# クラスター編集問題に対する効率的な解法の研究
# Study on Efficient Algorithms for Cluster Editing Problem

情報工学専攻　木村 陸　Riku Kimura

概要

クラスター編集問題とは, 単純無向グラフを最小回数の辺操作で非連結なクリークの集合に変換する問題である. この問題は NP 困難であることが知られているが, 本研究ではヒューリスティックや分枝限定アルゴリズムなどいくつかの手法を組み合わせることで計算の高速化を行う.

キーワード:クラスター編集問題, クラスタリング, 分枝限定法

## 1　はじめに

**クラスター編集問題** (または相関クラスタリング問題) は以下のように定義される最適化問題である. 無向グラフ $G$ が与えられる. 各連結成分がクリークであるグラフのことを**クラスターグラフ**と呼ぶ. $G$ に辺を追加したり, $G$ から辺を削除することで $G$ をクラスターグラフに変えたい. この際に, 必要な追加 · 削除の回数を最小化せよという問題である.

クラスター編集問題はクラスタリングへの応用を動機として導入された. クラスタリングとは, 与えられたオブジェクト同士を互いによく似たオブジェクトによって構成される部分集合に分割するタスクのことである. クラスター編集問題をクラスタリングに応用する際には, 頂点をオブジェクトに対応させ, 似ているオブジェクトに対応する 2 頂点間に辺を定義することで無向グラフを作成する. このグラフ上でクラスター編集問題を解くことでクラスタリングを行う. この手法は遺伝子やタンパク質をクラスタリングする際に用いられてきた [1,3,6].

クラスター編集問題は NP 困難であることが Křivánek と Moráve[4] により示された. しかしながら, クラスタリングへの応用を考えると, クラスター編集問題の大規模な問題例を現実的な時間で解く必要性が高い. そこで, 本研究ではより大規模な問題を厳密に解くアルゴリズムの研究を行う. 基礎的な分枝限定アルゴリズムを実装し, そこに様々な工夫を導入する. 例えば, 最適解の目的関数値の上界の計算, 再帰の深さに応じた計算の省略といった工夫である. これらのアルゴリズムをプログラムとして実装し, 計算実験を通してその効果を検証する.

## 2　問題設定

クラスター編集問題はコスト関数 $s{:}\binom{V}{2} \to \mathbb{Z}$ を導入することで以下のように拡張できる. 頂点の部分集合 $C \subseteq V$ に属する任意の 2 頂点ペア $u, v$ のコストが $s(uv) \geq 0$ であるとき, $C$ を $G$ の **クリーク** と言う. グラフ $(V, \{uv \in \binom{V}{2} | s(uv) > 0\})$ の各連結成分がクリークであるとき, $V$ と $s$ の組 $(V, s)$ を **クラスターグラフ** と呼ぶ. 頂点ペア $uv$ のコスト $s(uv)$ の符号を反転する操作を **uv に対する辺操作** と呼ぶ. この辺操作のコストは $|s(uv)|$ と定義される. コストが正である頂点ペアのことを**正ペア**, コストが負である頂点ペアのことを**負ペア**と呼ぶ. また, 3 頂点の組 $uvw$ について, $s(uv) > 0$, $s(vw) > 0$ であるが, $s(uw) < 0$ である場合, 組 $uvw$ を **衝突** と呼ぶ.

**定理 1**[2]. グラフ $G$ が **クラスターグラフ** であるための必要十分条件は, $G$ に衝突が無いことである.

クラスター編集問題の目標は, 辺操作を繰り返して衝突を全て削除することということもできる.

パラメータ $k$ を導入した問題についても, 説明する. 通常, 問題を解く上でパラメータは必須ではないが, クラスター編集問題は固定パラメータ容易であり, 本研究で用いる分枝限定アルゴリズムは固定パラメータ・アルゴリズムである. したがって, 辺操作のコストを $k$ とする問題を考える.

