# 選挙区割り問題

坂口　利裕，和田　淳一郎

## 1. はじめに

選挙区割り問題は，組合せ最適化問題，とりわけ集合分割問題の応用として知られている[5, 7, 8]．

本稿では，まず，区割り問題の一般的モデルを示し，既存のアルゴリズムについて整理・検討する．次に，我々が実装したプログラムのアルゴリズムの考え方について述べ，いくつかの実証例を示すことによって，アルゴリズムの評価を行う．最後に，今後の課題についてまとめる．

## 2. 選挙区割り問題の概要

### 2.1 最適化の要件

選挙区割りにおいて最も重要視されるのは，人口の平等化である．すべての選挙区で人口が等しいことが最良であるが，行政界との整合，地理的な連続性などの制約により，必ずしも実現できない．衆議院の小選挙区割りでは，「区割り案の作成方針」の中で，次の2点が地理的な基準として具体的に挙げられている．

(1) 飛び地を作らない．

(2) 郡の区域が現に他の都市により分断されている場合を除き郡を割らない．

ここに挙げた以外の基準としては，コンパクト性と呼ばれる，選挙区の形状についての基準が採用されることがある[3]．地理的な連続性を保ちつつ，複雑な形状の選挙区等を避けるための測度である．

### 2.2 最適化モデル

さて，小選挙区の区割りの最適化モデルは，一般に以下のように表記できる．

目的関数：$f(\boldsymbol{X})$→最適化

制約条件：$x_{ir}=\{0,1\}, \forall i\in \boldsymbol{N}, \forall r\in \boldsymbol{K}$

$$\sum_{r\in K} x_{ir}=1, \forall i\in \boldsymbol{N}$$

$$\sum_{i\in N} x_{ir}\geq 1, \forall r\in \boldsymbol{K}$$

さかぐち　としひろ，わだ　じゅんいちろう
横浜市立大学 商学部
〒236-0062 横浜市金沢区瀬戸 22-2

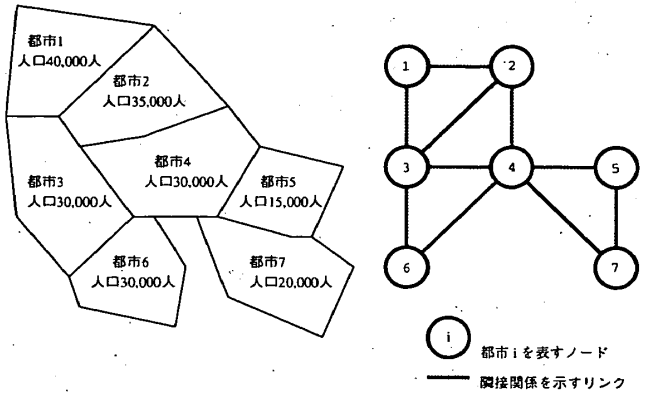


図1　仮想的な地域とそのグラフ表現

ここで，$\boldsymbol{X}\equiv\{x_{ir}|\forall i\in \boldsymbol{N}, \forall r\in \boldsymbol{K}\}$(区割りの解)，$\boldsymbol{N}\equiv\{1, 2, \cdots, n\}$ (都市番号の集合)，$\boldsymbol{K}\equiv\{1, 2, \cdots, k\}$ (選挙区番号の集合) である．

目的関数には，人口の等配分を示す指標，地理的な連続性や形状の状態を示す指標，および，これらの合成指標が採用される．本稿では，地理的な連続性やコンパクト性を，制約の一部として使用することで，列挙する解の数を抑えることに利用した．

### 2.3 データ表現

主たる評価指標である人口データの他に，地理的な連続性の評価が必要となる．先行研究では，地理的な連続性を考慮するために，多くの場合，隣接関係をグラフとして表現したモデルが採用されている．図1は，仮想的な地域を例にグラフ表現を示している．

節2.2で示したモデルの解は，目的関数に陽に組み込まれる場合を除いて，一般に選挙区ごとの地理的連続性を有している保証がないが，グラフモデルを採用することで，検証することが可能である．本稿でも，グラフ論的な距離によってコンパクト性を評価することとした．

