種々の比例代表制アルゴリズムの比較について

山梨大学工学部 有沢 誠

The French never care what they do,
actually, as long as they pronounce it
properly. —— Profesor Higgins
My Fair Lady (by Alan Jay Lerner)

議席配分アルゴリズムが適切でさえあれば,
どの政党が何議席獲得したかには興味がない.
—— M.Arith

概要

比例代表制は, 按分比例の考えかたによって, 整数単位にものを配分する方法である. 複数のアルゴリズムを比較する場合には, そのための基準が必要になる. ここでは, 得票数単調性, クォータ充足性, 総議席数単調性の3種類の基準をもとにして, ドント法, ハミルトン法, クォータ法をはじめとする, 種々の比例代表制アルゴリズムを比較する. この中では, 軽微な欠陥はあるが, クォータ法が最も適切なアルゴリズムであると結論する.

1. はじめに

比例代表制は, 按分比例の考えかたによって, 整数単位にものを配分する方法である. 代議員定数の配分や, 政党別当選者の配分など, 比例代表制を適用すべき応用分野は多い. 比例代表制のアルゴリズムとしては, これまでに種々の方法が提案になっている. 現在日本の参議院比例代表区で採用しているドント法は, そうしたアルゴリズムの中のひとつに過ぎない.

一般にここで扱う問題は, 次の形にモデル化できる.

On Proportional Representative Assignment Algorithms Comparison, by Makoto Arisawa (Yamanashi University)

箱 1〜 n の容量が $x[1]$〜 $x[m]$ であるとき, 総数 w の玉を箱の容量に応じて配分せよ. ただし配分数 $y[1]$〜 $y[n]$ はすべて正整数または 0.

ここで, 箱が選挙区や政党に相当し, 容量が有権者数や得票数に相当する. 配分数の合計が配分すべき総数になることから,

$$y[1] + \ldots + y[n] = w$$

がなりたつ.

議員定数の配分では, どんなに有権者が少ない場合にも最低限1名の代議員定数を確保しなければならない. したがって w は n に等しいかそれより大きいことが必要条件になる. それに対して代議員を政党に配分する場合には, そうした制約はない. また $y[1]$〜 $y[n]$ がすべて2以上などの制約条件を課すこともありうる.

しかし配分の基本アルゴリズムは，そうした条件によらず本質的には同じである．容量と配分値の比を制約条件の下でできるだけ近い値にするように，すなわち，

$x[1]/\ y[1] \sim x[n]/\ y[n]$

ができるだけ近い値になるように，$y[1] \sim y[n]$を決めてやればよい．

以下では，$y[1] \sim y[n]$ に関する制約がない場合，すなわち政党に議席を配分する場合を前提として議論を進めていく．

一般に個々の容量 $x[1] \sim x[n]$ と総容量 z の比を配分比率とよぶ．すなわち，

$x[1] + \ldots + x[n] = z$

とすると，

$x[1]/\ z, \ldots, x[n]/\ z$

が配分比率であり，それぞれ 0～ 1 の間の小数になる．配分すべき総数に配分比率を掛けた値を厳密な配分数とよぶ．すなわち，

$w \times x[1]/\ z, \ldots, w \times x[n]/\ z$

が厳密な配分数である．この厳密な配分数がすべて整数の場合には，そのまま $y[1] \sim y[n]$ として採用すれば，容量と配分値の比はすべて等しくなる．しかし現実には整数にはならない．

小数部分の端数を切上げた値を厳密な配分数の上限値，小数部分の端数を切り捨てた値を厳密な配分数の下限値とよぶ．両者はたかだか 1 しか違わない．厳密な配分数は無理としても，その上限値か下限値かのどちらかに配分値を定めたい，と考えることはきわめて自然である．

