

選挙区割問題に対するヒューリスティクスを用いた ZDD 構築の効率化

中央大学 *千原良太 CHIHARA Ryota
01015123 中央大学 今堀慎治 IMAHORI Shinji

1. はじめに

日本の衆議院議員選挙の小選挙区制における区割画定問題 (以下「選挙区割問題」と称する) とは、各都道府県にあらかじめ定められた選挙区数に対し、市区町村を複数組合せて条件を満たす選挙区を構築する問題である。選挙区割問題の解法の一つとしてゼロサプレス型二分決定グラフ (ZDD) による列挙法 [1] がある。本研究では、ZDD による解法にヒューリスティクスによる近似解法を組み合わせることで、従来より少ないメモリ使用量と実行時間で解を列挙する手法を提案する。

2. 選挙区割問題の定義

選挙区割問題を考えるにあたって、まず都道府県ごとに市区町村とその隣接関係、各市区町村の人口を重みつきグラフ $G = (V, E, w)$ で表現する。入力は、市区町村数を n として、市区町村集合 $V = \{v_1, \dots, v_n\}$ 、市区町村の隣接関係 $E = \{\{v_i, v_j\} \mid \text{市区町村 } v_i \text{ と } v_j \text{ は隣接}\}$ 、市区町村 v_i の人口 w_i 、選挙区数 $m (< n)$ が与えられる。そして出力は、 m 個の選挙区の集合 $S = (S_1, \dots, S_m)$ であり、 S_k は k 番目の選挙区に属する市区町村の集合を表す。ただし、選挙区は以下の制約を満たす必要がある。

制約 1: 選挙区に属する市区町村から誘導される部分グラフは連結である

制約 2: 全ての市区町村は唯一つの選挙区に属す

制約 3: 選挙区は空集合ではない

また、選挙区ごとの人口の和 $P = \{P_1, \dots, P_m\}$ ($P_k = \sum_{v_i \in S_k} w_i$ ($k = 1, \dots, m$)) を計算し、選挙区人口の最小値 $\min(P)$ と最大値 $\max(P)$ を調べる。制約を満たす選挙区割の中で、一票の格差が最小のもの、すなわち $\frac{\max(P)}{\min(P)}$ の値が最小であるものを最適な選挙区割と定める。

3. ゼロサプレス型二分決定グラフ

ゼロサプレス型二分決定グラフ (ZDD) [2] は、組合せ集合を表現する非巡回有向グラフ (DAG) を用

いたデータ構造である。ZDD の大きな特徴は、指数的な個数の組合せを効率的に表現できる点で、二分決定木から不要な節点の削除・共有を行うことで、疎な組合せ集合を高速に列挙することができる。

また、ZDD をトップダウンに構築する手法としてフロンティア法がある。この手法では、根から順に節点を作成し、処理中のラベルをもつ節点集合の中で、共通する節点はその場で共有を行い、制約を満たさない枝はその場で枝刈りすることで、高速に ZDD を構築することができる。

4. 選挙区割列挙アルゴリズム

選挙区割問題では、ZDD の節点のラベルに辺集合 E をそれぞれ割り当てることで、選んだ辺を選挙区の連結部分とし、連結部分からなる部分グラフで選挙区を表現する。節点には辺のラベルの他に、フロンティア上の市区町村の接続関係、確定した部分グラフの個数、部分グラフ中の各連結成分の重みの情報をもつ。また、制約として選挙区の人口上限 $Upper$ と人口下限 $Lower$ を最適な選挙区割を含むような値に定めておく。一般的には、 $ratio$ という許容格差の値を用いる。市区町村の人口の和を W として、 $Upper = \frac{ratio \cdot W}{ratio + (m-1)}$ 、 $Lower = \frac{W}{ratio \cdot (m-1) + 1}$ を求める。これらの情報をもとにフロンティア法により ZDD を構築し、確定した部分グラフの個数が m を超えた場合や $Upper$ や $Lower$ の制約を満たさない部分グラフが存在すると確定した場合に枝刈りを行い、最適な選挙区割の解を含む ZDD を得ることができる。