### 2.4 解の探索方法

区割りの最適化アルゴリズムを組合せ最適化問題として考えると，次のアプローチが考えられる[4]．

(1) **列挙的手法**：すべての解を列挙して厳密な最適解を見出す．

(2) **発見的手法**：最適解あるいは準最適解を生成するためのアルゴリズムまたはルールを利用して，ただ一つの解を求める．

(3) **探索的手法**：列挙的手法と発見的手法の中間に位置する方法で，解空間の部分空間内を探索することによって，(準)最適解を探し出す．

この種の最適化問題のNP困難性が認知されるようになった近年では，遺伝的アルゴリズムやニューラルネットワークによる，実用解を高速に求める人工知能的アプローチも行われている[6]．

我々自身は，すでに，衆議院小選挙区のうち，比較的サイズの小さな現実の県の小選挙区割りについて，列挙的手法によって検討を加えた[7]．いくつかの政治学的に興味深い知見を得ることはできたが，現実の区割りへの採用を主張するには不十分なものであったことは否めない．

## 3. 探索的手法による区割りアルゴリズム

### 3.1 原形アルゴリズム

本稿で提示するアルゴリズムの原形は，Mehrotraら[2]によっている．Mehrotraらは，このアルゴリズムを最適解を求めるために使用しているわけではなく，隣接状態を満足する初期解の作成プロセスに利用している．そのため，どちらかと言えば，発見的手法に分類されるものである．

基本的なアイデアは，都市群の中から適当なものを一つ選び，あらかじめ設定した選挙区人口の許容範囲に収まるまで隣接した都市を併合していくというものである．最初の選挙区を決めたら，未決定の都市群を対象に同様のプロセスを繰り返していき，合計$(k-1)$回繰り返すことで，$k$個の選挙区が得られる．しかし，以下の点に問題がある．

(1) $(k-1)$までに得られる選挙区は，地理的に連続している保証があるが，$k$番目に得られる選挙区が，単一の連結成分となる保証がない．

(2) すべての選挙区が人口の許容範囲に収まる保証がない．これをできるだけ回避するために，あらかじめ，都市を適当な人口単位に，分割あるいは併合させるという前処理を必要としている．

(3) 発見的手法であるために，基本的には単一の解しか求められない．つまり，併合可能な都市が複数あっても，その一つだけが選ばれて続行されるため，解が最適解でないことがある（バックトラックを考慮していない）．

以上のことから，市の分割・併合といった前処理を施さないことを前提に，次の2点を目標にアルゴリズムを改良することとした．

(1) 併合可能な都市が複数ある場合には，バックトラックできる体系にする．つまり，アルゴリズム全体を探索的手法に改め，明示的に示された条件下の最適解を得ることを目指す．

(2) 細長い形状の選挙区をできるだけ排除できるように併合の過程にコンパクト性を考慮した制約を加える．これによって，探索の数も減らされる．

### 3.2 改良アルゴリズム

原形のアルゴリズムを元にして，バックトラック可能とするための再帰呼出し可能な手続きを用いて，全体を以下に示すように改めた．次の記号を導入する．

$p_i$ ：都市$i$の人口．

$L$ ：選挙区未決定の都市ノードの集合．

$M_r$ ：選挙区$r$を構成する都市ノードの集合．

$\bar{q}$ ：選挙区あたりの平均人口（理想人口）．

$q_r$ ：選挙区$r$の人口．

$q_{\mathrm{min(max)}}$ ：$q_r$の許容範囲の下限（上限）．

$\deg_{M_r}(i)$ ：ノード$i$に接している（リンクを共有している）$M_r$内の都市ノードの数．

$d_{ij}$ ：ノード$i$と$j$とのグラフ上での最短距離（ノード$i$から$j$に至る最短経路の長さ．各リンクの長さを1とし，経路が存在しないものは$\infty$とする）．

$\mathrm{rad}_{M_r}$ ：選挙区$r$を構成するグラフの半径[1]．

$rad_{\max}$ ：コンパクト性を確保するための，各選挙区の$\mathrm{rad}_{M_r}$の上限．

$l$ ：探索レベル．選挙区生成の開始ノード$v$のみで選挙区が構成されている状態を$l=0$，$v$の直近のノードを併合した状態を$l=1$のように定める．つまり，$v$からの距離が$d_{vi}=l$となるノードまでを併合した状態を表している．