**パラメータ・コストありクラスター編集問題**
入力:
・頂点集合 $V$
・コスト関数 $s : \binom{V}{2} \to \mathbb{Z}$
・パラメータ $k \in \mathbb{Z}$
出力:
$(i)$ $(V, s)$ をクラスターグラフにするコストが $k$ 以下の辺操作集合が存在するならば, 最小コスト辺操作集合を出力する.
$(ii)$ それ以外の場合, 「解なし」を出力する

クラスター編集問題の整数計画問題としての表現と, 線形計画緩和について説明する. $x$ を 0,1 の 2 値をとるベクトルとする. 全ての頂点ペア $(i, j) \in \binom{V}{2}$ に対して, $x_{ij} = 0$ ならば $i$ と $j$ は同じクラスターに属し, $x_{ij} = 1$ ならば $i$ と $j$ は同じクラスターには属さないと対応させることで, クラスターグラフを $x$ で表現する. これにより, 整数計画問題は以下のように定式化できる.

$$\begin{aligned} \min \quad & \sum_{(ij|s(ij)>0)} s(ij)x_{ij} - \sum_{(ij|s(ij)<0)} s(ij)(1 - x_{ij}) \\ \text{s.t} \quad & x_{ij} \le x_{jk} + x_{ik}, \quad i, j, k \in V, \\ & x_{ij} \in \{0, 1\}, \quad ij \in \binom{V}{2}. \end{aligned}$$

$x$ を 0,1 の 2 値から $0 \le x \le 1$ の実数へと緩和することにより, 線形計画緩和問題が定義できる.

## 3 禁止操作 · 縮約操作

本章では, パラメータ · コストありクラスター編集問題について考える. 問題を解く上で, コスト $s(uv) \ge 0$ であるような頂点ペア $uv$ について, **禁止** あるいは **縮約** 操作を行う場合がある. 以下ではこれらの操作について定義する.

### 3.1 禁止操作

**禁止** とはペア $uv$ のコストを $-\infty$ に設定することである. 禁止後のコストが $-\infty$ なので, これ以降辺操作を $uv$ に行うことができないことを意味している. よって, $u$ と $v$ が同じクリークにならないことを強制する意味がある.

禁止操作によるコスト関数とパラメータの変化を定義する. 頂点集合 $V$, コスト関数 $s : \binom{V}{2} \to \mathbb{Z}$, パラメータ $k$ である問題例 $(V, s, k)$ の頂点ペア $uv$ を禁止操作した後の問題例を $(V, s', k')$ と定義する. このとき, $k' = k - s(uv)$, $s'$ は $s'(uv) = -\infty$, $s'(u'v') = s(u'v')$ $(u'v' \in \binom{V}{2} \backslash \{uv\})$ と定義する. $uv$ を禁止するときは, パラメータ $k$ を $s(uv)$ だけ減らす. これは, $uv$ を禁止することは, $uv$ に辺操作を行い, その後の操作を禁止することと同義であるからである.

### 3.2 縮約操作

**縮約** とは, 頂点 $u$, $v$ を単一の頂点 $u'$ に置き換え, 任意の頂点 $w \in V \backslash \{u, v\}$ について, 頂点ペア $uw$, $vw$ を $u'w$ に置き換える操作のことである. この操作は $u$ と $v$ が同じクリークになることを強制する意味がある. 縮約操作による頂点集合, コスト関数とパラメータの変化を定義する. 頂点集合 $V$, コスト関数 $s : \binom{V}{2} \to \mathbb{Z}$, パラメータ $k$ である問題例 $(V, s, k)$ の $uv$ を縮約操作した後の問題例を $(V^*, s^*, k^*)$ と書く. 縮約操作では, 頂点 $u$ と $v$ を新たな頂点 $u'$ で置き換える. つまり, $V^* = (V \cup \{u'\}) \backslash \{u, v\}$ となる. また, $k^* = k - \sum_{w \in V \backslash \{u,v\}, s(uw) \cdot s(vw) < 0} \min\{|s(uw)|, |s(vw)|\}$, $s^*$ は任意の $w \in V \backslash \{u, v\}$ について

$$s^*(xw) = \begin{cases} s(uw) + s(vw) & x = u' \\ s(xw) & x \ne u' \end{cases}$$

と定義する.

## 4 アルゴリズム

本研究では分枝限定法に基づくアルゴリズムについて検討する. 4.1 節で分枝限定法の全体像について説明し, その後, 詳細の説明を行う.

### 4.1 分枝限定アルゴリズム

このアルゴリズムでは, 頂点集合 $V$ と頂点間のコスト $s : \binom{V}{2} \to \mathbb{Z}$ が与えられる. コスト $k$ 以下の辺操作集合が存在するならば最小コスト辺操作集合, コスト $k$ 以下の辺操作集合が存在しないならば「解なし」というメッセージを出力とする.

