

サイズ制約付きハイパーグラフ分割問題に対する 高速アルゴリズム

土中 哲秀¹ 小野 廣隆² 山田 秀流¹ 吉渡 叶²

¹ 九州大学

² 名古屋大学

概要

本研究では、ハイパーグラフ上のグラフ分割問題である最大（最小）固定サイズハイパーグラフ分割問題（最大（最小）FCHGP）を考える。この問題の目的は、与えられた辺重み付きハイパーグラフ $\mathcal{H} = (V, \mathcal{E})$ と正の整数 k に対し、以下の関数 $\varphi_{\mathcal{H}}(S) := \sum_{1 \leq i \leq \max |e \cap S|} \alpha_i \sum_{|e \cap S| = i} w_e$ を最大（最小）とするようなサイズ k の頂点部分集合 $S \subseteq V$ を求めることである。ただし、 α_i は 0 以上 1 以下の実数として与えられ、 w_e はハイパー辺 $e \in \mathcal{E}$ の辺重みである。最大（最小）FCHGP は、最密（最疎） k -部分グラフ問題、最大（最小） k -頂点被覆問題、最大（最小） $(k, n-k)$ -カット問題を含む最大（最小）固定サイズグラフ分割問題（最大（最小）FCGP）をハイパーグラフ上に一般化したものである。本研究ではこの問題に対し、ハイパーグラフのライングラフが木である場合に多項式時間で解けることを示す。

1 はじめに

重み付き無向グラフ $G = (V, E)$ において、頂点部分集合 $S \subseteq V$ に対する関数を $\varphi_G(S) := (1 - \alpha) \cdot w(S) + \alpha \cdot w(S, V \setminus S)$ と定義する。ただし、 α は $0 \leq \alpha \leq 1$ を満たす定数である。このとき、**最大（最小）固定サイズグラフ分割問題（Fixed Cardinality Graph Partitioning, FCGP）** とは、重み付き無向グラフ $G = (V, E)$ と正の整数 k が与えられたとき、 $\varphi_G(S)$ を最大（最小）にする大きさ k の頂点部分集合 S を求める問題である。この問題は、最密（最疎） k -部分グラフ問題（ $\alpha = 0$ ）、最大（最小） k -次数和問題（ $\alpha = 1/3$ ）、最大（最小） k -部分頂点被覆問題（ $\alpha = 1/2$ ）、最大（最小） $(k, n-k)$ -カット問題（ $\alpha = 1$ ）などのグラフ最適化問題を一般化したものである。最大（最小） k -次数和問題を除くこれらの問題は NP 困難であることが知られているため、FCGP も NP 困難である。パラメータ化計算量に関しては、 k に関して FCGP は W[1] 困難であることが知られている一方で [2]、木幅に関する固定パラメータ容易アルゴリズムが知られている [1]。さらに、縮退数と解サイズをパラメータとしたときのパラメータ化計算量 [4, 5] や解サイズをパラメータとしたパラメータ化近似アルゴリズム [3] についても研究されている。

本研究では、ハイパーグラフにおける最大（最小）固定サイズグラフ分割問題を考える。 $\mathcal{H} = (V, \mathcal{E})$ を重み付きハイパーグラフとする。各ハイパー辺 $e \in \mathcal{E}$ の辺重みを $w_e : \mathcal{E} \rightarrow \mathbb{R}$ としたとき、頂点部分集合 $S \subseteq V$ に関する評価関数

$$\varphi_{\mathcal{H}}(S) := \sum_{1 \leq i \leq \max |e \cap S|} \alpha_i \sum_{|e \cap S| = i} w_e$$

を最大化（最小化）するサイズ k の頂点部分集合 $S \subseteq V$ を求める問題を固定サイズハイパーグラフ分割問題（Fixed Parameter Hiper Graph Partitioning, FCHGP）と呼ぶこととする。この問題は、ヒッティングセット問題などハイパーグラフ上の様々な最適化問題の一般化となっている。

2 準備

$\mathcal{H} = (V, \mathcal{E})$ をハイパーグラフとする。このとき、 \mathcal{H} のライングラフを $L(\mathcal{H}) = (\mathcal{E}, E_{\mathcal{H}})$ 、ただし、 $E_{\mathcal{H}} = \{\{e_1, e_2\} \mid e_1, e_2 \in \mathcal{E}, e_1 \cap e_2 \neq \emptyset\}$ で定義する。ハイパーグラフ $\mathcal{H} = (V, \mathcal{E})$ の要素 $v \in V$ に対して、 $f(v) = |\{v \subseteq e \mid e \in \mathcal{E}\}|$ を v の頻度と定義する。また、 $f(G) = \max_{v \in V} f_v$ を G の頻度と定義する。

