重み付き投票ゲームにおける投票力指数の計算の高速化

宇野 毅明
国立情報学研究所, 〒 101-8430 東京都千代田区一ツ橋 2-1-2, e-mail: uno@nii.ac.jp

抄録: 議会においては, 各党が保有する議席数, 法案決定に必要な賛成の票数によって, 各党の発言力は変化する. その発言力を数理的なモデルを用いて表現したものが投票力指数である. いくつかのモデルが提案されており, それぞれについて, その指数を計算するアルゴリズムが研究されている. 本稿では, これらのアルゴリズムの改良方法を示し, 1 つの党の指数計算と同じ計算量で, すべての党の指数を計算するアルゴリズムを提案する.

Fast Computation of Power Indices on Weighted Majority Games

Takeaki UNO
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, JAPAN, e-mail: uno@nii.ac.jp

Abstract: In a council, the power of parties changes as the number of persons in the parties, and the number of votes to win. Power indices are indices representing power of parties in the mathematical way by using mathematical models. There proposed several models, and algorithms for computing indices for these models. In this paper, we propose an improvement for these algorithms, and propose improved algorithms for computing indices of all parties running in the time for computing indices of constant number of parties.

1 導入

ある議会で, 法案成立に必要な賛成票数が q であるとする. 議会に党 $p_1, ..., p_n$ があり, 議席をそれぞれ $w(p_1), ..., w(p_n) < q$ 持つとする. 各党は, 自身の法案を成立させるためには, 必ず他党と協力して合計議席 q 以上のグループを作らなければならないとする. これを重み付き投票ゲームと呼ぶ. ゲームの用語で, 党をプレイヤー, 議席を重みといい, 以下ではそれらの用語を使用する. この状況下で, 各党の力を数理的なモデルを用いて指数化したものが投票力指数である.

モデルの違いにより, いくつかの指数が提案されている. その中でもバンザフ (Banzhaf) 指数 [1], シャープレイ・シュービック (Sharpley-Shubik) 指数 [7], ディーガン・パックル (Deegan-Packle) 指数 [3] が有名であり, その性質, 計算を行うアルゴリズムとともに研究されている. 各指数のモデル・数学的な性質などについては参考文献を参照されたい [5, 6].

これら指数の計算方法はいくつか提案されている. 一つ目は, 単純にすべての組合わせを虱潰しに調べて計算する方法. 計算時間は $\mathrm{O}(n2^n)$ となる. もう一つは動的計画法を用いた方法である. 計算時間は, バンザフ指数は $\mathrm{O}(qn)$ [4, 2], シャープレイ・シュービック指数は $\mathrm{O}(qn^2)$ [4, 2], ディーガン・パックル指数は $\mathrm{O}(qn^3)$ [5] となる. いずれも 1 プレイヤーの指数の計算時間である. 全員分の計算には, この n 倍程度の時間がかかる.

近年のコンピューターは 1 時間に 3600 億回程度の演算が行える．よって, 指数時間アルゴリズムは $n=30$ となると実用的 (1 時間程度) では解けなくなる．動的計画で全員分の指数を計算する場合も, qn^2, qn^3, qn^4 がこの 3600 億よりも大きくなると, 現実的な時間では解けない可能性が出てくる．たとえばアメリカの大統領選挙の場合, プレイヤー数 n は 50, q は 1 億程度であるので, $qn^3 = 625,000$ 億となり, とても解けそうにない．通常このような問題は, 対話型, あるいは何かの問題の子問題として解かれる場合が多いので, 前者なら 5 秒程度, 後者ならば 1/100 秒程度で解かないと実用的とはならない．まだまだ高速化の必要性がある．

本稿では, 動的計画法のアルゴリズムを改良し, 全プレイヤーの指数を 1 プレイヤーの計算と同じ計算量で計算するアルゴリズムを提案する．ただし, ディーガン・パックル指数については, 1 プレイヤーを $\mathrm{O}(n^2q)$ で計算する方法を考え, それを元に高速化しているので, 全プレイヤーの計算も $\mathrm{O}(n^2q)$ 時間となっている．およそプレイヤー 3 人分の計算時間で全員分の指数計算が終了するので, たとえば $n=50$ であれば, およそ 16 倍程度の高速化になる．当然, n が大きくなればそれだけ速度比率は大きくなる．

