RA-001

正モジュラ関数の最適化

Posimodular Function Optimization

石井 利昌 † **牧野 和久** ‡
Toshimasa Ishii Kazuhisa Makino

1 はじめに

V を有限集合とし, $n = |V|$ とする. 集合関数 $f : 2^V \to \mathbb{R}$ は，任意の集合 $X, Y \subseteq V$ に対して,

$$f(X) + f(Y) \geq f(X \setminus Y) + f(Y \setminus X) \qquad (1.1)$$

をみたすとき，正モジュラ関数と呼ばれる．ただし，$\mathbb{R}$ は実数の集合を表す．正モジュラ性は，様々な組合せ最適化問題に関連して現れ，効率的なアルゴリズム設計の鍵となる重要かつ中心的な構造的性質の一つである (例えば，[6, 9, 14, 16, 17, 21] など)．その一例として，無向グラフのカット関数は正モジュラ性をみたすが，有向グラフのカット関数は正モジュラ性をみたさない．この正モジュラ性の有無の違いにより，多くのグラフ・ネットワーク最適化問題において，無向版の問題が有向版の問題より効率的に解ける．局所辺連結度増大問題 [5] や連結度要求またはコストが一様である場合の供給点配置問題 [1, 11, 23] は, 無向グラフでは多項式時間で解けるが, 有向グラフでは NP 困難である．より一般的には，正モジュラかつ劣モジュラ関数を最小化する現在最速のアルゴリズムの計算時間は $O(n^3 T_f)$ である [15] のに対して，劣モジュラ関数を最小化する現在最速の計算時間は $O(n^5 T_f + n^6)$ である [19]．ただし，任意の集合 $X, Y \subseteq V$ に対して,

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$$

をみたす集合関数 $f : 2^V \to \mathbb{R}$ を劣モジュラ関数といい, T_f は集合 $X \subseteq V$ に対して関数値 $f(X)$ の計算に要する時間を表す．こうした現象の要因として，正モジュラ関数に関する次の二つの構造的性質が挙げられる．

V の空でない真部分集合 X の任意の空でない真部分集合 Y が $f(Y) > f(X)$ をみたすとき，X は極点集合と呼ばれる．f が正モジュラ関数のとき，極点集合族 $\mathcal{X}(f)$ はラミナである (すなわち，任意の $X, Y \in \mathcal{X}(f)$ に対して, $X \cap Y = \emptyset$, $X \subseteq Y$, または $X \supseteq Y$) ことが知られている．無向グラフのカット関数 f に対する極点集合族 $\mathcal{X}(f)$ は, グラフの連結性の構造を表し，多くの効率的なグラフアルゴリズム設計に役立てられている [12, 25]．例えば, $\mathcal{X}(f)$ が既知であれば，連結度要求が一様である場合の無向供給点配置問題は線形時間で解ける [13]．また，f が無向カット関数のとき，$\mathcal{X}(f)$ は $O(n(m + n \log n))$ 時間で求められる [13] (n, m はそれぞれグラフの節点数，辺数を表す)．より一般的に，f が正モジュラかつ劣モジュラの場合，$\mathcal{X}(f)$ は $O(n^3 T_f)$ 時間で計算できる [14].

もう一方は，ソリッド集合と呼ばれる集合の構造的性質である．V の要素 v に対して，v を含む部分集合 X $(\subseteq V)$ が，$v \in Y$ である X の任意の空でない真部分集合 Y に対して $f(Y) > f(X)$ をみたすとき, X は (v に関する) ソリッド集合と呼ばれる．すべてのソリッド集合から成る族を，$\mathcal{S}(f)$ で表す (言い換えれば，$\mathcal{S}(f) = \bigcup_{v \in V} \{v$ に関するソリッド集合 $X\}$)．f が正モジュラであるとき, $\mathcal{S}(f)$ は木ハイパーグラフと呼ばれる特別な構造をもつ [21]．上記の極点集合族 $\mathcal{X}(f)$ に関する議論と同様に, $\mathcal{S}(f)$ に対する基本木 T が前もって分かっていれば，正モジュラ関数 f に関する最小横断問題は多項式時間で解ける．この最小横断問題は，コストが一様な場合の無向供給点配置問題 [23] や無向外部ネットワーク問題 [24] を一般化した問題である．さらに，f が劣モジュラでもある場合，基本木 T は多項式時間で求められる [21].

$\mathcal{X}(f)$ と $\mathcal{S}(f)$ に関するこれらの構造的性質は f の正モジュラ性から導出される性質で，劣モジュラ性はこれらの構造を効率的に計算するために仮定されている．より正確には，劣モジュラ性は，$\min\{f(X) \mid \emptyset \neq X \subseteq V\}$ が多項式時間で計算できる，という性質のために仮定されている．

一方で，我々が知る限り，正モジュラ性が現れる最適化問題に対する過去の結果では，劣モジュラ性または対称性を同時に仮定している．これは，考察された最適化問題がほとんど無向グラフに関連するものであり，そのカット関数が対称劣モジュラであることによる．ただし，任意の $X \subseteq V$ に対して，$f(X) = f(V \setminus X)$ をみたす集合関数 $f : 2^V \to \mathbb{R}$ を対称関数という．ここで，集合関数が対称であるとき，正モジュラ性と劣モジュラ性は同等であることに注意されたい．この関係は，f の対称性から $f(X) + f(Y) = f(V \setminus X) + f(Y)$ と $f(X \setminus Y) + f(Y \setminus X) = f((V \setminus X) \cup Y) + f((V \setminus X) \cap Y)$ が成り立つことから導出される．

以上の理由から，本研究では，次に定義される正モジュラ関数最小化問題の計算の複雑さについて考える．

正モジュラ関数最小化問題

入力: 正モジュラ関数 $f : 2^V \to \mathbb{R}$,

出力: $f(X^*) = \min_{X \subseteq V : X \neq \emptyset} f(X)$ をみたす $\emptyset \neq X^* \subseteq V$.