$l_{\max}$ ：探索レベルの上限．選挙区の生成過程で，$l \geq 2rad_{\max}$となれば，次のレベル$(l+1)$のノードを加えても，グラフの半径を$rad_{\max}$以下にできない．したがって，$l_{\max} \equiv 2rad_{\max}$．

$A$ ：併合可能なノードを候補から除外するための集合．レベル$l$の状態で，$L$から$(l+1)$の候補を探索する際，レベル$l$以

下のノードは別に探索されるので，それらを除外するために使用する．

改良されたアルゴリズムの全容を以下に示す．

**0：初期設定**

$r\leftarrow 1$，$\boldsymbol{L}\leftarrow\boldsymbol{N}$ として，手続き **do_dist**($r$) を実行する．

**1：手続き do_dist**($r$)

**1-a**：$\boldsymbol{L}$ でなるグラフの連結成分を求め，その数を $n_g$，各連結成分のノード集合を $\boldsymbol{S}_g(g=1,\cdots,n_g)$ とおく．$r-1+n_g$ の値に応じ，以下のいずれかを実行する．

(1) $r-1+n_g=k$ のとき

**1-b**：を実行して呼出し元に戻る．

(2) $r-1+n_g<k$ のとき

**1-c**：を実行して呼出し元に戻る．

(3) $r-1+n_g>k$ のとき

何もしないで呼出し元に戻る．

**1-b**：$g=1,\cdots,n_g$ に対して，$\boldsymbol{M}_{r+g}\leftarrow\boldsymbol{S}_g$，$q_{r+g}\leftarrow\sum_{i\in S_g}p_i$ とする．

$q_{\min}\le q_{r+g}\le q_{\max}$ かつ $\mathrm{rad}_{M_{r+g}}\le rad_{\max}(g=1,\cdots,n_g)$ を満たすなら，目的関数の値を計算して解を更新する．

**1-c**：$g=1,\cdots,n_g$ に対して以下を繰り返す．

(1) $\boldsymbol{S}_g$ の中で $p_i$ が最大のものを $v$ とおく．

(2) $\boldsymbol{L}\leftarrow\boldsymbol{L}\backslash\{v\}$，$\boldsymbol{M}_r\leftarrow\{v\}$，$l\leftarrow 0$，$\boldsymbol{A}\leftarrow\{v\}$，$q_r\leftarrow p_v$ とする．

(3) 手続き **do_merge**($r, l, \boldsymbol{A}$) を実行する．

(4) $q_r\leftarrow q_r-p_v$，$\boldsymbol{M}_r\leftarrow\boldsymbol{M}_r\backslash\{v\}$，$\boldsymbol{L}\leftarrow\boldsymbol{L}\cup\{v\}$ とする．

**2：手続き do_merge**($r, l, \boldsymbol{A}$)

**2-a**：$q_{\min}\le q_r\le q_{\max}$ かつ $\mathrm{rad}_{M_r}\le rad_{\max}$ を満たすなら，手続き **do_dist**($r+1$) を再帰的に実行する．

**2-b**：$l\ge l_{\max}$ ならば呼出し元に戻る．

**2-c**：$\boldsymbol{L}'\leftarrow\boldsymbol{L}\backslash(\boldsymbol{L}\cap\boldsymbol{A})$ とし，$\boldsymbol{L}'$ から，$\deg_M(i)>0$ かつ $q_r+p_i\le q_{\max}$ を満たすノードの集合 $\boldsymbol{C}$ を作る．

**2-d**：$\boldsymbol{C}=\emptyset$ ならば呼出し元に戻る．

**2-e**：$\boldsymbol{C}$ の空(くう)でない部分集合 $\boldsymbol{B}$ について，

$q_r+\sum_{m\in B}p_m\le q_{\max}$ ならば以下を実行する．

(1) $\boldsymbol{L}\leftarrow\boldsymbol{L}\backslash\boldsymbol{B}$，$\boldsymbol{M}_r\leftarrow\boldsymbol{M}_r\cup\boldsymbol{B}$，$q_r\leftarrow q_r+\sum_{m\in B}p_m$ とする．