これらアルゴリズムは, 以下の節で解説する．次節で用語・記法の定義を行う．3 節で最も単純なバンザフ指数に対するアルゴリズムを解説する．ここで, どの指数計算アルゴリズムにも共通の改良方法を述べる．その後の節で, シャープレイ・シュービック指数, ディーガン・パックル指数に対するアルゴリズムの解説を行う．

2　記法

$p_1, ..., p_n$ をプレイヤーとする．n はプレイヤーの数である．$w(p)$ をプレイヤー p の重み (票数) とする．また, q を法案を通すために必要な票数とする．プレイヤーの部分集合を**提携**という．提携 S に対して, $w(S) = \sum_{p \in S} w(p)$ とする．$w(\emptyset) = 0$ とする．$w(S) \geq q$ ならば, S は**勝利提携**と呼ばれ, 逆に $w(S) < q$ である場合には, S は**敗北提携**と呼ばれる．

P_i をプレイヤー 1 からプレイヤー i までの提携, すなわち $P_i = \{p_1, ..., p_i\}$ とする．同様に $\bar{P}_i = \{p_i, ..., p_n\}$ とする．特に, $i < 1$ に対して, $P_i = \emptyset$, $i > n$ に対して $\bar{P}_i = \emptyset$ とする．

3　バンザフ指数の計算

プレイヤー p_i を含む勝利提携から p_i が抜けると敗北提携になるとする．すると, この提携で p_i は発言力があると考えられる．今, すべての提携が等確率で発生するとすると, p_i が発言力を持つ確率は

$$| \{S | S \subseteq \{p_1, ..., p_n\}, p_i \in S, w(S) \geq q, w(S \setminus \{p_i\}) < q\} | \ \ / \ \ 2^n$$

となる．これを p_i のバンザフ指数 ($Bz(p_i)$) と定義する [1].

まず, プレイヤー p_n のバンザフ指数 $Bz(p_n)$ を計算する既存の方法を紹介しよう [4, 2]．最初に, $0 \leq i \leq n$, $0 \leq y \leq q-1$ に対して以下の関数 f を定義する．

$$f(p_i, y) = \begin{cases} 1 & \text{if} \quad i = 0 \quad \text{and} \quad y = 0 \\ 0 & \text{if} \quad i = 0 \quad \text{and} \quad y > 0 \\ | \{S | S \subseteq P_i, w(S) = y\} | & \text{otherwise} \end{cases}$$

f は引数として p_0 を取れるものとし, その場合は $f(p_0, 0) = 1$, $f(p_0, y) = 0 (y > 0)$ となる．

$Bz(p_n) \times 2^n$ は p_n を含まない提携 S (つまり $S \subseteq P_{n-1}$) で $q - w(p_n) \le w(S) \le q-1$ を満たすものの数と等しいので,

$$Bz(p_n) \times 2^n = |\ \{S | S \subseteq P_{n-1}\}, q - w(p_n) \le w(S) \le q-1\}\ |$$

が成り立つ. $q - w(p_n) \le y \le q-1$ である y に対して, $w(S) = y$ となる $S \subseteq P_{n-1}$ の個数は $f(p_{n-1}, y)$ である. よって,

$$Bz(p_n) \times 2^n = \sum_{y=q-w(p_n)}^{q-1} f(p_{n-1}, y)$$

となる. よって, p_{n-1} に対し, すべての y について f の値が求まれば, $Bz(p_n)$ が計算できる. f の計算は以下の再帰式を用いて行う.

性質 1 $1 \le i \le n$, $0 \le y \le q-1$ に対して, 以下の再帰式が成り立つ.

$$f(p_i, y) = \begin{cases} f(p_{i-1}, y) + f(p_{i-1}, y - w(p_i)) & \textit{if} \quad w(p_i) \le y \le q-1 \\ f(p_{i-1}, y) & \textit{if} \quad 0 \le y < w(p_i) \end{cases}$$

Proof: もし $0 \le y < w(p_i)$ が成り立つならば, 任意の $S \subseteq P_i$ に対して, $w(S) = y$ ならば $p_i \notin S$ である. よって, $f(p_i, y) = f(p_{i-1}, y)$ である. もし $w(p_i) \le y \le q-1$ であるならば,

$$\begin{aligned} f(p_i, y) &= |\ \{S | S \subseteq P_i, w(S) = y\}\ | \\ &= |\ \{S | S \subseteq P_i, w(S) = y, p_i \notin S\}\ | + |\ \{S | S \subseteq P_i, w(S) = y, p_i \in S\}\ | \\ &= |\ \{S | S \subseteq P_{i-1}, w(S) = y\}\ | + |\ \{S \cup \{p_i\}\ | S \subseteq P_{i-1}, w(S) = y - w(p_i)\}\ | \\ &= f(p_{i-1}, y) + f(p_{i-1}, y - w(p_i)) \end{aligned}$$