ここで，入力の関数 f の値は， $X \subseteq V$ に対して $f(X)$ を返すオラクルにより与えられると仮定する．また，最適値 $f(X^*)$ も出力されると仮定する．この正モジュラ関数最小化問題は，2010 年にエグレス (Egres) 未解決問題 のリストに，負モジュラ関数最大化問題として掲載されている [4]．ただし，集合関数 f は，$-f$ が正モジュラであるとき，負モジュラと呼ばれる．負モジュラ関数の一般化である弱優モジュラ関数を最大化する問題は，局所辺連結度増大問題や節点領域辺連結度増大問題など様々な問題 [3, 5, 8, 18, 22] に応用があることからも，正モジュラ関数最小化問題を考えることは重要である．また，本研究では，劣モジュラ関数最大化問題が (最小化問題と

* † 北海道大学，Hokkaido University
‡ 京都大学， Kyoto University

ともに) 近年よく研究されているのと同様，正モジュラ関数最大化についても考える．

本研究の主な成果として，以下の 1.–4. について示す．

1. 正モジュラ関数最小化問題を解くどのアルゴリズムも $\Omega(2^{\frac{n}{7.54}})$ 回のオラクル呼び出しが必要である．
2. 非負整数 d に対して，$D = \{0, 1, \ldots, d\}$ を f の値域とする (換言すると，$f : 2^V \to D$) とき，正モジュラ関数最小化問題を解くには $\Omega(2^{\frac{d}{15.08}})$ 回のオラクル呼び出しが必要である．一方，$O(n^d T_f + n^{2d+1})$ 時間 (すなわち，定数 d に対する多項式時間) で計算可能である．
3. 2. の状況下で, 極点集合族 $\mathcal{X}(f)$ を $O(n^d T_f + n^{2d+1})$ 時間で計算できる．さらに, すべての最適解を $O(nT_f)$ 時間遅延で列挙できる．ただし，最初の最適解は 2. の $O(n^d T_f + n^{2d+1})$ 時間必要である．
4. 正モジュラ関数最大化問題を解くには $\Omega(2^{n-1})$ 回のオラクル呼び出しが必要である．また，$D = \{0, 1, \ldots, d\}$ が f の値域である場合，d が定数ならば，計算時間が $\Theta(n^{d-1} T_f)$ である．

1. の計算困難性に関する結果は，劣モジュラ関数最小化問題が多項式時間で解けるという結果と対照的である．2. の結果は，d が定数であれば最小化問題は多項式時間で解けることを示している．また，3. の結果は，2. を利用することで得られる．最後に，4. の結果は，最大化問題も計算困難であることを示している．

本論文の構成は以下の通りである．2 章では, 正モジュラ関数最小化問題の計算困難さを示す．3 章では, D が f の値域である場合，$O(n^d T_f + n^{2d+1})$ 時間で最小化問題を解くアルゴリズムを与える．4 章では，正モジュラ関数最大化問題について考える．ページ数の都合上，いくつかの証明を省略する．証明の詳細は [10] を参照されたい．

2　正モジュラ関数最小化問題の計算困難性

V を有限集合とし, $n = |V|$ とする．$f : 2^V \to \mathbb{R}$ を正モジュラ関数とする．$X = Y$ のとき (1.1) は $f(X) + f(X) \geq f(\emptyset) + f(\emptyset)$ であることから，任意の正モジュラ関数 f において，すべての $X \subseteq V$ は $f(X) \geq f(\emptyset)$ をみたす．本論文では，$f(\emptyset) = 0$ と仮定する (そうでないときは $f(X)$ を $f(X) - f(\emptyset)$ と定義し直して考えればよい)．

本章では，正モジュラ関数最小化問題を解くのに必要なオラクル呼び出し回数について考察する．

$g : 2^V \to \mathbb{R}_+$ を，任意の $X \subseteq V$ に対して $g(X) = |X|$ である集合関数とする．g は単調 ($X \supseteq Y$ である $X, Y \subseteq V$ に対して $g(X) \geq g(Y)$) なので，明らかに正モジュラである．$k \leq n/2$ である正整数 k に対して，S を $|S| = 2k$ である V の部分集合とする．集合関数 $g_S : 2^V \to \mathbb{R}_+$ を

$$g_S(X) = \begin{cases} 2k - |X| & (X \subseteq S, |X| \geq k+1 \text{ の場合}) \\ g(X) & (\text{それ以外}) \end{cases}$$

と定義する．このとき，g_S も正モジュラである．

主張 2.1 **g_S は正モジュラ関数である．** □

$\mathcal{G} = \{g\} \cup \{g_S \mid S \subseteq V, |S| = 2k\}$ とする．以下では，$\mathcal{G}$ に含まれる正モジュラ関数を区別するために指数回のオラクル呼び出しが必要であることを示す．

$\mathcal{S} = \{S \subseteq V \mid |S| = 2k\}$，$\mathcal{T} = \{T \subseteq V \mid k+1 \leq |T| \leq 2k\}$ とする．ここで，次の整数計画問題を考える．

$$\begin{array}{lll} \text{minimize} & \sum_{T \in \mathcal{T}} z_T & \\ \text{subject to} & \sum_{T \in \mathcal{T} : T \subseteq S} z_T \geq 1 & \forall S \in \mathcal{S}, \\ & z_T \in \{0, 1\} & \forall T \in \mathcal{T}. \end{array} \tag{2.1}$$