(2) 手続き **do_merge**($r, l+1, \boldsymbol{A}\cup\boldsymbol{C}$) を再帰的に実行する．

(3) $q_r\leftarrow q_r-\sum_{m\in B}p_m$，$\boldsymbol{M}_r\leftarrow\boldsymbol{M}_r\backslash\boldsymbol{B}$，$\boldsymbol{L}\leftarrow\boldsymbol{L}\cup\boldsymbol{B}$ とする．

$\boldsymbol{C}$ のすべての部分集合について終えたら呼出し元に戻る．

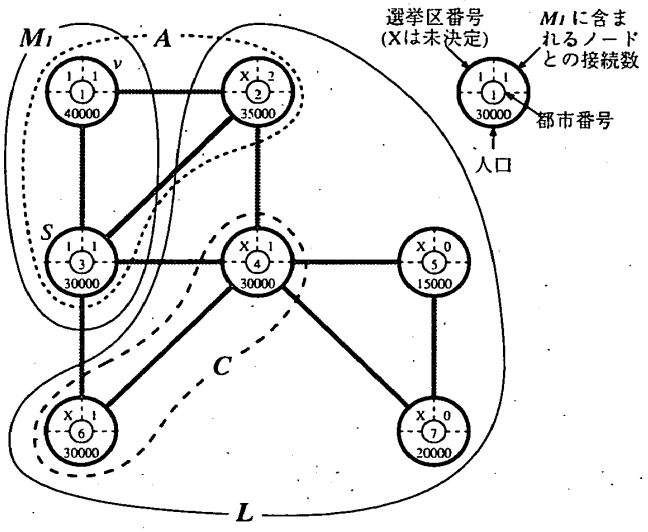


図2　改良アルゴリズムによる探索の様子

図2は，図1に改良アルゴリズムを適用したときの途中経過を示している．ここでは，$k=2$，$q_{\min}=95000$，$q_{\max}=105000$，$rad_{\max}=1(l_{\max}=2)$ としてある．

$\boldsymbol{M}_1$ の開始ノード $v$ は，人口最大の都市1となり，$l=1$ での併合の候補 $\boldsymbol{C}$ は $\{2,3\}$ となる．図2は，3つある $\boldsymbol{C}$ の部分集合のうち，$\{3\}$ を加えた場合の様子である．このとき，$l=2$ の併合の候補は，$\{2,4,6\}$ と存在するように見えるが，都市2については，$l=1$ での探索対象であるため（$\boldsymbol{A}$ に含まれるので）除外され，結果として，$\boldsymbol{C}=\{4,6\}$ のみが $l=2$ での候補となる．

探索の様子全体は，図3のようなツリーとして捉えることができる．探索ツリー上の各ノードで，作成中の選挙区の制約条件が満足されれば，次の選挙区の作成に分岐される．

改良アルゴリズムは，所与の条件下での解の最適性が保証されるので，人工知能的な解法などによる準最適解のように，疑義を生み出す余地はない．

## 4. アルゴリズムの実装と小選挙区割りへの適用

### 4.1 アルゴリズムの実装

改良アルゴリズムは，C言語を用いて実装した．標準的な関数のみを使用したテキストベースのインタフェースとしたため，ANSI準拠のコンパイラであれば，パソコン上でもコンパイルと実行が可能である．