である. $i = 1$ のときは $P_{i-1} = \emptyset$ であるので, $|\ \{S | S \subseteq P_{i-1}, w(S) = y\}\ | = f(p_0, y)$ であることを注意しておく. ∎

この補題から, p_i に関して f の値を計算すれば, p_{i+1} の f の値を $\mathrm{O}(q)$ で計算できる. よって, 動的計画法で p_{n-1} の f の値を $\mathrm{O}(nq)$ 時間で求められる. 他のプレイヤーについてバンザフ指数を求めるには, プレイヤーの添え字を付け替えて, これと同じ操作を $n-1$ 回行えばよい. 計算時間は $\mathrm{O}(n^2 q)$ となる. これが [4, 2]. のアルゴリズムの概略である.

この動的計画法は, 1 プレイヤー分の計算, すなわち動的計画法の部分に関しては, 無駄に思える計算は少なく, 計算量を減少させることは困難であろうと推測される. しかし, 全員分の計算を行うとなると, 同じような動的計画法を n 回も解くことになり, この点に関しては効率化が望めそうである. そこで, 添え字をつけ直す, という操作を行わずに $S_1 \subseteq P_{i-1}, S_2 \subseteq \bar{P}_{i+1}$ を用いて, 提携 S の分割 $S = S_1 \cup S_2$ を考えよう. すると, $q - w(p_i) \le w(S) \le q-1$ を満たす S の個数は $q - w(p_i) \le w(S_1) + w(S_2) \le q-1$ を満たすような S_1 と S_2 の組の個数と等しくなる. ゆえに, 任意の $z, 0 \le z \le q-1$ に対して, P_{i-1} の部分集合で重みが z になるものの数と, $\bar{P}_{i+1}$ の部分集合で重みが z になるものの数がわかっていれば, 動的計画法を解き直さなくても, 両者の積を取り, $Bz(p_i)$ を計算できる.

P_{i-1} の部分集合で重みが z になるものの数は $f(p_{i-1}, z)$ である. 同様に, $\bar{P}_{i+1}$ の部分集合で重みが z になるものの数を $g(p_{i+1}, z)$ とし, $1 \le i \le n+1$, $0 \le y \le q-1$ に対して以下のように

定義する．f と同じく, g は p_{n+1} を引数として取れるとする．

$$g(p_i, y) = \begin{cases} 1 & \text{if} \quad i = n+1 \quad \text{and} \quad y = 0 \\ 0 & \text{if} \quad i = n+1 \quad \text{and} \quad y > 0 \\ \mid \{S \mid S \subseteq \bar{P}_i, w(S) = y\} \mid & \text{otherwise} \end{cases}$$

これは, プレイヤーの添え字を逆順につけなおした場合の f と同じである．g の性質は基本的には f と同じであり, 動的計画法により $\mathrm{O}(nq)$ 時間で求めるられる．f と g を用いると,

$$\begin{aligned} Bz(p_i) \times 2^n &= \sum_{z=0}^{q-1} \sum_{y=\max\{0, q-w(p_i)-z\}}^{q-1-z} (f(p_{i-1}, z) \times g(p_{i+1}, y)) \\ &= \sum_{z=0}^{q-1} \left(f(p_{i-1}, z) \times \sum_{y=\max\{0, q-w(p_i)-z\}}^{q-1-z} g(p_{i+1}, y) \right) \end{aligned}$$

となる．この計算は, 素直に行うと $\mathrm{O}(q^2)$ 時間かかるので, 関数

$$h(p_i, y) = \sum_{z=0}^{y} g(p_i, z)$$

を用意する．これを用いると,

$$\sum_{y=\max\{q-z-w(p_i),0\}}^{q-z-1} g(p_{i+1}, y) = h(p_{i+1}, q-z-1) - h(p_{i+1}, \max\{q-z-w(p_i), 0\})$$

となるので, h が求まれば, 各プレイヤーのバンザフ指数は $\mathrm{O}(q)$ 時間で求められる．$h(p_i, y) = h(p_i, y-1) + g(p_i, y)$ であるので, p_i についての h の計算時間は $\mathrm{O}(q)$ となる．よって, 全員分のバンザフ指数の計算時間は $\mathrm{O}(nq)$ となる．