$\mathcal{G}$ に属す任意の正モジュラ関数 f は，$|X| \leq k$ または $|X| \geq 2k + 1$ のとき $f(X) = g(X)$ をみたすことに注意されたい．したがって，このような集合 X に対するオラクル呼び出しは $\mathcal{G}$ の関数の区別に役立たない．このことから，$\mathcal{T}$ に属す集合 T に対するオラクル呼び出しに着目し，次の補題を得る．

補題 2.2 **q_k を整数計画問題 (2.1) の最適値とする．このとき，$\mathcal{G}$ に属す正モジュラ関数を区別するのに少なくとも q_k 回のオラクル呼び出しが必要である．**

証明: 高々 $q_k - 1$ 回のオラクル呼び出しで $\mathcal{G}$ に属す正モジュラ関数を区別するアルゴリズム A が存在すると仮定して，矛盾を導く．関数 g が A の入力である場合，A によって呼び出される V の集合族を $\mathcal{X}$ で表す．$|\mathcal{X}| \leq q_k - 1$ なので, $\mathcal{S}$ のある集合 S に対して，$\mathcal{X}$ に属すどの集合 X も $X \not\subseteq S$ または $|X| < k + 1$ をみたす．このことは，任意の $X \in \mathcal{X}$ に対して $g_S(X) = g(X)$ が成立することを意味し，A が g と g_S を区別することに矛盾する． □

補題 2.2 により，正モジュラ関数最小化問題を解くには q_k 回のオラクル呼び出しが必要である．ここで，整数計画問題 (2.1) を線形緩和することにより次の補題を得る．

補題 2.3 **q_k を整数計画問題 (2.1) の最適値とする．このとき，$q_k \geq \binom{n}{k+1} / \binom{2k}{k+1}$ が成り立つ．** □

$k \geq 2$ に対して，

$$\frac{\binom{n}{k+1}}{\binom{2k}{k+1}} \geq \frac{2\pi}{e^2} \cdot \left(\frac{n}{4k}\right)^{k+1} \tag{2.2}$$

が成り立つ．このことは，スターリングの公式 $\sqrt{2\pi} n^{n+1/2} e^{-n} \leq n! \leq e n^{n+1/2} e^{-n}$，および $n \geq n - k - 1$ と $(1 - \frac{1}{k})^{k-1/2} \geq \frac{1}{2\sqrt{2}}$ ($k \geq 2$ のとき) から得られる．$n = \lceil 4ek \rceil$ のとき，(2.2) の右辺は (ほぼ) 最大となり，$\Omega(e^{\frac{n}{4e}}) = \Omega(2^{\frac{n}{7.54}})$ を得る．

定理 2.4 **正モジュラ関数最小化問題を解くどのアルゴリズムも $\Omega(2^{\frac{n}{7.54}})$ 回のオラクル呼び出しが必要である．** □

次に，f の値域が非負整数 d に対して $D = \{0, 1, \ldots, d\}$ である場合について考える．定理 2.4 の証明と同様に指数回のオラクル呼び出しが必要であることを示す．

T を，$|T| = \lfloor d/2 \rfloor$ である V の部分集合とする．集合関数 $g : 2^V \to D$ を

$$g(X) = \begin{cases} |X| & (X \subseteq T \text{ の場合}) \\ |T| + |T \cap X| & (\text{それ以外}) \end{cases}$$

と定義する．$2k \leq |T|$ である正整数 k に対して，S を $|S| = 2k$ である T の部分集合とする．集合関数 $g_S : 2^V \to D$ を

$$g_S(X) = \begin{cases} 2k - |X| & (X \subseteq S, |X| \geq k+1 \text{ の場合}) \\ g(X) & (\text{それ以外}) \end{cases}$$

と定義する．このとき，g と g_S はともに正モジュラ関数である． $k \geq 2$ かつ $|T| \approx 4ek$ (すなわち $d \approx 8ek$) ならば, 補題 2.2, 2.3 と同様の議論により次の定理が成り立つ.

定理 2.5 与えられた正モジュラ関数の値域が D であるとき，正モジュラ関数最小化問題を解くには $\Omega(2^{\frac{d}{15.08}})$ 回のオラクル呼び出しが必要である． □

3 d が定数の場合の正モジュラ関数最小化問題に対する多項式時間アルゴリズム

本章では，入力の正モジュラ関数の値域が $\{0, 1, \ldots, d\}$ に限定される場合を考え，d が定数ならば正モジュラ最小化問題が多項式時間で解けることを示す．まず，$d \leq 3$ の場合，半極点集合と呼ばれる集合の縮約操作を繰り返すことで問題が効率的に解けることを示し，その後 一般の d に対して $O(n^d T_f + n^{2d+1})$ 時間アルゴリズムを与える．

この章を通じて，問題の最適解を f の最小解と呼ぶ.

3.1 $d \in \{0, 1, 2, 3\}$ の場合

$f : 2^V \to \{0, 1, \ldots, d\}$ を集合関数とし, s を V に属さない要素とする．$S \subseteq V$ に対して, $f' : 2^{(V \setminus S) \cup \{s\}} \to \{0, 1, \ldots, d\}$ を,

$$f'(X) = \begin{cases} f(X) & (s \notin X \text{ の場合}) \\ f((X \setminus \{s\}) \cup S) & (s \in X \text{ の場合}) \end{cases}$$

である関数とする．このとき， f' は f から集合 S を要素 s に縮約することで得られるという．ここで，正モジュラ関数から縮約により得られる関数も正モジュラであることに注意されたい．V の空でない真部分集合 X のすべての空でない真部分集合 Y が $f(Y) \geq f(X)$ をみたすとき，X は半極点集合と呼ばれる．次の補題は，どの半極点集合を縮約しても影響を受けない f の最小解が少なくとも一つ存在することを示唆する．

補題 3.1 f を正モジュラ関数とする. 任意の半極点集合 X に対して, $Y \supseteq X$ または $X \cap Y = \emptyset$ をみたす f の最小解 Y が存在する． □

次の補題は，$d \leq 3$ ならば，サイズ 2 の集合を高々 n 回縮約することで f の最小解が求められることを示す.