以上の問題設定と個々のアルゴリズムで用いる記号を，以下に整理しておく。

n: 政党数 (integer);
i: 政党を表わす添字変数
　$= (1..n)$の要素 (integer);
$x[i]$: 政党別得票数 (integer array);
$z = \sum x[i]$: 総得票数 (integer);
$y[i]$: 政党別議席配分数 (integer array);
$w = \sum y[i]$: 総議席数 (integer);
$z/w = u$: 議席あたり得票数 (real);
$x[i]/z = p[i]$: 政党別得票比率 (real);
$x[i]/u = q[i]$: 政党別議席配分比率 (real);
$ceiling(q[i])$:政党別配分上限数 (integer);
$floor(q[i])$: 政党別配分下限数 (integer);
$z/(w+1) = u'$: 議席あたり疑似得票数 (real);
$x[i]/q' = q'[i]$: 政党別疑似議席配分数
　(real array);

2．アルゴリズムの比較基準

一般に複数のアルゴリズムを比較する場合には，そのための基準が必要になる．比例代表制アルゴリズムの場合には，次の 3条件を採用することが妥当だと考えられる．ただし「決めかたをどう決めるか」は，本質的に再帰構造であって，最終的にはどこかで，無限ループを断ち切る必要がある．そのことについては本稿ではふれない．

[1]　容量に対する単調性があること．すなわち，得票数が多い政党は少ない政党より議席が少なくなることはないこと．式を用いれば，任意の添字 i と j について，$x[i] \leq x[j]$ であれば $y[i] \leq y[j]$ となること．これを「得票数単調性」と呼ぶ．

[2]　厳密な配分数の上下限に収まること．これは，$y[1] \sim y[n]$ が厳密な配分数の上限または下限のどちらかに等しいことを意味する．これを「クォータ充足性」と呼ぶ．

[3]　定数増加に伴うパラドックスがないこと．これは，容量 $x[1] \sim x[n]$ を固定した状態で，総数 w を増加した場合に，$y[1] \sim y[n]$ は増加することはあっても減少することはないことを意味する．これを「総議席数単調性 (house monotone)」と呼ぶ．この反例を，アラバマパラドックスと呼ぶ．

実は 4番目の性質として，得票率単調性を考えることができる．これは，ある政党が前回の選挙に比べて今回の選挙で得票率を増加させた場合に，前回より議席が減少することはないという性質である．しかし，この性質を満足することが不可能なことは，スタインハウスの甲羅三角形 [Steinhaus 48]によって証明できる．したがって，この意味での選挙のパラドックスは，比例代表性アルゴリズムによらず，生じる可能性

がある．

ここにはいくつかの比例代表制アルゴリズムについての甲羅三角形の例を藤井等君のプログラムで作図した[有沢 81]，政党数が 3，議席数が 5の場合について描いたものを図 1にあげておく．ある政党が得票を増加させても，他の 2政党の得票増減によっては配分議席数が減少する可能性があることを，これらの図から読みとることができる．

いっぽう[2]のクォータ充足性は，配分数を厳密な割当数の上限下限の範囲に収めることで，できるだけ按分比例に近い形にすることを意味している．ただし以下に述べるクォータ法やハミルトン法など，この性質を満足する方法にも，ひとつ問題がある．

たとえば 10の議席をめぐって政党が 122あり，得票総数 10000票の割れかたが，次に示す通りだったとする．

$x[1] = 5900\ (59\%)$,

$x[2] = 2900\ (29\%)$,

$x[3] = x[4] = \cdots = x[122] = 10\ (0.1\%)$.

このとき，議席獲得数 $y[i] (1 \leq i \leq 122)$ は，次のようになる．

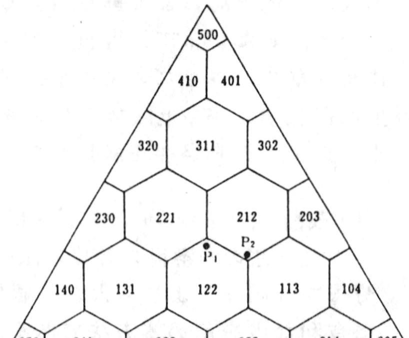
$y[1] = 6$,

$y[2] = 3$,

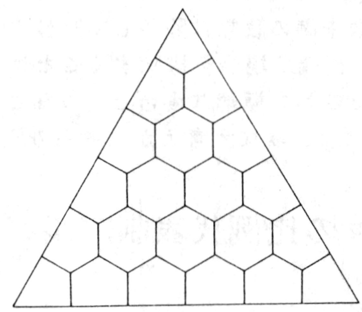
$y[3] = 1$,

$y[4] = y[5] = \cdots = y[122] = 0$.