**分枝限定アルゴリズム**

(1) $(V, s)$ を入力として最適解の上界計算を行い, 出力された解のコストを $k$ とする.

(2) 衝突が 1 つも無ければ $\emptyset$ を解として出力して終了. $(V, s)$ の衝突に含まれる正ペア $uv$ を選択する.

(3) 問題例 $(V, s, k)$ で $uv$ を禁止した問題例 $(V, s', k')$ を再帰的に解き, 得られた解を $\mathbb{F}'$ とする. $\mathbb{F}'$ のコストを $s(\mathbb{F}')$ とする. $\mathbb{F}'$ が「解なし」ではない場合, $k := s(\mathbb{F}) + s(uv)$ とする.

(4) 問題例 $(V, s, k)$ で $uv$ を縮約した問題例 $(V^*, s^*, k^*)$ を再帰的に解き, 得られた解を $\mathbb{F}^*$ とする. $\mathbb{F}^*$ から $(V, s, k)$ の解 $\overline{\mathbb{F}^*}$ を計算する.

(5) $\overline{\mathbb{F}^*}$ が「解なし」ではない場合, $\overline{\mathbb{F}^*}$ を出力する. $\overline{\mathbb{F}^*}$ が「解なし」, $\mathbb{F}'$ が「解なし」ではない場合, $\mathbb{F}' \cup \{uv\}$ を出力する. それ以外の場合, 「解なし」を出力する.

### 4.2 上界計算

4.1 節で与えた分枝限定アルゴリズムではパラメータ $k$ を設定する. このパラメータは, 最適解のコストの上界でなければアルゴリズムは解を出力しない. よって, 本研究では, 分枝限定アルゴリズムを呼び出す前にヒューリスティックアルゴリズムで最適解のコストのなるべく小さな上界を求めることで高速化を目指す. 本研究で使用したヒューリスティックアルゴリズムは以下の 3 つである.

**・ランダムピボット**:ランダムに頂点を選びクリークを作る. この作業をクラスターグラフになるまで繰り返し行う.

**・LP ピボット**:ランダムに頂点を選び, その頂点の同じクラスターに含まれるかどうかを線形計画緩和の最適解の値を用いて判定しクリークを作る. この作業をクラスターグラフになるまで繰り返し行う.

**・LP を用いた分枝限定ヒューリスティック**:衝突内の正ペア $uv$ について, 線形計画緩和の解で $uv$ が同じクラスターに属するなら縮約操作を, 別のクラスターなら禁止操作を行う. その上で分枝限定アルゴリズムを適用し解を計算する.

### 4.3 正ペア $uv$ の選択

本節では分枝限定アルゴリズムのステップ (2) で行う正ペア $uv$ の選択方法について, いくつかのアルゴリズムを説明する.

**・正ペア $uv$ を選択する素朴なアルゴリズム**:未探索の正ペア $uv$ について, ある頂点 $w \in V \backslash \{u, v\}$ について, $uvw$ が衝突ならば $uv$ を選択する.

**・衝突の数を考慮した選択**:正ペア $uv$ を選択する素朴なアルゴリズムにおいて, $uv$ を含む衝突の数が最大の正ペアを選択する.

**・衝突の数を考慮した選択の簡略化**:再帰計算の度に行っていた衝突の計算を再帰の深さに応じて計算するようにする.

### 4.4 LP の解に応じて再帰中の禁止と縮約の順番を変える

4.1 節の分枝限定アルゴリズムではどの衝突も禁止, 縮約の順番で再帰計算を行なっていた. そこで線計画緩和の解に応じて禁止と縮約の順番を入れ替えることにする. 分枝限定アルゴリズムで選択された正ペア $uv$ に対して線形計画緩和の解で同じクラスターに属するなら禁止と縮約の順番を入れ替え, そうでないなら通常通りに再帰計算を行う.

## 5 実験

本研究では, 4 章で提案したアルゴリズムを実装し, その性能を計算実験を通して評価した. 実装したアルゴリズムをいくつかの問題例に適用し, 結果を比較する. データはクラスター編集問題を扱ったコンペティション, PACE Challenge 2021[5] で用いられた公開インスタンスの一部を用いる.

