weighted majority games by means of binary decision diagrams," *European Journal of Operational Research*, **210**, pp. 258–272, 2011.
- [23] A. M. A. Rushdi and O. M. Ba-Rukab, "Map calculation of the Shapley-Shubik voting powers: An example of the European Economic Community," *International Journal of Mathematical, Engineering and Management Sciences (IJMEMS)*, **2**, pp. 17–29, 2017.
- [24] R. Berghammer, S. Bolus, A. Rusinowska and H. De Swart, "A relation-algebraic approach to simple games," *European Journal of Operational Research*, **210**, pp. 68–80, 2011.
- [25] B. Klinz and G. J. Woeginger, "Faster algorithms for computing power indices in weighted voting games," *Mathematical Social Sciences*, **49**, pp. 111–116, 2005.
- [26] Y. Ushioda, M. Tanaka and T. Matsui, "Monte Carlo methods for calculating Shapley-Shubik power index in weighted majority games," *arXiv preprint*, arXiv:2101.02841, 2021.
- [27] J. M. Bilbao, J. R. Fernandez, N. Jiménez and J. J. Lopez, "Voting power in the European Union enlargement," *European Journal of Operational Research*, **143**, pp. 181–196, 2002.
- [28] D. S. Felsenthal and M. Machover, "The treaty of nice and qualified majority voting," *Social Choice and Welfare*, **18**, pp. 431–464, 2001.
- [29] G. Owen, *Game Theory*, Academic Press, 1995.
- [30] A. W. van der Vaart and J. A. Wellner, *Weak Convergence and Empirical Processes: With Applications to Statistics*, Springer, 1989.