また, f の再帰式を変形することにより, p_{i+1} に関する f の値から p_i の f の値を計算できる．よって, 最初に p_{n-1} の f と p_{n+1} の g を計算して, p_n のバンザフを計算し, 以後 p_{n-2} の f と p_n の g, とバックワードで f と g の値を更新していけば, 各プレイヤーのバンザフ値を逐次降順で計算できる．そのとき, 同時に保持する必要のある f と g の値は高々 q 個であるので, 空間計算量は $\mathrm{O}(q)$ となる．

さらに, f, g は非ゼロとなる部分しか計算する必要はない, $y < q - w(P_i)$ が成り立つならば $h(p_{i+1}, q-y-1) - h(p_{i+1}, \max\{q-y-w(p_i), 0\}) = 0$ であるので, $f(p_{i-1}, y)$ を計算する必要はないことから, 計算量は減少しないが, 実用的には計算時間を減少できる．

以上の改良を行ったアルゴリズムを以下に示す．

Algorithm Banzaf_Index $(n, w())$

1) Compute $f(p_{n-1}, y)$ for $y = \max\{0, q-w(p_n)\}, ..., q-1$ with $f(p_{n-1}, y) > 0$

2) Set $g(p_{n+1}, y) = 1, i = n$

3) Compute $h(p_{i+1}, q-y-1)$ and $h(p_{i+1}, \max\{q-y-w(p_i), 0\})$
for $y = \max\{0, q-w(P_i)\}, ..., q-1$ with $f(p_{n-1}, y) > 0$

4) $B[i] = \displaystyle\sum_{y=\max\{0, q-w(P_i)\}, ..., q-1, f(p_{i-1}, y)>0} f(p_{i-1}, y) \times (h(p_{i+1}, q-y-1) - h(p_{i+1}, \max\{q-y-w(p_i), 0\}))$

5) **Output** Banzaf index of player $i = B[i]$
6) $i := i - 1$; **If** $i = 0$ **then End**
7) Compute $f(p_{i-1}, y)$ for $y \geq q - w(\bar{P}_i)\}$ with $f(p_i, y) > 0$ or $f(p_i, y + w(p_i)) > 0$
8) Compute $g(p_{i+1}, y)$ for $y \geq q - w(P_i)\}$ with $g(p_{i+2}, y) > 0$ or $g(p_{i+2}, y - w(p_i)) > 0$
9) **Go to** 3)

定理 1 プレイヤー $p_1, ..., p_n$, 各プレイヤーの重み, q が与えられたとき全プレイヤーのバンザフ指数の計算は $O(nq)$ 時間で行える. 空間計算量は $O(q)$ となる. ∎

4 シャープレイ・シュービック指数の計算

提携 S が最初空集合であり, プレイヤーが一人一人その提携に参加していく, という状況を考えよう. このとき, プレイヤー p_i が参加した際に, 提携が敗北提携から勝利提携に変わったとする. このとき, p_i が発言力を持つと考えよう. プレイヤーが提携に参加していく, そのしかたは $n!$ 通り存在する. 今, それらが等確率で発生するとすると, p_i が発言力を持つ確率は, π_n を 1 から n までの順列の集合として,

$$|\ \{(j_1, ..., j_n) | (j_1, ..., j_n) \in \pi_n, w(\{p_{j_1}, p_{j_2}, ..., p_i\}) \geq q, w(\{p_{j_1}, p_{j_2}, ..., p_i\}) - w(p_i) < q\}\ | / n!$$

となる. これが, p_i のシャープレイ・シュービック指数 $Ss(p_i)$ の定義である [7].