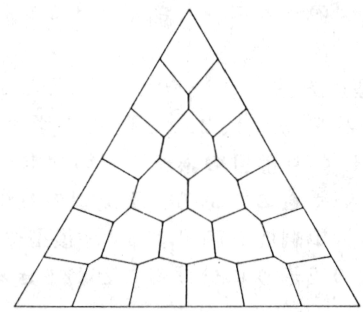
大政党の 1と 2がすでに上限値を得てしまった状態で，まだ 1議席残っている．このためどのアルゴリズムを用いるにせよ，残りの 1議席は 0.1どこかに与えることになる．ミニ政党はそれぞれ個々には 0.1けれども，120全部の支持を合計すると全体の 12ミニ政党全体として議席 1を確保している．



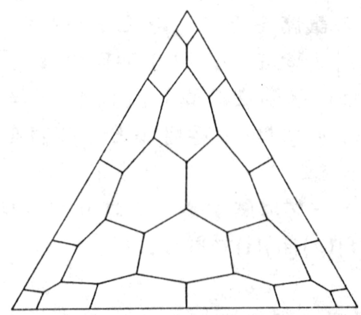
ヘア法の甲羅三角形(定数 5)



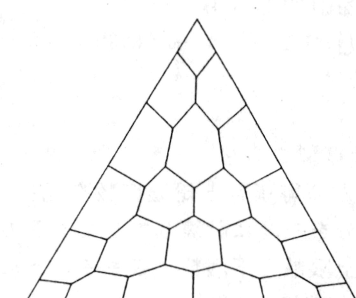
ドループ法の甲羅三角形



ドント法の甲羅三角形



改良ドント法（定率 1.3の配分）の甲羅三角形



改良ドント法（フィボナチ配分）の甲羅三角形

図 1　各種アルゴリズムの甲羅三角形

現実の選挙でこうした状況が万一生じても，規定得票数未満の政党は除外して計算するなどの方法で，極端な場合を排除することは可能である．またこれを極端な場合とはみなさず，少数意見の代表とみなす考えかたもありうる．

3．種々の比例代表制アルゴリズム

比例代表制アルゴリズムは，一回計算型のものと反復型のものに，大きく2分できる．

(1)反復型の方法

総議席数を $v = 1, 2, \ldots$ と順に1ずつ増加して，新しく v 番目の議席をどの政党に与えるか決めていくものである．反復型の方法は，必ず総議席数単調性を満足する．反復型の方法は，さらに次の2通りに分けることができる．

(1-1)既に決定した $y[i]$ に関するある基準で次の v 番目の議席をどこに与えるかを決定する．それを $y[k]$ とすると，$v+1$ 番目では $y[k]$ に関する部分だけ基準を計算し直して，$i \neq k$ となる $y[i]$ については，v 番目のときの値をそのまま使用できる．

このグループに属するアルゴリズムは，以下の(a)(b)(f)(g)(h)(i)である．

(1-2) v 番目の議席をどこに与えるか決定しても，$v+1$ 番目についてはすべての $y[i]$ について基準値を計算しなおす．

以下の(j)がこのグループに属するアルゴリズムである．

(2)一回計算型の方法

政党ごとの得票数と総議席数を元に，すべての議席配分を一度に計算してしまうものである．このとき上限数と下限数に収まるようにすれば，クォータを満足する．

このグループに属するアルゴリズムは，以下の(c)(d)(e)である．

本稿では，特に考慮すべきアルゴリズムとして，以下の10種類をとりあげる．それぞれがある方針で議席配分を行なうように設計してある．ただし政党に議席を配分するのでなく，選挙区に議員定数を配分することを前提とし，y[i]が正数(ゼロにならない)としたものも混じっている．