5. ヒューリスティクスを用いた手法

選挙区割問題の場合、ZDD の節点に持つ情報が複数あり、節点の共有が行うのが難しい構造となる。そこで、ヒューリスティクスでできる限り最適な選挙区割に近い実行可能解を求め、この解の選挙区人口を用いて $Upper$ と $Lower$ の値を定めることで、ZDD の節点の削除を効率よく行い、実

表 1: 入力データ

id	name	$ V $	$ E $	m
4	Miyagi	39	86	5
6	Yamagata	35	85	3
8	Ibaraki	44	94	7
20	Nagano	77	187	5
27	Osaka	72	168	19

表 2: ヒューリスティック解

id	$Upper$	$Lower$
4	467,561	451,162
6	356,505	355,396
8	419,212	391,937
20	410,752	408,772
27	569,011	424,530

行時間の短縮と省メモリ化を目指す。本研究では、ヒューリスティクスとして焼きなまし法を実装し、連結性を保ちながらある市区町村を別の選挙区に組み込む近傍探索を繰り返す。

6. 計算機実験

令和 2 年度の国勢調査のデータを用いて、制約に $ratio$ を使用する手法とヒューリスティック解を用いる手法の 2 通りで、選挙区割を列举する ZDD の構築を行った。 $ratio$ はほぼ全ての都道府県の列举が可能な値として 1.4 を採用した。実験環境は CPU が Intel Xeon E5-2687W-v4 (3.00GHz)、メモリが 512GB のマシンを利用し、プログラムは C++17, ライブラリとして SAPPOROBDD¹ と TdZDD² を使用した。本稿では一部の結果を抜粋して掲載する。表 1 は入力データのサイズを表し、表 2 では、ヒューリスティクスを用いて導出した $Upper$ と $Lower$ の値を示している。表 3 と表 4 では、それぞれの手法において生成した ZDD の結果を表しており、node は ZDD のノード数、solve は解の個数、time は実行時間、mem はメモリ消費量を示している。

実験結果では、ほぼ全ての都道府県でヒューリスティクスを用いた手法が、 $ratio$ を用いる手法に比べて実行時間とメモリ消費量は大きく削減されていた。長野県は $ratio=1.4$ ではメモリ不足で解

表 3: $ratio = 1.4$ を用いた ZDD 構築結果

id	node	solve	time(sec)	mem(MB)
4	6,581	40,106	3.38	652
6	319,171	7,493,473	33.81	6,702
8	1,077,156	36,745,326	212.75	30,690
20	N/A	N/A	N/A	N/A
27	N/A	N/A	N/A	N/A

表 4: ヒューリスティクスを用いた ZDD 構築結果

id	node	solve	time(sec)	mem(MB)
4	55	2	0.01	4
6	4,416	541	0.08	19
8	1,340	390	0.02	6
20	10,302	17,657	2.3	471
27	N/A	N/A	N/A	N/A

が得られなかったものの、ヒューリスティクスの方では 2.3 秒で解を得ることができた。ただし、大阪府は $Upper$ と $Lower$ の差が大きく、どちらの手法もメモリ不足で解を得られなかった。

7. 結論

本研究では、選挙区割問題に対して、ヒューリスティクスで得られる解を利用し、効率的に ZDD を構築できることを提案した。 $ratio$ で制約を決める既存手法では、人の手で ZDD が構築可能な値を探索しなければならなかったが、提案手法でその作業を不要にし、さらに ZDD 構築において計算時間と消費メモリを大幅に削減する効果が得られることを確認した。

謝辞

ZDD の実装にあたって多大なるご協力をいただいた川原純准教授（京都大）に心から感謝いたします。

参考文献

- [1] Kawahara, J. et al.: Generating All Patterns of Graph Partition Within a Disparity Bound, WALCOM, pp. 119-131 (2017).
- [2] Minato, S.: Zero-suppressed BDDs for set manipulation in combinatorial problems, Proceedings of the 30th international Design Automation Conference, pp. 272-277 (1993).

¹<https://github.com/Shin-ichi-Minato/SAPPOROBDD>

²<https://github.com/kunisura/TdZdd>