シャープレイ・シュービック指数に対しても, 動的計画法を用いたアルゴリズムが提案されている [4, 2]. このアルゴリズムも, バンザフ指数の場合と同様, 関数 f を定義し, それを動的計画法によって求め, 指数の計算を行う. 関数 f の定義はバンザフ指数の場合とは異なり, $0 \leq i \leq n$, $0 \leq y \leq q - 1, k \leq i$ に対して以下のようにする.

$$f(p_i, k, y) = \begin{cases} 1 & \text{if} \quad i = 0 \quad \text{and} \quad y = 0 \\ 0 & \text{if} \quad i = 0 \quad \text{and} \quad y > 0 \\ |\ \{S | S \subseteq P_i, w(S) = y, |S| = k\}\ | & \text{otherwise} \end{cases}$$

ここで, $Ss(p_n) \times n!$ を求める方法を考えよう. 順列 $(j_1, ..., j_n)$ で, $\{p_{j_1}, p_{j_2} ..., p_i\} = S$ となるようなものは $(|S| - 1)!(n - |S|)!$ 個ある. よって, $Ss(p_n) \times n!$ は, 提携 $S \subseteq P_{n-1}$ で, $q - w(p_n) \leq w(S) \leq q - 1$ を満たすものに関して $(|S| - 1)!(n - |S|)!$ の和をとったものになる. $q - w(p_n) \leq y \leq q - 1, 0 \leq k \leq n - 1$ なる y, k に対して, $w(S) = y, |S| = k$ となる $S \subseteq P_{n-1}$ の個数は $f(p_{n-1}, k, y)$ である. よって,

$$Ss(p_n) \times n! = \sum_{y=q-w(p_n)}^{q-1} \sum_{k=0}^{n-1} f(p_{n-1}, k, y)(k - 1)!(n - k)!$$

となる. p_{n-1} に関してすべての y, k について f の値が求まれば, $Ss(p_n)$ が計算できる. f の計算は, 以下の再帰式を用いた動的計画法で $\mathrm{O}(n^2 q)$ 時間で行う. 証明は性質 1 とほぼ同じであので, 省略する.

性質 2 $1 \leq i \leq n$ に対して, 以下の再帰式が成り立つ.

$$f(p_i, k, y) = \begin{cases} f(p_{i-1}, k, y) + f(p_{i-1}, k - 1, y - w(p_i)) & \mathit{if} \quad w(p_i) \leq y \leq q - 1, k > 0 \\ f(p_{i-1}, k, y) & \mathit{if} \quad 0 \leq y < w(p_i) \quad \mathit{or} \quad k = 0 \end{cases}$$ ∎

バンザフ指数と同様, すべてのプレイヤーについて $Ss(p_i)$ を計算するためには添え字を付け替えて動的計画法を解きなおし, $O(n^3q)$ 時間必要である.

このアルゴリズムも, 動的計画法に関する無駄な計算は少ないが, バンザフ指数と同様の方法で, 全員分の計算の効率化は望めそうである. しかし, シャープレイ・シュービック指数の場合, 一つ一つの計算に S の要素数が関係してくるので, バンザフ指数計算の改良と同じ方法で, f と同じ形の g を定義すると,

$$Ss(p_i)\times n! = \sum_{z=0}^{q-1}\sum_{k=0}^{i-1}\sum_{y=\max\{0,q-w(p_i)-z\}}^{q-1-z}\sum_{l=0}^{n-(i-1)} f(p_{i-1},k,z)\times g(p_{i+1},l,y)\times |k+l|!|n-k-l-1|!$$

となり, 1 プレイヤーの計算に $O(n^2q)$ 時間かかってしまい, これは計算量の減少になっていない. そこで g の定義を変更し, $1\le i\le n+1$, $0\le y\le q-1, k\le i-1$ なる i,y,k に対して,

$$g(p_i,k,y) = \begin{cases} k!(n-k-1)! & \text{if} \quad i=n+1 \quad \text{and} \quad y=0 \\ 0 & \text{if} \quad i=n+1 \quad \text{and} \quad y>0 \\ \displaystyle\sum_{S,S\subseteq \bar{P}_i, w(S)=y} (|S|+k)!(n-|S|-k-1)! & \text{otherwise} \end{cases}$$

とする. g に関しても, 以下の再帰式が成り立ち, 動的計画法で値を求められる. 証明は, 性質 1,2 とほぼ同じであるので, 省略する.