(a)ドント(d'Hondt)法

最も広く知られている方法で，別名も多い．Thomas Jefferson 法，最大除数法(method of greatest divisors:GD)，ハーゲンバッハビショフ法はすべてドント法と等価である．

政党ごとの得票数をその政党がこれまでに得た議席数+1で割った値，すなわち，

$$x[i]/(y[i]+1)$$

が最大な政党に次の議席を与える，という方法で計算する．仮の議席総数 v を 0 から 1 ずつ増加しながら，

$$x[i]/(y[i]+1) \leq x[j]/(y[j]+1)$$

で等号を満たす添字 j について，1議席を追加する．

あるいは，仮の議席総数 v を w から1ずつ増加しながら，各政党に $floor(q[i])$ ずつ議席を与える．配分議席総数がちょうど w になったところで反復を打ち切る．一般には $w \leq v$ で，通常は $w < v$ である．

このどちらの計算方法を採用しても，同じ議席配分結果になる．

参議院の比例代表区で採用になっているアルゴリズムがドント法である．このアルゴリズムは，それぞれの政党が最適な人数を立候補させて完全な票割を行ったという状況で，得票順に上位から配分総数まで割り当てることを意味している．したがって，容量の大きな箱(大政党)に有利なアルゴリズムである．ドント法は下限数は保証するが，上限数より大きい場合が生じる．クォータ充足性を満足しない．

(b) 改訂ドント法(サンラゲ法ほか)

ドント法の2番目の計算方法の途中の過程で，分母を $y[i]+1$ よりも大きい値にして，大政党に有利な点を薄める方法である．たとえばサンラゲ法は，$1, 2, 3, \ldots$ の代わりに，$1, 3, 5, \ldots$ で割

る方法である．ここの記法では，

$$x[i]/(y[i]*2+1)$$

の値が最大な政党に次の議席を与える．

また，票割が完全という条件をゆるめて，票割が不完全な場合を想定して当選者数を決める方法がいくつかある．統計的にみて，票割は調和級数に近い形になることが多い．そこで各政党の得票を調和級数に応じて候補者に配分したという前提で，得票の多い順に当選者を決めていく方法が調和ドント法である．

すなわち通常のドント法では，

$$1:1:1:\ldots$$

という比率で票割をするが，調和ドント法では，

$$1:1/2:1/3:1/4:\ldots$$

という比率で票割をする．このようにすると大政党にとって不利な結果が生じやすい．

なお調和級数の代わりに，単純な等比級数 r^k や，二項級数 ${}_kC_2$ などを用いることも提案されている．しかしどの級数についても一般的傾向は大政党に不利であり，また特定の級数を用いる妥当性の説明が難しい．

(c) ハミルトン(Alexander Hamilton)法

この方法は Samuel F.Vinton 法とも呼ぶ．この方法では，まず各政党に $floor(q[i])$ ずつ議席を与える．次に，$c[i]=q[i]-floor(q[i])$ を計算し，$c[i]$ が大きい順に，余った議席，すなわち $\sum c[i]$ を 1 議席ずつ追加配分する．

下限数と上限数のどちらかに収まるが，アラバマパラドックスを生じる．総議席数単調性を満足しない．

ハミルトン法は19世紀にアメリカ下院の議席定数を州に配分する方法として採用になっていたことがある．しかしアラバマ州で総議席数単調性を満たさない現象が実際に起こったことから，アラバマパラドックスという名称がついた．

以下にハミルトン法の欠陥例を単純化した例で示す．5党の得票率がここに与えたような比率であるとき，総数を 26 から 27 へ増加させたときの配分数をみると，B 党と C 党の配分は増加するが，D 党の配分は減少している．