4.2 節で述べた, 3 つの上界計算アルゴリズムを解の質と計算速度の観点から比較する. 解の目的値の結果を表 1 に示す. ランダムピボットを手法 1, LP ピボットを手法 2, LP を用いて分枝限定ヒューリスティックを手法 3 とする. 手法 1 は手法 2 に対して優れたアルゴリズムとは言えないが, 両者ともに高速なアルゴリズムなので, 両方計算し優れた解を用いることが効果的であると言える. 手法 3 は解の目的値は他の 2 つより優れているが, 計算時間が厳密アルゴリズムと大差ないことを考慮すると効果的な

アルゴリズムであるとは言えない.

4.3 節と 4.4 節で与えた再帰回数削減のための 3 つの手法を分枝限定アルゴリズムに導入した際の効果について検討する. それぞれを単体および組み合わせて導入した場合の再帰回数と計算時間を比較していく. 計算時間の結果を表 2 に示す. 正ペア $uv$ を選択する素朴なアルゴリズムを工夫なし, 衝突の数を考慮した選択を工夫 1, 衝突の数を考慮した選択の簡略化を工夫 2, LP の解に応じて再帰中の禁止と縮約の順番を変えるを工夫 3, 3 つの工夫全てを組み合わせたものを全てと呼ぶことにする. 単体で導入した場合の性能を比較すると, 工夫 1 が一番優れていた. 複数の手法を組み合わせた場合を比較した場合, 工夫 1+2 が最も優れていた. 逆に, 工夫 3 は他の手法と組み合わせても良くなるどころかかえって悪化してしまった.

表 1: ヒューリスティックアルゴリズムの出力解の目的値

| 問題例 | 手法 1 | 手法 2 | 手法 3 |
|---|---|---|---|
| 001 | 3 | 3 | 3 |
| 003 | 50 | 59 | 45 |
| 005 | 55 | 46 | 46 |
| 007 | 110 | 125 | 91 |
| 009 | 116 | 140 | 100 |
| 011 | 97 | 81 | 81 |

表 2: 再帰回数削減のための工夫を導入した際の計算時間 [s]

| 問題例 | 工夫なし | 工夫 1 | 工夫 1+2 | 全て |
|---|---|---|---|---|
| 001 | $4 \times 10^{-6}$ | $3 \times 10^{-6}$ | $10^{-6}$ | $10^{-5}$ |
| 003 | 0.872301 | 0.81555 | 0.669425 | 1.09219 |
| 005 | 1.16827 | 1.0278 | 0.717705 | 1.2961 |
| 007 | 540.535 | 434.05 | 245.095 | 545.024 |
| 009 | 347.233 | 172.684 | 78.8172 | 207.93 |
| 011 | 18.499 | 15.7262 | 5.65564 | 16.5961 |

## 6 結論

本研究では, 大規模なクラスター編集問題を解くための厳密アルゴリズムについて研究した. 基本となる分枝限定アルゴリズムにカーネル化アルゴリズムのアイデアを用いた前処理など高速化のために様々な工夫を導入し, 実際に実装し評価した.

導入した工夫の中では特にヒューリスティックで計算した最適解の上界をパラメータとして用いる手法は大きな効果を発揮した. また, 線形計画緩和を用いた上界計算や再帰回数制限の工夫はともに期待した効果は発揮されなかった.

## 参考文献


[1] Ben-Dor, A., Shamir, R., Yakhini, Z.: Clustering gene expression patterns. J. Comput. Biol. 6(3-4), 281–297 (1999)

[2] Böcker, S., Baumbach, J.: Cluster editing. In Paola Bonizzoni, Vasco Brattka, and Benedikt Löwe, editors, Proceedings of the 9th Conference on Computability in Europe, CiE 2013. LNCS, vol. 7921, pp. 33–44. (2013)

[3] Hartuv, E., Schmitt, A.O., Lange, J., Meier-Ewert, S., Lehrach, H., Shamir, R.: An algorithm for clustering cDNA fingerprints. Genomics 66(3), 249–256 (2000)

[4] Křivánek, M., Moráve, J.: NP-hard problems in hierarchical-tree clustering. Acta Inform. 23(3), 311–323 (1986)

[5] PACE Challenge. https://pacechallenge.org/2021/.

[6] Sharan, R., Maron-Katz, A., Shamir, R.: CLICK and EXPANDER: A system for clustering and visualizing gene expression data. Bioinformatics 19(14), 1787–1799 (2003)