性質 3 $1\le i\le n, 0\le k\le i-1$ に対して,

$$g(p_i,k,y) = \begin{cases} g(p_{i+1},k,y)+g(p_{i+1},k+1,y-w(p_i)) & \textit{if} \quad w(p_i)\le y\le q-1 \\ g(p_{i+1},k,y) & \textit{if} \quad 0\le y< w(p_i) \end{cases} \blacksquare$$

$g(p_i,k,y)$ は, $S_1\subseteq P_{i-1}$ の要素数が k であるときに, $w(S_2)=y$ が成り立つような S_2 について, $(|S_1|+|S_2|)!(n-|S_1|-|S_2|-1)!$ の総和を取ったものである. すると, $w(S_1)=z, |S_1|=k, w(S_2)=y, q-w(p_i)\le z+y\le q-1$ を満たす S_1,S_2 の組すべてについて $(|S_1\cup S_2|-1)!(n-|S_1\cup S_2|)!$ の和をとったものは, $f(p_{i-1},k,z)\times g(p_{i+1},k,y)$ と等しくなる. よって,

$$\begin{aligned} Ss(p_i)\times n! &= \sum_{z=0}^{q-1}\sum_{k=0}^{i-1}\sum_{y=\max\{0,q-w(p_i)-z\}}^{q-1-z}(f(p_{i-1},k,z)\times g(p_{i+1},k,y)) \\ &= \sum_{z=0}^{q-1}\sum_{k=0}^{i-1}\left(f(p_{i-1},k,z)\times\sum_{y=\max\{0,q-w(p_i)-z\}}^{q-1-z} g(p_{i+1},k,y)\right) \end{aligned}$$

となる. 以上より, 3 節と同じ方法でアルゴリズムを構築し, 以下の定理を得る.

定理 2 プレイヤー $p_1,...,p_n$, 各プレイヤーの重み, q が与えられたとき全プレイヤーのシャープレイ・シュービック指数の計算は $O(n^2q)$ 時間で行える. 空間計算量は $O(nq)$ となる. $\blacksquare$

5 ディーガン・パックル指数の計算

勝利提携 S から任意のプレイヤーを取り除くと S が敗北提携となる場合, S は**極小勝利提携**であるという. 極小勝利提携に属するプレイヤーは, 発言力を持つと考えられる. ただし, 提携の

大きさが大きいと, 発言力は小さくなると考えられるので, 各プレイヤーは $1/|S|$ の発言力を持つとしよう. すべての極小勝利提携が等確率で発生すると仮定したときに, プレイヤー p_i の持つ発言力の期待値がディンガー・パックル指数の定義である [3].

一般性を失うことなく, この節では, 任意の $1 \leq i < j \leq n$ に対して $w(p_i) \geq w(p_j)$ を仮定する. そして, 提携 S に対して, $d(S)$ を S からもっとも添え字の大きいプレイヤーをのぞいて得られる提携とする. $S = \emptyset$ の場合, $d(S) = \emptyset$ とする. すると, S が極小勝利提携であることは,

$$w(S) \geq q, \quad w(d(S)) < q$$

であることと同値である. ゆえに, 極小勝利提携の集合を $\mathcal{X}$ とすれば, プレイヤー p_i のディーガン・パックル指数 $Dp(p_i)$ は以下のように定義される.

$$Dp(p_i) = (\sum_{S, S \subseteq \mathcal{X}, p_i \in S} \frac{1}{|S|})/|\mathcal{X}| = (\sum_{S, S \subseteq P_n, p_i \in S, w(S) \geq q, w(d(S)) < q} \frac{1}{|S|})/|\mathcal{X}|$$

ディーガン・パックル指数に関しても, [5] において動的計画法を用いたアルゴリズムが提案されている. 上記 2 指数のアルゴリズムと同様, 関数 f を定義して, それを動的計画法で計算し, 指数の計算を行うものである. このアルゴリズムでは, 提携 $S \subseteq P_i$ をその重み, 要素数, S の要素で最小重みを持つ要素がどれか, で分類を行っている. あるプレイヤーの指数を計算するために, f の値を最大で $\mathrm{O}(n^3 q)$ 個計算しなければならないので, 全員分の計算は $\mathrm{O}(n^4 q)$ 時間となる.

この節では, プレイヤーの添え字付けを工夫することにより, 分類が, 重みと要素数だけとなる動的計画法を使用している. 以下でそのアルゴリズムを解説を行うが, 今回は直接, 全員分の計算を高速に行うアルゴリズムを解説する.