政党	得票率	議席配分数の変化	
A党	34.8%	9.300 → 9	9.396 → 9
B党	27.6%	7.176 → 7	7.452 → 8
C党	20.2%	5.252 → 5	5.454 → 6
D党	12.7%	3.302 → 4	3.429 → 3
E党	4.5%	1.170 → 1	1.215 → 1
合計	100.0%	26	27

(d) ドループ法

まず各政党に $floor(q'[i])$ ずつ議席を与える．次に，

$$c'[i]=q'[i]-floor(q'[i])$$

を計算し，$c'[i]$ が大きい順に，余っている議席を 1 議席ずつ追加配分する．

下限数と上限数のどちらかに収まるが，アラバマパラドックスを生じる．総議席数単調性を満足しない．この方法はハミルトン法の改善をめざしているが，同じ欠陥を除去できていない．

(e) 改訂ハミルトン法

ハミルトン法を一部修正した方法で，別名改訂 Vinton 法．まず各政党に $floor(q[i])$ ずつ議席を与える．次に，

$$e[i]=(q[i]-floor(q[i]))/x[i]$$

を計算し，$e[i]$ が大きい順に，余っている議席を 1 議席ずつ追加する．

下限数と上限数のどちらかに収まるが，アラバマパラドックスを生じる．やはり総議席数単調性を満足しない．したがって同じ欠陥を除去できていない．

(f) ウェブスター(Daniel Webster)法

別名を，端数四捨五入法(method of major fractions:MF)ともいう．仮の議席総数 v について，

$$w \le v=\sum floor(x[i]/\lambda+1/2)$$

となるような除数 λ を選ぶ．そして仮の議席総数 v を 1 ずつ増加しながら，各政党に，

$$floor(q[i]+1/2)$$

ずつ議席を与えていく．配分議席総数がちょうど w になったところで打ち切る．$w \le v$ という保証はないから，下限数を満足する保証もない．

(g) ハンティントン(E.V.Huntington)法

等比例法(method of equal proportions:EP)という別名もある．仮の議席定数 v を 0から順に 1ずつ増加しながら，ランク指数(rank index) $r[i]$を計算比較し，$1 \le i \le n$ の範囲のすべての i について，

$$r[i] \le r[j]$$

となるような政党 j に $v+1$ 番目の議席を与える．ただしランク指数として，

$$r[i]= x[i]/\sqrt{y[i]\times (y[i]+1)}$$

を用いる．

アラバマパラドックスは生じないが，クォータを満足しない．

ランク指数の定義を変えると，他の方法と等価な方法を得ることができる．

(h) 調和平均法(harmonic mean method)

除数法(divisor method)の一種である．調和平均法の除数基準は，

$$d[a]= 2a(a+1)/(2a+1)$$

である．

除数法[Balinski 75]は一般に次のように述べることができる．

議席定数 w を与えたとき，$0 \le \lambda \le \infty$ の範囲の任意の λ について，

$a(i,\lambda)$を $x[i]/\lambda \le d[a(i,\lambda)]$

となる最小の整数とする．

$$\sum a(i,\lambda)= w' \le w$$

かつ十分小さい ϵ について，

$$\sum a(i,\lambda-\epsilon)= w''> w$$

となるように λ を選ぶ．

また添字集合 $E(\lambda)$ を，

$$E(\lambda)= i : d[a(i,\lambda)]= x[i]/\lambda,$$

$$\mid E(\lambda)\mid = W''- W' \ge 1$$

とする．もし $w''- w'> 1$であれば，$E(\lambda)$の要素を適当に順序づけして α 個だけ要素を選ぶ．ただし $w'+\alpha< w''$.そのα個の添字についてのみ，$y[i]$ を 1ずつ増す．それ以外の添字については $y[i]$ はそのまま変えない．

除数法は，アラバマパラドックスを生じないが，クォータを満足する保証はない．除数基準の定義を変えると，他の方法と等価になる．

(i) 最小除数法(smallest divisors method:SD)