目的関数については，標準偏差の最小化によるもの

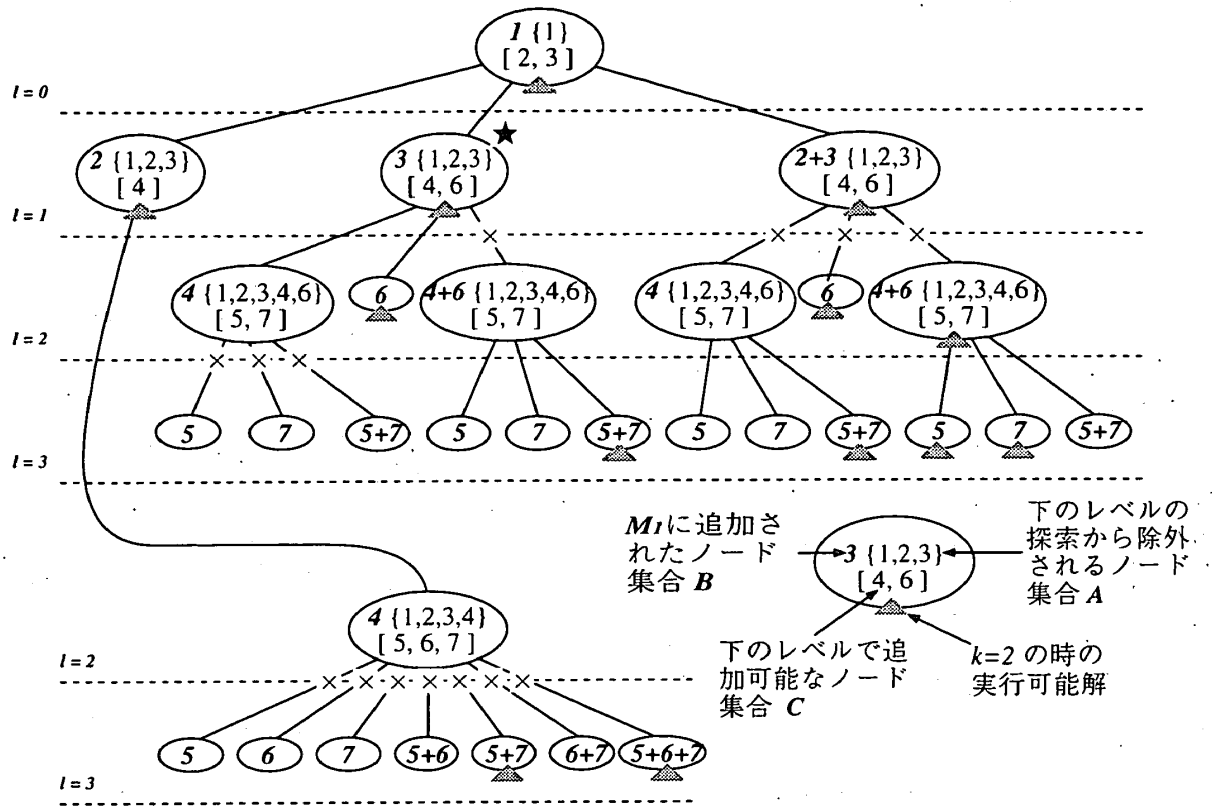


図3 改良アルゴリズムによる探索ツリー（図2の状態は，この図で★を付した部分に相当する．⛰で示した実行可能解は，$q_{min}$，$q_{max}$，$rad_{max}$ を考慮しない場合の解である．本文中に示した条件 $q_{max}=105000$ と $l_{max}=2$ を考慮すれば，図中の×で示した部分より下のレベルは探索されず，実行可能解は5個に減る．さらに，$q_{min}=95000$ を考慮すれば，$M_1$ が，{1} {1,2} {1,3} のときは，次選挙区の生成を行わないので，最終的に目的関数が評価される解は，$M_1=\{1,3,6\}$ のときと，$M_1=\{1,2,3\}$ のときの2個のみとなる）

を使用した．

### 4.2 データ

データは，2002年4月1日現在の市町村割りを基準に，国勢調査（2000年）の人口データ（確定値）を，市および郡（すでに分断されている場合は，便宜的にA，Bなどに区分した）の単位に整理した．隣接関係については，行政界を共有するものを市郡ごとに地図上にて数えあげた．なお，比較対象とした現行区割りは2002年7月31日に公布されたものである．

### 4.3 計算対象

47都道府県を対象にデータ化を試みたが，表1に示す都道府県については，それぞれに付した理由から，作業の対象外とした．よって，対象県は表2に示す25県である．

### 4.4 制約条件

$q_{min}$，$q_{max}$ の目安として，県内での人口格差（以下，県内格差）が1.5倍を超えるケースは意味がないと考えたため，$\bar{q}\pm20\%$ とした．また，$rad_{max}$ は，2および3について計算を行った．