補題 3.2 $d \leq 3$ の場合, $|X| = 2$ である半極点集合 X が存在するか，$|Y| \in \{1, n-1, n\}$ である f の最小解 Y が存在する． □

この補題により，次の定理を得る.

定理 3.3 $d \leq 3$ の場合, 正モジュラ関数最小化問題は $O(n^2 T_f)$ 時間で解くことができる． □

一方で，d が一般の場合，補題 3.2 のような性質は必ずしも成立しない．次の例では，自明でない (2 要素以上から成る) 半極点集合のサイズが d に依存せず，いくらでも大きくなり得ることを示す.

例 3.4 S を $|S| \geq 4$ である V の部分集合とし，$f : 2^V \to \{0, 1, \ldots, 7\}$ を

$$f(X) = \begin{cases} 0 & (X = \emptyset, S \text{ の場合}) \\ 1 & (X \subseteq S, |X| \in \{1, |S|-1\} \text{ の場合}) \\ 2 & (X \subseteq S, 2 \leq |X| \leq |S|-2 \text{ の場合}) \\ 2 & (X \cap S = \emptyset, |X| = 1 \text{ の場合}) \\ 3 & (X \cap S = \emptyset, |X| \geq 2 \text{ の場合}) \\ 4 & (X \setminus S \neq \emptyset, |X \cap S| = 1 \text{ の場合}) \\ 5 & (X \setminus S \neq \emptyset, 2 \leq |X \cap S| \leq |S|-2 \text{ の場合}) \\ 6 & (X \setminus S \neq \emptyset, |X \cap S| = |S|-1 \text{ の場合}) \\ 7 & (X \setminus S \neq \emptyset, X \cap S = S \text{ の場合}) \end{cases}$$

と定義される正モジュラ関数とする．この例では，$\{S\} \cup \{\{v\} \mid v \in V\} \cup \{S \setminus \{v\} \mid v \in S\}$ が f の半極点集合の族である．したがって，自明でない半極点集合のサイズは, $|S|$ または $|S|-1$ であり，これらは d と独立である.

3.2 一般の d の場合

ここでは，一般の d に対する多項式時間アルゴリズムを与える．ただし，前節の $d \leq 3$ の場合と異なり，半極点集合の縮約を用いない．その代わりに，正モジュラ性から得られる次の性質 (補題 3.5) に着目し，双対ホーン充足可能性問題を利用することで問題を解く.

補題 3.5 非負整数 d に対して，$f : 2^V \to \{0, 1, \ldots, d\}$ を正モジュラ関数とする．このとき，

$$f(X) \geq f(X \cup \{s\}) \tag{3.1}$$

をみたす集合 $X \subseteq V$ と要素 $s \in V \setminus X$ が存在するならば，$Y \cap X = \emptyset$ である任意の集合 Y は $f(Y) \geq f(Y \setminus \{s\})$ をみたす.

証明: $s \notin Y$ の場合は明らかに $f(Y) = f(Y \setminus \{s\})$. $s \in Y$ の場合は, (1.1) より $f(X \cup \{s\}) + f(Y) \geq f(X) + f(Y \setminus \{s\})$ であることから, $f(Y) \geq f(Y \setminus \{s\})$ が成り立つ. □

ここで, 局所的に極小な f の最小解を考える．ただし，集合 X^* が，任意の $v \in X^*$ に対して，$f(X^*) < f(X^* \setminus \{v\})$ であるとき，X^* を局所的に極小という．1 点からなる最小解が存在しなければ，局所的に極小な最小解は必ず存在し，そのサイズは 2 以上であることに注意されたい.

(3.1) をみたす集合 $X \subseteq V$ と要素 $s \in V \setminus X$ を考える．補題 3.5 により, 任意の局所的極小集合 X^* は，$X^* \cap X = \emptyset$ ならば必ず $s \notin X^*$ をみたす．これを論理式として表現するために，論理変数 $x_v, v \in V$ を導入し，論理ベクトル $x \in \{0, 1\}^V$ を $S_x = \{v \in V \mid x_v = 1\}$ と同一視する (換言すると，x は S_x の特性ベクトル)．このとき，上記の局所的極小集合 X^* に関する性質は

$$x_v = 0 \ (v \in X) \text{ ならば, 必ず } x_s = 0 \text{ が成り立つ,} \tag{3.2}$$

と表現できる．この (3.2) は，双対ホーン節

$$\bigvee_{v \in X} x_v \vee \overline{x}_s. \tag{3.3}$$

をみたすことと同値である．ただし，論理変数 x_v またはその否定 $\overline{x}_v$ のことをリテラルといい，リテラルが論理和で結合された論理式 $c = \bigvee_{v \in P(c)} x_v \vee \bigvee_{v \in N(c)} \overline{x}_v$ は，$P(c) \cap N(c) = \emptyset$ のとき，節と呼ばれる．特に，高々一つの負リテラルをもつ節を双対ホーン節という．

(3.1) をみたす X と s のペアが多ければ，対応するルール (3.3) により局所的極小最小解を見つけるための探索空間を減らすことができる．これらに対応する (3.3) は双対ホーン CNF として表現できるため，その充足可能性問題は (入力サイズの) 線形時間で解け，すべての充足解は線形時間遅延 (換言すると，連続する二つの解を出力する間隔が線形時間) で列挙できる [20]. ただし，節を論理積により結合した論理式を論理積標準形 (CNF) と呼び，特に各節が双対ホーン節である CNF を双対ホーン CNF と呼ぶ．一方，このような X と s のペア数は一般に n の指数である．本論文では，以下の 1. –3. の条件をみたす X と s のペア集合 $\mathcal{P}$ の構成法を示す (ただし，d は定数とする).