任意の提携の組 $S_1 \subseteq P_{i-1},\ S_2 \subseteq P_i,\ p_i \in S_2$ に対して, $S = S_1 \cup S_2$ が $w(S) \geq q,\ w(d(S)) < q$ を満たすことは,

$$q \leq w(S_1 \cup S_2), \quad w(d(S_1 \cup S_2)) < q$$

と同値であり, さらに $p_i \in S_2$ であることから $S_2 \neq \emptyset$ であるので

$$q \leq w(S_1) + w(S_2), \quad w(d(S_2)) < q$$

とも同値である. よって, 任意の $0 \leq z \leq q-1$, $0 \leq k \leq i-1$ を満たす整数の組 z, k それぞれに対して

$$w(S_1) = z, \quad q - z \leq w(S_2), \quad w(d(S_2)) < q - z, \quad |S_1| = k$$

を満たす S_1, S_2 の組に関して $1/(k + |S_2|)$ の総和を取る. さらに, それらの総和を取ったもの, すなわち

$$\sum_{z=0}^{q-1} \sum_{k=0}^{i-1} (\sum_{(S_1, S_2), w(S_1) = z, q - z \leq w(S_2), w(d(S_2)) < q - z, |S_1| = k} \frac{1}{k + |S_2|}$$

は $Dp(p_i) \times |\mathcal{X}|$ と等しくなる.

$w(S_1) = z, |S_1| = k$ を満たす S_1 の数は, 前節で用いた $f(p_{i-1}, k, z)$ と同じである. また, $q - z \leq w(S_2),\ \ w(d(S_2)) < q - z$ を満たすような S_2 に関して $\frac{1}{k+|S_2|}$ の総和を取ったものを $g(p_i, k, q-z)$ とする. 正確には $1 \leq i \leq n+1$, $y \leq q-1, k \leq i-1$ に対して,

$$g(p_i, k, y) = \begin{cases} 1/k & \text{if} \quad y < 0 \\ 0 & \text{if} \quad i = n+1 \quad \text{and} \quad y \geq 0 \\ \displaystyle\sum_{S, S \subseteq \bar{P}_i, p_i \in S, w(S) \geq y, w(d(S)) < y} \frac{1}{|S| + k} & \text{otherwise} \end{cases}$$

とする. 前節同様, g について再帰式が成り立ち, 動的計画法を用いて $\mathrm{O}(n^2 q)$ 時間で求められる.

$$
\begin{aligned}
Dp(p_i) \times X &= \sum_{z=0}^{q-1} \sum_{k=0}^{i-1} (\sum_{(S_1,S_2),w(S_1)=z,q-z \leq w(S_2),w(d(S_2))<q-z,|S_1|=k} \frac{1}{k+|S_2|} \\
&= \sum_{z=0}^{q-w(p_i)} \sum_{k=0}^{i-1} (f(p_{i-1},k,y) \times g(p_i,k-1,q-z))
\end{aligned}
$$

が成り立つので, 3 節と同じ方法でアルゴリズムが構築できる.

定理 3 プレイヤー $p_1, ..., p_n$, 各プレイヤーの重み, q が与えられたとき全員のプレイヤーのディーガン・パックル指数の計算は $O(n^2 q)$ 時間で行える. 空間計算量は $O(qn)$ となる. ∎

6 まとめ

本稿では, 重み付き投票ゲームにおける 3 つの代表的な投票力指数バンザフ指数, シャープレイ・シュービック指数, ディーガン・パックル指数の計算を行う, 動的計画法を用いたアルゴリズムの高速化を行い, 全員分の指数の計算量が 1 プレイヤーの指数の計算量と等しくなった. 動的計画法の仕組みはそのままに, 動的計画法を前向きで解いたデータと後ろ向きで解いたデータを組み合わせることにより, 高速に全員分の計算ができる方法を考案した. ディーガン・パックル指数については, 1 プレイヤーの計算も高速化されたので, 全員分の計算はさらに高速になった.

参考文献

[1] J. F. Banzhaf III, *"Weighted Voting doesn't work,"* Rutgers Law Review, Vol. 19 pp. 317-343 (1965).

[2] S. J. Brams and P. J. Affuso, *"Power and size: a new paradox,"* mimeographed paper (1975).

[3] J. Deegan and E. W. Packel, *"A New Index of Power for Simple n-person Games,"* Internatinal Journal of Game Theory, Vol. 7, pp. 113-123 (1978).

[4] W.F.Lucas, *"Measuring Power in Weighted Voting Systems,"* in S. J. Brams, W. F. Lucas and P. D. Straffin Eds., *Political and related models,* Springer-Verlag, pp.183-238 (1983).

[5] 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, Vol. 43, No. 1, pp. 71-86 (2000).

[6] 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.

[7] L. S. Shapley and M. Shubik, *"A Method for Evaluating the Distribution of Power in a Committee System,"* American Political Science Review, Vol. 48, pp. 787-792 (1954).