表1 計算対象から除外した都道府県とその理由

| 都道府県 | 除外した主な理由 |
|---|---|
| 北海道 東京 新潟 兵庫 島根 香川 愛媛 長崎 熊本 鹿児島 沖縄 | 隣接関係の認定が困難な離島を抱えている |
| 埼玉（東京）神奈川 静岡 大阪 岡山（熊本） | 全国の理想人口（総人口÷300）の4/3を超える市または特別区を抱えている |
| （北海道）宮城 千葉（神奈川）愛知 京都（大阪）（兵庫）広島 福岡 | 政令指定都市を抱えている |

### 4.5 計算結果の概要

アルゴリズムの適用可能性を広く主張するために，FreeBSDで動作するPentium IIIマシン（クロック周波数933 MHz，メインメモリ256 MB）上で実行した．

解の妥当性は，GIS（*GRASS*）上で図化することでも確認した．紙面の都合から，図化の結果のすべては掲載できないので例示にとどめる（図4）．

計算結果全体の概要を，表2に示す．計算に要する

時間は，$rad_{max}=3$の場合でも，CPU時間で10秒未満のものが14県，残りのうち，7県が，10～180秒である．これ以上のCPU時間を必要とした県は，岩手・茨城・長野・岐阜の4県であった．岩手は約45分のCPU時間を必要としたが，他の3県は60分を超える計算量となったため，人口の制約条件を厳しくして解を求めた．

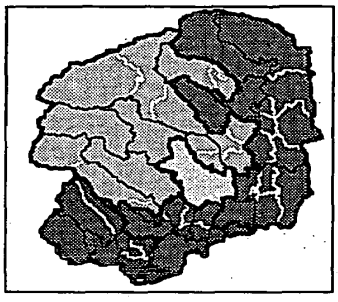


図4　栃木県（最適解）（形状がやや複雑になるものの，現行区割りの県内格差（1.6101）が1.1599に改善される）

### 4.6　計算結果の吟味

現行区割りより県内格差を改善できたものは，現行区割りに一致した石川と福井を含め，21県であった．三重・鳥取・高知・佐賀については，現行区割りより悪い数値となったが，これらは，いずれも「区割り案の作成方針」に忠実には従っていない県である．

三重は，四日市市が，県北の地域を分断するように位置しているため，北部の地域が，単独で選挙区を構成するか（このとき，$\bar{q}-27.20\%$の人口），四日市市に併合されるか（このとき，$\bar{q}+50\%$を超える）しかないためである．このため，人口の制約を±30%に緩和して解を求めざるを得なかった．