1. $|\mathcal{P}|$ は n の多項式オーダー.
2. 対応する双対ホーン CNF の充足解の数は n の多項式オーダー.
3. $\mathcal{P}$ を多項式時間で計算可能である.

定義 3.6 X **を** $k = |X|$ **である** V **の部分集合とする．任意の** $i = 1, 2, \ldots, k$ **に対して** $f(X_i) > f(X_{i-1})$ **をみたす集合の列** $X_0 (= \emptyset) \subsetneq X_1 \subsetneq \cdots \subsetneq X_k (= X)$ **が存在するとき，**X **は (**$\emptyset$ **から) 到達可能であるといい，そうでないとき到達不可能という．**

定義より，$\emptyset$ は到達可能である．真部分集合がすべて到達可能であるような到達不可能集合は極小と呼ばれる．極小到達不可能集合の族を $\mathcal{U}$ で表す．到達可能性の定義から次の補題を得る．

補題 3.7 $U \in \mathcal{U}$ **を極小到達不可能集合とする．すべての** $u \in U$ **に対して** $f(U) \leq f(U \setminus \{u\})$ **である．** □

補題 3.8 X^* **を正モジュラ関数** f **に関して局所的に極小な集合とする．このとき，**X^* **の特性ベクトルは，**

$$\varphi_f = \bigwedge_{U \in \mathcal{U}} \bigwedge_{s \in U} (\bigvee_{u \in U \setminus \{s\}} x_u \vee \overline{x}_s) \qquad (3.4)$$

で定義される双対ホーン *CNF* **を充足する．** □

以上の議論より，次のアルゴリズムは正モジュラ関数の最小解を出力する．

アルゴリズム MinPosimodular(f)

ステップ 1. $f(\{v^*\}) = \min\{f(\{v\}) \mid v \in V\}$ をみたす 1 点集合 $\{v^*\}$ を計算する．

ステップ 2. $|S_x| \geq 2$ かつ $\varphi_f(x) = 1$ をみたす集合 S_x の中で f-値が最小のもの S_{x^*} を計算する．

ステップ 3. $f(\{v^*\}) \leq f(S_{x^*})$ ならば $\{v^*\}$ を出力し，そうでなければ S_{x^*} を出力し，停止する． □

以下では，φ_f の節数，充足解数がともに (d を定数とするとき) n の多項式オーダーであることを示す．

まず，次の補題で極小到達不可能集合に関する基本的な性質を示す．ただし，集合 $I \subseteq V$ は，$\mathcal{U}$ に属すどの集合も含まないとき，$\mathcal{U}$ と独立であるという．

補題 3.9 正モジュラ関数 $f : 2^V \to \{0, 1, \ldots, d\}$ **に対して，次の三つの性質が成り立つ．**

(i) **任意の** $U \in \mathcal{U}$ **は** $1 \leq |U| \leq d + 1$ **をみたす．**
(ii) $\mathcal{U}$ **と独立な任意の集合** I **は** $|I| \leq d$ **をみたす．**
(iii) **一点集合** $\{u\}$ **が** $\mathcal{U}$ **に属するならば，**$f(\{u\}) = 0$ **が成立し，従って** $\{u\}$ **は** f **の最小解である．** □

補題 3.9 (i) により，$d < n/2$ のとき $|\mathcal{U}| = O(\binom{n}{d+1})$, そうでないとき $O(\binom{n}{n/2})$ である．すなわち，$|\mathcal{U}| = O(n^{d+1}/d)$ が成り立つ．

ここで，φ_f の充足解数を解析する．議論を簡単にするため，確定ホーン CNF $\varphi_f(\overline{x})$ を考える．ただし，$\overline{x}$ は x の否定を表し，各節に含まれる正リテラルの数が丁度一つである CNF を確定ホーン CNF と呼ぶ．$\varphi_f(\overline{x}) = 1$ である集合 S_x は局所的極小な f の最小解の補集合の候補であることに注意する．確定ホーン CNF φ と集合 $T \subseteq V$ に対して，以下に示す前向き鎖手続き (forward chaining procedure, FCP) は φ の充足解を求める方法としてよく知られている [2, 7].

手続き FCP(φ; T)

ステップ 0. $Q := T$ とする．

ステップ 1. φ において，$N(c) \subseteq Q$ かつ $P(c) \cap Q = \emptyset$ をみたす節 c が存在するかぎり，次の手続きを行う．

$$Q := Q \cup P(c).$$

Step 2. Q を $FCP(\varphi; T)$ として出力し，停止する． □

任意の集合 T に対して $T \subseteq FCP(\varphi; T)$ が成り立ち，$T \subseteq T'$ ならば $FCP(\varphi; T) \subseteq FCP(\varphi; T')$ が成り立つことに注意されたい．さらに，確定ホーン CNF φ に対して，T が φ の充足解に対応する (換言すると T の特性ベクトルが φ の充足解) ならば，そのときに限り，$T = FCP(\varphi; T)$ であることが知られている [2, 7]. このことは，任意の集合 T に対する $FCP(\varphi; T)$ は φ の充足解に対応し，φ の任意の充足解 α に対応する集合 T (換言すると $S_\alpha = FCP(\varphi; T)$) が存在することを示唆する．