除数法の一種である．ここでの除数基準は，

$$d[a]= a$$

である．

ランク指数を用いる方法と除数基準による方法は関連性があり，(a)(f)(g)(h)(i)の方法について，ランク指数 $r[x,a]$ と除数基準 $d[a]$ は，次のような対応になっている．ただし，

$x \leftarrow x[i]$, $a \leftarrow y[i]$, $a+1 \leftarrow y[i]+1$,

として適用する．

(a) $r[x,a]=x/(a+1)$, $d[a]=a+1$

(f) $r[x,a]=x/(a+0.5)$, $d[a]=a+0.5$

(g) $r[x,a]=x/\sqrt{a(a+1)}$,
$d[a]=\sqrt{a(a+1)}$

(h) $r[x,a]=x/(2a(a+1)/2(a+1)$,
$d[a]=2a(a+1)/2(a+1)$

(i) $r[x,a]=x/a$, $d[a]=a$.

(j) クォータ法(Quota method)

上限下限の範囲内に収まり，しかもハミルトン法のような欠陥もない方法である．バリンスキィの文献[Balinski 75]にこの方法の数学的な性質が出ている．

最初に，$y[1]$〜 $y[n]$ の初期値を 0に設定しておく．総配分数を h として，$h= 1,2,...,w$ と順に，厳密な配分数から既に配分した数を引いた値が最大な箱に 1ずつ追加配分する．すなわち，

$$h\times x[1]/ z-y[1],...,h\times x[n]/ z-y[n]$$

が最大となる添字 k について，$y[k]$ を 1増加する．これを w 回反復する．

アラバマパラドックスを生じない議席総数単調性を満足する方法で，かつ上限数と下限数に収まりクォータを満足する唯一の方法である．

直感的には，クォータ法はハミルトン法に沿った考えかたを受け継ぎながら，その欠陥を是正したアルゴリズムだといえる．

4．おわりに

以上のアルゴリズムを比較してみると，2節にあげた3条件，得票数単調性，クォータ充足性，総議席数単調性をすべて満足する方法は，(j)

クォータ法だけである．したがって比例代表制アルゴリズムとしては，クォータ法を採用することが最善だと結論できる．また2節に示したクォータ法の問題点については，選挙区の有権者数に応じて代議員数を割り当てるのか，政党の得票に応じて当選者数を割り当てるのか，あるいはその他の場合かに応じて，あらかじめ対応策を用意しておくべきであろう．

クォータ法による配分は，クォータ充足性を満足することから，ドント法による配分に比べて容量が小さい箱により多く配分する傾向がある．すなわち，少数意見の尊重の度合が高いといえる．

参考のために，ドント法，クォータ法，ハミルトン法による議席配分のようすを，最近の参議院議員選挙比例代表区の実際のデータに適用したものを付録に示す．

ここでは，個々にどの政党が何議席を獲得したかには興味がなく，アルゴリズムごとの差異にだけ興味がある．したがって，個々の政党名は明示しない．

比例代表制アルゴリズムの選択では，ドント法だけが唯一無二の比例代表制アルゴリズムだと信じ込まないで，別の可能性を念頭に置いて「決めかたをどう決めるか」を考えていくことが重要である．また，個々の政党がどのくらい議席を獲得するかという結果によって，アルゴリズムの選択をすることがないような機構を備えていることも大切である．

謝辞

本研究のプログラミング実験等は，大川情報通信基金および松下通信工業からの研究助成金で購入したワークステーション上で行なった．また本研究の一部は，藤井等君(1979年度:現,日本電気)，本村貞美君(1981年度:現,写研)，小島隆義君(1988年度:現,NTTデータ通信)の卒業研究テーマとしてとりあげ，井内稔技官(山梨大学)ともども共同で研究を行なったものである．ここに記して感謝を表わしたい．