鳥取は，東伯郡が東西に分割されているため，現行区割りの県内格差が小さく現れているためである．

表2　計算結果の概要

| 県名 | $n$ | $k$ | リンク数 | 現行区割り 最小 | 現行区割り 最大 | 現行区割り 比率 | $rad_{max}=2$, $\bar{q}\pm20\%$ 最小 | $rad_{max}=2$, $\bar{q}\pm20\%$ 最大 | $rad_{max}=2$, $\bar{q}\pm20\%$ 比率 | $rad_{max}=3$, $\bar{q}\pm20\%$ 最小 | $rad_{max}=3$, $\bar{q}\pm20\%$ 最大 | $rad_{max}=3$, $\bar{q}\pm20\%$ 比率 | 備考 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 青森 | 20 | 4 | 37 | -16.92 | +19.35 | 1.4366 | -12.98 | +11.69 | 1.2835 | -10.01 | +9.07 | 1.2121 | |
| 岩手 | 29 | 4 | 66 | -8.17 | +9.54 | 1.1928 | -1.95 | +0.78 | 1.0279 | -0.20 | +0.59 | 1.0079 | |
| 秋田 | 18 | 3 | 32 | -15.08 | +18.48 | 1.3951 | -6.11 | +6.15 | 1.1306 | -3.43 | +6.15 | 1.0992 | |
| 山形 | 24 | 3 | 50 | -7.51 | +7.21 | 1.1591 | -0.58 | +0.95 | 1.0154 | -0.29 | +0.30 | 1.0060 | |
| 福島 | 27 | 5 | 59 | -23.50 | +27.54 | 1.6674 | -2.95 | +2.90 | 1.0603 | -2.95 | +2.66 | 1.0578 | |
| 茨城 | 43 | 7 | 91 | -31.36 | +20.43 | 1.7544 | -11.23 | +4.01 | 1.1716 | -11.23 | +4.01 | 1.1716 | ‡1,2 |
| 栃木 | 23 | 5 | 53 | -23.41 | +23.32 | 1.6101 | -4.58 | +10.69 | 1.1599 | -4.58 | +10.69 | 1.1599 | ‡2 |
| 群馬 | 27 | 5 | 59 | -12.83 | +19.77 | 1.3739 | -2.53 | +4.48 | 1.0720 | -2.53 | +4.48 | 1.0720 | ‡3 |
| 富山 | 17 | 3 | 33 | -15.32 | +28.14 | 1.5132 | -15.32 | +12.20 | 1.3250 | -2.70 | +2.22 | 1.0506 | |
| 石川 | 18 | 3 | 26 | -14.95 | +15.95 | 1.3633 | — No solution — | | | -14.95 | +15.95 | 1.3633 | ‡4 |
| 福井 | 18 | 3 | 32 | -0.95 | +0.88 | 1.0185 | — No solution — | | | -0.95 | +0.88 | 1.0185 | ‡4 |
| 山梨 | 19 | 3 | 41 | -5.31 | +4.08 | 1.0992 | -0.11 | +0.20 | 1.0031 | -0.11 | +0.20 | 1.0031 | ‡2 |
| 長野 | 40 | 5 | 98 | -28.24 | +21.09 | 1.6875 | -0.76 | +0.66 | 1.0453 | -0.12 | +0.23 | 1.0035 | ‡1 |
| 岐阜 | 36 | 5 | 80 | -14.23 | +20.67 | 1.4069 | -1.16 | +2.89 | 1.0409 | -0.95 | +2.39 | 1.0338 | ‡1 |
| 三重† | 32 | 5 | 56 | -20.36 | +9.15 | 1.3705 | — No solution — | | | -27.20 | +15.80 | 1.5906 | ‡5 |
| 滋賀 | 21 | 4 | 42 | -15.35 | +8.80 | 1.2853 | -7.72 | +4.91 | 1.1368 | -3.37 | +4.25 | 1.0789 | |
| 奈良 | 19 | 4 | 42 | -4.35 | +3.44 | 1.0814 | -1.15 | +1.52 | 1.0271 | -0.96 | +1.52 | 1.0250 | |
| 和歌山 | 15 | 3 | 22 | -17.61 | +9.22 | 1.3257 | -15.17 | +8.39 | 1.2777 | -8.17 | +8.39 | 1.1804 | |
| 鳥取† | 11 | 2 | 14 | -7.20 | +7.20 | 1.1551 | -18.67 | +18.67 | 1.4592 | -18.67 | +18.67 | 1.4592 | ‡2,5 |
| 山口 | 28 | 4 | 52 | -9.80 | +17.22 | 1.2995 | -1.38 | +1.25 | 1.0266 | -1.38 | +0.97 | 1.0239 | |
| 徳島† | 15 | 3 | 30 | -1.26 | +2.02 | 1.0332 | -1.26 | +1.81 | 1.0311 | -1.26 | +1.81 | 1.0311 | ‡2 |
| 高知† | 19 | 3 | 30 | -0.20 | +0.24 | 1.0044 | — No solution — | | | -12.38 | +21.87 | 1.3909 | ‡5 |
| 佐賀† | 21 | 3 | 34 | -1.12 | +1.55 | 1.0271 | — No solution — | | | -2.93 | +4.94 | 1.0810 | |
| 大分† | 27 | 3 | 51 | -5.10 | +7.23 | 1.1299 | — No solution — | | | -4.83 | +7.23 | 1.1267 | |
| 宮崎 | 20 | 3 | 38 | -4.93 | +8.24 | 1.1385 | -3.32 | +1.70 | 1.0518 | -0.66 | +0.72 | 1.0140 | |