以下では，$\varphi_f(\overline{x})$ の任意の充足解 α に対して，$|T| \leq d$ かつ $S_\alpha = FCP(\varphi_f(\overline{x}); T)$ をみたす集合 T が存在することを証明する．この性質は，φ_f の充足解の数は $\sum_{i=0}^{d} \binom{n}{i}$ 以下であることを示唆する．

$\varphi(\overline{x})$ の充足解 α に対して，$\mathcal{U}_\alpha = \{U \in \mathcal{U} \mid U \subseteq S_\alpha\}$ とし，$I \subseteq S_\alpha$ を，$\mathcal{U}_\alpha$ と独立な S_α の部分集合で極大なものとする (すなわち，任意の $v \in S_\alpha \setminus I$ に対して $I \cup \{v\}$ は $\mathcal{U}_\alpha$ と独立でない). $\emptyset$ は $\mathcal{U}_\alpha$ と独立であることから，そのような集合 I は必ず存在することに注意されたい．

補題 3.10 正モジュラ関数 $f : 2^V \to \{0, 1, \ldots d\}$ **に対して，**α **を** $\varphi_f(\overline{x})$ **の充足解，**I **を補題の直前で定義される集合とする．このとき** $S_\alpha = FCP(\varphi_f(\overline{x}); I)$ **が成立する．**

証明: $I = S_\alpha$ ならば，α が $\varphi_f(\overline{x})$ の充足解であることから，$S_\alpha = FCP(\varphi_f(\overline{x}); S_\alpha)$ である．$S_\alpha \setminus I \neq \emptyset$ の場合を考える．このとき， 任意の要素 $v \in S_\alpha \setminus I$ に対して，$I \cup \{v\}$ は $\mathcal{U}_\alpha$ と独立でない．すなわち，ある $U \in \mathcal{U}_\alpha$ が

$U \setminus I = \{v\}$ をみたす．このことは，$\varphi_f(\overline{x})$ が，$P(c) = \{v\}$ かつ $N(c) = U \setminus \{v\} (\subseteq I)$ をみたす節 c を含むことを示唆する．これらのことから，$\mathrm{FCP}(\varphi_f(\overline{x}); I)$ は $S_\alpha \setminus I$ に属すすべての点 v を含み，$S_\alpha \subseteq FCP(\varphi_f(\overline{x}); I)$ が成り立つ．一方，$I \subseteq S_\alpha$ かつ α は $\varphi_f(\overline{x})$ の充足解であることから，$S_\alpha = FCP(\varphi_f(\overline{x}); I)$ である． □

補題 3.11 正モジュラ関数 $f : 2^V \to \{0, 1, \ldots d\}$ に対して, $|\{x \in \{0,1\}^n \mid \varphi_f(x) = 1\}| \le \sum_{i=0}^{d} \binom{n}{i} (= O(n^d))$ **が成り立つ**.

証明: 補題 3.10 より, φ_f の任意の充足解 $\overline{\alpha}$ に対して，$S_\alpha = FCP(\varphi_f(\overline{x}); I)$ であり，$\mathcal{U}_\alpha$ と独立な集合 I が存在する．I は $\mathcal{U}$ とも独立なので, 補題 3.9 より $|I| \le d$ が成り立ち，補題が証明される． □

注意 3.12 補題 3.10 は，MinPosimodular(f) のステップ 2 が $|T| \le d$ であるすべての集合 $T \subseteq V$ に対して FCP を適用することで実行されることを示唆する. 各 T に対して, $FCP(\varphi_f(\overline{x}); T)$ は $\mathcal{U}$ を用いて $O(d|\mathcal{U}|) = O(n^{d+1})$ 時間で計算できる．したがって，$\mathcal{U}$ を計算した後，ステップ 2 は $O(n^d(n^{d+1} + T_f)) = O(n^{2d+1} + n^d T_f)$ 時間で実行できる．

以上の議論より次の定理を得る．

定理 3.13 一般の d に対して, 正モジュラ関数最小化問題は $O(n^d T_f + n^{2d+1})$ 時間で解ける．

証明: MinPosimodular(f) の計算時間を解析する. 明らかに，ステップ 1 と 3 はそれぞれ $O(nT_f)$ 時間，$O(n)$ 時間で実行できる．ステップ 2 に関して，極小到達不可能集合族 $\mathcal{U}$ は $O(n^d T_f + n^{d+1})$ 時間で計算できる．このとき，$\mathcal{U}$ を計算するのに，サイズ d 以下の集合の f-値を知っていれば十分で，サイズ $d + 1$ の集合の f-値は必要ないことに注意する．このことと注意 3.12 より，ステップ 2 は $O(n^d T_f + n^{2d+1})$ 時間で計算できる． □

定理 3.13 の系として, 次の結果が得られる．

系 3.14 正モジュラ関数 $f : 2^V \to \{0, 1, \ldots, d\}$ に対して, MinPosimodular(f) により $O(n^d T_f + n^{2d+1})$ 時間で最小解を見つけた後，$O(nT_f)$ 時間遅延ですべての f の最小解を列挙できる． □

系 3.15 正モジュラ関数 $f : 2^V \to \{0, 1, \ldots, d\}$ に対して, $O(n^d T_f + n^{2d+1})$ 時間で f の極点集合族 $\mathcal{X}(f)$ を計算できる． □

4 正モジュラ関数の最大化

本章では，次に定義される正モジュラ関数最大化問題について考える．

正モジュラ関数最大化問題

入力: 正モジュラ関数 $f : 2^V \to \mathbb{R}$,

出力: $f(X^*) = \max_{X \subseteq V: X \neq \emptyset} f(X)$ をみたす $\emptyset \neq X^* \subseteq V$.