3 ライングラフが木であるときの多項式時間アルゴリズム

本節では、ハイパーグラフ \mathcal{H} のライングラフが木であるとき、FCHGP に対する多項式時間アルゴリズムが存在することを示す。まず、以下の補題を示す。

補題 3.1. ハイパーグラフ $\mathcal{H} = (V, \mathcal{E})$ のライングラフが木であるならば、 \mathcal{H} の頻度は高々2である。

Proof. ハイパーグラフ \mathcal{H} の頻度が3以上であるとする、 $|\{v \subseteq e \mid e \in \mathcal{E}\}| \geq 3$ となる頂点 v が存在する。したがって、 v を含むハイパー辺が少なくとも3つ存在する。これらのハイパー辺は v に関して互いに交差するので、 \mathcal{H} のライングラフは大きさ3のクリークを含む。これはライングラフが木であることに矛盾する。 \square

以下では補題 3.1を用いて、動的計画法に基づく多項式時間アルゴリズムを設計する。

定理 3.1. ライングラフが木であるハイパーグラフ \mathcal{H} 上において、FCHGP に対する多項式時間アルゴリズムが存在する。

Proof. 木 $L(\mathcal{H})$ 上の動的計画法に基づくアルゴリズムを以下では設計する。本証明では、最大化の場合のみ考える。最小化も同様に設計可能である。

ノード $e \in V(T)$ において、各 e の子を f_1, \dots, f_q とする。ここで、 q は e の子の数である。このとき、補題 3.1より、各頂点は高々2つのハイパー辺にしか現れないので、 $i \neq j$ となる任意の $i, j \leq q$ に対して、 $f_i \cap f_j = \emptyset$ である。さらに、 T における e を根とする部分木 T_e によって誘導される部分ハイパーグラフを $\mathcal{H}_e = (\{v \in e \mid v \in V(T_e)\}, V(T_e))$ とする。このとき、任意の $e, e' \in \mathcal{E}$ に対して \mathcal{H}_e と $\mathcal{H}_{e'}$ は共通の頂点を持たない。

\mathcal{H}_e において大きさ k' で $|S \cap e| = d$ を満たす S に対する $\varphi_{\mathcal{H}_e}(S)$ の最大値を $\text{OPT}_1[e, k, d]$ とする。定義より、 T の根を $r \in \mathcal{E}$ とすると $\max_{0 \leq d \leq |e|} \text{OPT}_1[r, k, d]$ が $\mathcal{H} = \mathcal{H}_r$ における FCHGP の最適値となる。以下では、各 $e \in V(T)(= \mathcal{E})$ において、 $\text{OPT}_1[r, k, d]$ を再帰的に計算する。ここで、 $\text{OPT}_1[e, k, d]$ について $d \leq k$ のみ計算すれば十分であることに注意されたい。

まず、 e が葉である場合は以下のようになる。

$$\text{OPT}_1[e, k, d] = \begin{cases} \alpha_d w_e & (d = k) \\ -\infty & (\text{otherwise}) \end{cases}.$$

内部ノード e において, $\text{OPT}_1[e, k, d]$ を計算するために, f_1 から f_i を根とする部分木のみに含まれるハイパー辺とその内部の頂点で構築される部分ハイパーグラフ $\mathcal{H}_e^{(i)} = (\{v \in e \mid e \in \bigcup_{j=1}^i V(T_{f_j})\}, \bigcup_{j=1}^i V(T_{f_j}))$ を考える. この時 $\text{OPT}_2[e, i, k', d']$ を, $\mathcal{H}_e^{(i)}$ において k' の頂点が解に含まれ, そのうち d' 個の頂点が e にも含まれている場合の最適値とする. よって, e の子数を q とすると, $\text{OPT}_2[e, q, k', d']$ は, 部分木 T_{f_1}, \dots, T_{f_q} が誘導する部分ハイパーグラフで, k' の頂点が解に含まれ, そのうち d' 個の頂点が e にも含まれている場合の最適値である.

k_e を T_e において, e のみに含まれる解の個数とする. このとき, $\text{OPT}_1[e, k, d]$ は, 各部分木で k_i 個ずつの頂点が解に選ばれ, かつ e と d_i 個共有するもので, $\sum_i k_i + k_e = k$ かつ $\sum_i d_i + k_e = d$ を満たす各部分木の評価値の総和に等しい. よって, $\text{OPT}_2[e, q, k', d']$ の定義より, $\text{OPT}_1[e, k, d]$ は以下のように書ける.

$$\text{OPT}_1[e, k, d] = \max_{\substack{0 \leq k' \leq k \\ 0 \leq d' \leq d \\ k - k' = d - d'}} (\text{OPT}_2[e, q, k', d'] + \alpha_d w_e).$$

式の第2項は, e が d 個の解に含まれる頂点を含むことからその分を足し合わせたものである.