$n$は市郡の数．$k$は議席数．リンク数は，県全体を構成するグラフ上におけるリンクの総数である．アルゴリズムの性質上，実行時間がリンク数に依存すると考えられるため，参考値として示した．最小・最大は，それぞれ，県内の理想人口$\bar{q}$からの最小人口区・最大人口区の隔たりを示している（単位は%）．比率は，最大人口区の人口を最小人口区の人口で除したもの（県内格差）．

† 「区割り案の作成方針」に反し，市や連続する郡の分割が行なわれている県．佐賀では，その上「飛び地」が存在する．

‡1 CPU時間を多く必要とするため，人口の制約条件を厳しくして求めた解．茨城は，計算途中で打ち切った暫定解．

‡2 $rad_{max}=2$と$rad_{max}=3$は，同一解．

‡3 $rad_{max}=2$と$rad_{max}=3$は，目的関数の値の異なる別解．

‡4 解は，現行区割りと一致．

‡5 本文参照．

高知は，高知市が単独で $\bar{q}+21.87\%$の人口を抱えているためである．このため，人口の制約を±25%まで緩和して解を求めざるを得なかった．

佐賀は，第2区の飛び地（佐賀郡の南部）に加えて，神埼郡が二つに分割されている．

$rad_{\max}=3$にすることによって大きく改善される県は，細長く市郡が並んでいる，郡の中央部に位置する町村が市として独立したために多くの分断された郡が発生している，といったことが原因となり，$rad_{\max}=2$のコンパクト性条件を満足するのが困難な県であると言える．一方で，$rad_{\max}=2$と3とであまり改善されない県では，選挙区の形状が悪化するものがある．これらの問題に対しては，長く伸びた半島部は一つの郡と同等にみなすといった，客観的にも容認できる作業をした上で，一律$rad_{\max}=2$で解を求めるといった解決策を講じるべきかも知れない．

## 5. まとめ

本稿では，小選挙区の区割りを最適化する手法について概観した上で，列挙法では実用的な速度で解くことの困難な区割りに対しても，最適解を求める実用上有効なアルゴリズムの提示を行った．また，現実の区割りに対する適用の可能性を示した．今回は，行政界を共有しているという隣接関係のみでの区割りを行ったが，交通網や生活圏の実状を反映した，客観的かつ普遍的な指標が与えられれば，それを拠り所とした隣接関係で代替することで，より実効的な解を得ることができるかも知れない（少なくとも，計算量は軽減される）．

パソコン程度でも十分適用可能であることは示せたので，探索の過程をGUIによりモニターできる機能を追加するなど，より手軽なインタフェースを追加することが実務上有益であろう．また，アルゴリズムのさらなる高速化のために，理論的究明を深めることも今後の課題として挙げられる．

**謝辞** 本研究の一部は，2001年度の横浜市立大学研究奨励交付金に基づいて行われた．また，Public Choice Society（2001年，San Antonio），日本選挙学会（2001年，香川大学），および，東京工業大学，U. C. Irvineをはじめとするいくつかの大学のセミナーにおいて，研究成果の一部を報告し，貴重なコメントを頂く機会を与えられた．ここに記して感謝したい．

### 参考文献

[1] Diestel, R.: *Graph Theory,* Springer-Verlag, 2000, 邦訳：根上生也・太田克弘(2000)，「グラフ理論」，シュプリンガー・フェアラーク東京．

[2] Mehrotra, A., Johnson, E. L. and Nemhauser, G.: An Optimization Based Heuristic for Political Districting, *Management Science,* Vol. 44 (1998), pp. 1100-1114.

[3] Young, H. P.: Measuring the compactness of legislative districts, *Legislative Studies Quaterly,* Vol. 13 (1988), pp. 105-115.

[4] 北野宏明(編)：遺伝的アルゴリズム3，産業図書，1997.

[5] 今野浩，鈴木久敏：整数計画法と組合せ最適化，日科技連，1984.

[6] 斎藤孝之，武藤佳恭：ニューラルコンピューティングの遊び方9『小選挙区区割り問題』，bit，Vol. 28 (1996), pp. 88-91.

[7] 坂口利裕，和田淳一郎：「選挙区割りの最適化について」，『三田学会雑誌』，93巻 (2000), pp. 109-137.

[8] 柳浦睦憲，茨木俊秀：組合せ最適化—メタ戦略を中心として—，朝倉書店，2001.