ここで，最適値 $f(X^*)$ も出力されると仮定する．最小化問題の場合と同様に，この問題も一般の場合計算困難である．

定理 4.1 正モジュラ関数最大化問題を解くどのアルゴリズムも 2^{n-1} 回のオラクル呼び出しが必要である.

証明: まず n が偶数の場合について考える．正整数 k に対して，$n = 2k$ とする．$g : 2^V \to \mathbb{R}_+$ を，$|X| \le k - 1$ のとき $g(X) = |X|$，それ以外のとき $g(X) = k$ である集合関数とする．$|S| \ge k$ である集合 $S \subseteq V$ に対して，$g_S : 2^V \to \mathbb{R}_+$ を，$X \neq S$ のとき $g_S(X) = g(X)$，$X = S$ のとき $g_S(X) = k + 1$ である集合関数と定義する．このとき，g, g_S ともに正モジュラである．

$q = \sum_{i=k}^{n} \binom{n}{i} (\ge 2^{n-1})$ とする. 高々 $q - 1$ 回のオラクル呼び出しで正モジュラ関数最大化問題を解くアルゴリズム A が存在すると仮定する．関数 g が A の入力である場合，A によって呼び出される V の集合族を $\mathcal{X}$ で表す．$|\mathcal{X}| \le q - 1$ より，$S \notin \mathcal{X}$ かつ $|S| \ge k$ をみたす集合 S が存在する．このことは，任意の $X \in \mathcal{X}$ に対して $g_S(X) = g(X)$ が成立することを意味し，A が g と g_S を区別することに矛盾する (言い換えると，A は最適値が k か $k + 1$ か判別できない).

n が奇数の場合については，非負整数 k に対して $n = 2k+1$ とし, $g : 2^V \to \mathbb{R}_+$ を, $|X| \le k$ のとき $g(X) = |X|$, それ以外のとき $g(X) = k+1$ である集合関数とし, $|S| \ge k+1$ である集合 $S \subseteq V$ に対して，$g_S : 2^V \to \mathbb{R}_+$ を，$X \neq S$ のとき $g_S(X) = g(X)$，$X = S$ のとき $g_S(X) = k + 2$ である集合関数とすることで，同様に証明できる． □

次に，非負整数 d に対して $f : 2^V \to \{0, 1, \ldots, d\}$ である場合について考える．このとき，正モジュラ関数の最大化問題の計算の複雑さに関して次のタイトな性質が成り立つ．

定理 4.2 $f : 2^V \to \{0, 1, \ldots, d\}$ に対する正モジュラ関数最大化問題は，定数 d に対して，$\Theta(n^{d-1} T_f)$ 時間で解くことができる． □

次の補題は，問題の計算量の下界に関するものである．上界に関しては次の節で示す．

補題 4.3 $n \ge 2d - 2$ の場合，$f : 2^V \to \{0, 1, \ldots, d\}$ に対する正モジュラ関数最大化問題を解くには，$\Omega(n^{d-1})$ 回のオラクル呼び出しが必要である．

証明: $g : 2^V \to \{0, 1, \ldots, d\}$ を，$|X| \le d - 2$ のとき $g(X) = |X|$，$|X| \ge d - 1$ のとき $g(X) = d - 1$ と定義される集合関数とする．$|S| \ge n - d + 1 (\ge d - 1)$ である集合 $S \subseteq V$ に対して，$g_S : 2^V \to \{0, 1, \ldots, d\}$ を，$X \neq S$ のとき $g_S(X) = g(X)$，$X = S$ のとき $g_S(X) = d$ と定義される集合関数とする．このとき，g, g_S ともに正モジュラである．さらに，$\{g\} \cup \{g_S \mid S \subseteq V, |S| \ge n-d+1\}$ に属す関数を区別するために，少なくとも $\sum_{i=n-d+1}^{n} \binom{n}{i} = \Omega(n^{d-1})$ 回のオラクル呼び出しが必要であることを示すことができる (証明は省略). □

4.1 定数 d に対する多項式時間アルゴリズム

ここでは，定数 d に対して，正モジュラ関数最大化問題が $O(n^{d-1} T_f)$ 時間で解けることを示す．問題の最適解を，f の最大解と呼ぶ.

補題 4.4 $f : 2^V \to \{0, 1, \ldots, d\}$ を正モジュラ関数とする．このとき，任意の極大な最大解 S は $|S| \ge n - d$ をみたす． □

サイズが $n-d+1$ 以上の f の最大解が存在する場合は，$|X| \geq n-d+1$ であるすべての集合 X を調べることで極大な f の最大解を $O(n^{d-1}T_f)$ 時間で見つけることができる．

以下では，サイズが $n-d+1$ 以上の f の最大解が存在しない場合を考える．このとき，次の補題が成り立つ．

補題 4.5 サイズが $n-d+1$ 以上の f の最大解が存在しないと仮定する．

(*i*) $|S_1| = |S_2| = d$ **をみたす，互いに素な最大解** S_1, S_2 **が存在する**.

(*ii*) S **を** $|S| = d$ **である** f **の最大解とし，**X **を** $|X| = f(X) = d-1$ **かつ** $X \cap S = \emptyset$ **である** V **の部分集合とする．このとき，**$f(X \cup \{v\}) = d$ **である要素** $v \in V \setminus (S \cup X)$ **が存在する．** □

補題 4.6 $\mathcal{X}$ **を，サイズ** d **の任意の最大解** S **に対して** $X \cap S \neq \emptyset$ **をみたす，サイズ** $d-1$ **の集合** X **の族とする．このとき，サイズが** $n-d+1$ **以上の** f **の最大解が存在しないならば，**$|\mathcal{X}| = O(n^{d-3})$ **が成り立つ**.