以下では, $\text{OPT}_2[e, i, k, d]$ の計算方法を示す. 常に $d \leq k$ なので, OPT_2 についても $d \leq k$ の場合のみ計算すれば十分である. d' を e と $i-1$ 個目までの子との共通部分に含まれる解の個数, d_i を f_i が e との共通部分以外に持つ解の個数とする. T_{f_i} と T_{f_j} は共通の要素を持たないことと, $\text{OPT}_2[e, i, k, d]$ は $\mathcal{H}_e^{(i-1)}$ の最大評価値と \mathcal{H}_{f_i} の最大評価値を足し合わせたものに等しいため, 以下の再帰式が得られる.

$$\text{OPT}_2[e, i, k, d] = \max_{\substack{k = k' + k_i \\ d = d' + d_i \\ 0 \leq d_i \leq \max |e \setminus f_i|}} (\text{OPT}_2[e, i-1, k', d'] + \text{OPT}_1[f_i, k_i, d_i])$$

特に $i = 1$ については,

$$\text{OPT}_2[e, 1, k, d] = \text{OPT}_1[f_1, k, d]$$

である.

最後に計算時間について議論する. OPT_1 の DP テーブルのサイズは $O(m \cdot k^2)$ である. $\text{OPT}_1[e, k, d]$ の計算にかかる時間は, $O(k^2)$ であるので, 全ての $\text{OPT}_1[e, k, d]$ の計算にかかる時間は, $O(mk^4)$ である.

一方, OPT_2 の DP テーブルのサイズは, e の子数を q_e とすると, $O(\sum_{e \in \mathcal{E}} q_e \cdot k^2)$ となる. $\text{OPT}_2[e, i, k, d]$ の計算時間は高々 $O(k^2)$ であるので, 全体の計算時間は, $O(\sum_{e \in \mathcal{E}} q_e \cdot k^4)$ となる. ここで, $\sum_{e \in \mathcal{E}} q_e = |E(L(\mathcal{H}))| = m - 1$ なので, 全体の計算時間は $O(m \cdot k^4)$ となる. ゆえに, FCHGP は $O(m \cdot k^4)$ 時間で解ける. \square

参考文献

- [1] Edouard Bonnet, Bruno Escoffier, Vangelis Th. Paschos, and Emeric Tourniaire. Multi-parameter analysis for local graph partitioning problems: Us-

- ing greediness for parameterization. *Algorithmica*, 71(3):566–580, 2015. URL: <https://doi.org/10.1007/s00453-014-9920-6>, doi:10.1007/S00453-014-9920-6.
- [2] Leizhen Cai. Parameterized complexity of cardinality constrained optimization problems. *Comput. J.*, 51(1):102–121, 2008. doi:10.1093/comjnl/bxm086.
- [3] Fedor V. Fomin, Petr A. Golovach, Tanmay Inamdar, and Tomohiro Koana. FPT approximation and subexponential algorithms for covering few or many edges. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023, August 28 to September 1, 2023, Bordeaux, France*, volume 272 of *LIPIcs*, pages 46:1–46:8. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.
- [4] Tomohiro Koana, Christian Komusiewicz, André Nichterlein, and Frank Sommer. Covering many (or few) edges with k vertices in sparse graphs. In *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual Conference)*, volume 219 of *LIPIcs*, pages 42:1–42:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.STACS.2022.42.
- [5] Suguru Yamada and Tesshu Hanaka. Fixed-parameter algorithms for cardinality-constrained graph partitioning problems on sparse graphs. In Amitabh Basu, Ali Ridha Mahjoub, and Juan José Salazar González, editors, *Combinatorial Optimization - 8th International Symposium, ISCO 2024, La Laguna, Tenerife, Spain, May 22-24, 2024, Revised Selected Papers*, volume 14594 of *Lecture Notes in Computer Science*, pages 220–232. Springer, 2024. doi:10.1007/978-3-031-60924-4_17.