参考文献

[有沢 78] 有沢誠：ソフトウェア思考法，講談社 ブルーバックス B367, 1978

[有沢 81] 有沢誠：比例代表制と甲羅三角形，数理科学 1981年 11月号，pp.78-80

[有沢 84] 有沢誠：プログラム言語への招待，岩波書店 New Science Age 3, 1984

[有沢 86] 有沢誠：コンピュータ言語の学び方，岩波書店 New Science Age 18, 1986

[有沢 89] 有沢誠：比例配分アルゴリズム クォータ法，JUST MOAI 1989年 7月号，pp.18-19

[有沢 90] 有沢誠：比例代表制のプログラム，月刊情報処理試験 1990年 1月号，pp.96-101

[朝日新聞社 83] 比例代表制，朝日新聞社 朝日ブックレット 3, 1983

[Balinski 75] A.L.Balinski, H.P.Young: The Quata Method of appointment, American Mathematical Monthly 1975年 8～9月号，pp.701-730

[小島 89] 小島隆義，有沢誠：決めかたアルゴリズムの比較，情報処理学会第 38回全国大会 5B-10, 1989

[西平 81] 西平重喜：比例代表制，中央公論社 中公新書 615, 1981

[佐伯 80] 佐伯ゆたか:「きめ方」の論理，東京大学出版会，1980

[Steinhaus 48] H.Steinhaus(遠山訳): 数学スナップショット，紀伊国屋書店，1948(1957, 1976)

付録

◇ 1986年のデータによる配分結果の比較

政党数＝ 27 当選者数＝ 50 総得票数＝ 57362745

順位	得票	(a)	(j)	(c)
1=	22132575	22	19	19
2=	9869088	9	9	9
3=	7438501	7	7	7
4=	5430838	5	5	5
5=	3940325	3	3	3
6=	1803051	1	2	2
7=	1759484	1	2	2
8=	1455532	1	1	1
9=	1367291	1	1	1
10=	570995	0	1	1
11=	353334	0	0	0
12=	247559	0	0	0
13=	156100	0	0	0
14=	146243	0	0	0
15=	138656	0	0	0
16=	109607	0	0	0
17=	103375	0	0	0
18=	68972	0	0	0
19=	60488	0	0	0
20=	42804	0	0	0
21=	41275	0	0	0
22=	31464	0	0	0
23=	29278	0	0	0
24=	18025	0	0	0
25=	17827	0	0	0
26=	16048	0	0	0
27=	14010	0	0	0

◇ 1989年のデータによる配分結果の比較

政党数＝ 40 当選者数＝ 50 総得票数＝ 56171228

順位	得票	(a)	(j)	(c)
1=	19688252	20	18	18
2=	15343455	15	14	14
3=	6097971	6	5	6
4=	3954408	4	4	4
5=	2726419	2	2	2
6=	1250022	1	1	1
7=	1179939	1	1	1
8=	993989	1	1	1
9=	872326	0	1	1
10=	711980	0	1	1
11=	682610	0	1	1
12=	341003	0	1	0
13=	334805	0	0	0
14=	319298	0	0	0
15=	173314	0	0	0
16=	161523	0	0	0
17=	150735	0	0	0
18=	147090	0	0	0
19=	145194	0	0	0
20=	139682	0	0	0
21=	132130	0	0	0
22=	121248	0	0	0
23=	72894	0	0	0
24=	60193	0	0	0
25=	44736	0	0	0
26=	43048	0	0	0
27=	41481	0	0	0
28=	41464	0	0	0
29=	32305	0	0	0
30=	29929	0	0	0
31=	24030	0	0	0
32=	23790	0	0	0
33=	18953	0	0	0
34=	15872	0	0	0
35=	14514	0	0	0
36=	10192	0	0	0
37=	8857	0	0	0
38=	8585	0	0	0
39=	8127	0	0	0
40=	4865	0	0	0

本 PDF ファイルは 1990 年発行の「第 31 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の捜索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者捜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間： 2020 年 12 月 18 日 ～ 2021 年 3 月 19 日
掲載日： 2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程
https://www.ipsj.or.jp/copyright/ronbun/copyright.html