証明: 補題 4.5(i) より，$|S_1| = |S_2| = d$ かつ $S_1 \cap S_2 = \emptyset$ をみたす二つの最大解 S_1 と S_2 が存在する．明らかに，$|\mathcal{X}|$ は，$X \cap S_1, X \cap S_2 \neq \emptyset$ をみたすサイズ $d-1$ の集合 X の数以下である．この X の数は $\sum_{i,j>0, i+j \leq d-1} \binom{d}{i}\binom{d}{j}\binom{n-2d}{d-1-i-j} = O(n^{d-3})$ である． □

定数 c は，補題 4.6 の $\mathcal{X}$ に対して，$|\mathcal{X}| \leq cn^{d-3}$ をみたすものとする．以上の性質から，集合族 $\mathcal{X}_1 = \{X \subseteq V \mid |X| = d-1, f(X) = d-1\}$ から $\min\{cn^{d-3}+1, |\mathcal{X}_1|\}$ 個の集合を調べることで，ある要素 $v \in V \setminus X$ に対して，$X \cup \{v\}$ が f の最大解である集合 $X \in \mathcal{X}_1$ を求められる．したがって，$O((n^{d-1} + n|\mathcal{X}|)T_f) = O(n^{d-1}T_f)$ 時間で f の最大解を見つけることができる．

これらの性質に基づき，次の定理を得る．

定理 4.7 $f : 2^V \to \{0, 1, \ldots, d\}$ **に対する正モジュラ関数最大化問題は，定数** d **に対して，**$O(n^{d-1}T_f)$ **時間で解くことができる．** □

参考文献

[1] K. Arata, S. Iwata, K. Makino, and S. Fujishige, *Locating sources to meet flow demands in undirected networks*, Journal of Algorithms, 42 (2002), pp. 54–68.

[2] M. Arias and J. L. Balcázar, *Canonical Horn representations and query learning*, in Algorithmic Learning Theory, 2009, pp. 156–170.

[3] J. Bang-Jensen, A. Frank, and B. Jackson, *Preserving and increasing local edge-connectivity in mixed graphs*, SIAM J. Discrete Math., 8 (1995), pp. 155–178.

[4] Egres open problem list, http://lemon.cs.elte.hu/egres/open/Maximizing_a_skew-supermodular_function.

[5] A. Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM Journal on Discrete Mathematics, 5 (1992), pp. 25–53.

[6] S. Fujishige, *A laminarity property of the polyhedron described by a weakly posi-modular set function*, Discrete Applied Mathematics, 100 (2000), pp. 123–126.

[7] P. L. Hammer and A. Kogan, *Optimal compression of propositional Horn knowledge bases: Complexity and approximation*, Artificial Intelligence, 64 (1993), pp. 131–145.

[8] T. Ishii, *Minimum augmentation of edge-connectivity with monotone requirements in undirected graphs*, Discrete Optimization, 6 (2009), pp. 23–36.

[9] T. Ishii and K. Makino, *Posi-modular systems with modulotone requirements under permutation constraints*, Discrete Mathematics, Algorithms and Applications, 2 (2010), pp. 61–76.

[10] ———, *Posimodular function optimization*, CoRR, abs/1410.6030 (2014).

[11] H. Ito, K. Makino, K. Arata, S. Honami, Y. Itatsu, and S. Fujishige, *Source location problem with flow requirements in directed networks*, Optimization Methods and Software, 18 (2003), pp. 427–435.

[12] E. L. Lawler, *Cutsets and partitions of hypergraphs*, Networks, 3 (1973), pp. 275–285.

[13] H. Nagamochi, *Graph algorithms for network connectivity problems*, Journal of the Operations Research Society of Japan, 47 (2004), pp. 199–223.

[14] H. Nagamochi, *Minimum degree orderings*, Algorithmica, 56 (2010), pp. 17–34.

[15] H. Nagamochi and T. Ibaraki, *A note on minimizing submodular functions*, Inf. Process. Lett., 67 (1998), pp. 239–244.

[16] ———, *Polyhedral structure of submodular and posi-modular systems*, Discrete Applied Mathematics, 107 (2000), pp. 165–189.

[17] H. Nagamochi, T. Shiraki, and T. Ibaraki, *Augmenting a submodular and posi-modular set function by a multigraph*, Journal of Combinatorial Optimization, 5 (2001), pp. 175–212.

[18] Z. Nutov, *Approximating connectivity augmentation problems*, ACM Transactions on Algorithms, 6 (2009).

[19] J. Orlin, *A faster strongly polynomial time algorithm for submodular function minimization*, Mathematical Programming, 118 (2009), pp. 237–251.

[20] D. Pretolani, *A linear time algorithm for unique Horn satisfiability*, Information Processing Letters, 48 (1993), pp. 61 – 66.

[21] M. Sakashita, K. Makino, H. Nagamochi, and S. Fujishige, *Minimum transversals in posi-modular systems*, SIAM Journal on Discrete Mathematics, 23 (2009), pp. 858–871.

[22] Z. Szigeti, *Edge-connectivity augmentation of graphs over symmetric parity families*, Discrete Mathematics, 308 (2008), pp. 6527 – 6532.

[23] H. Tamura, H. Sugawara, M. Sengoku, and S. Shinoda, *Plural cover problem on undirected flow networks*, IEICE Transactions, J81-A (1998), pp. 863–869. (in Japanese).

[24] J. van den Heuvel and M. Johnson, *The external network problem with edge- or arc-connectivity requirements*, in Combinatorial and Algorithmic Aspects of Networking, vol. 3405 of Lecture Notes in Computer Science, Springer, 2004, pp. 114–126.

[25] T. Watanabe and A. Nakamura, *Edge-connectivity augmentation problems*, Journal of Computer System Sciences, 35 (1987), pp. 96–144.